

Práticas recomendadas para criar pastas de trabalho eficientes do Tableau

Tableau 10

Alan Eldridge
Tableau Software

Sobre este documento

Eu gostaria de lembrar que este documento é composto por materiais escritos por diversos autores. O que fiz foi reuni-los em um único documento e definir uma estrutura. Algumas pessoas reconhecerão suas palavras em cada uma das seções, e até mesmo em blocos inteiros de texto. Agradeço a todos pelo excelente trabalho e por não reclamarem os direitos autorais. Se não fosse por vocês, este documento não existiria.

Eu também gostaria de agradecer às pessoas que revisaram este documento para garantir sua precisão e fluidez. Sua atenção aos detalhes e explicações tornaram este texto muito mais legível do que ele seria se eu o tivesse produzido sozinho.

Este documento foi atualizado para refletir os recursos do Tableau 10. As próximas versões do Tableau terão novos recursos e funcionalidades que podem alterar algumas destas recomendações.

Atenciosamente,

Alan Eldridge
Junho de 2016

Muito longo, não li

Várias pessoas que leram ou recomendaram este whitepaper já me disseram que ele é muito longo. Costumo responder que ele é longo o suficiente para abordar um material tão extenso de forma tão detalhada quanto ele merece.

No entanto, segue abaixo um resumo dos principais tópicos abordados neste documento:

- Não há uma solução mágica para pastas de trabalho ineficientes. Comece consultando o registrador de desempenho para saber onde o tempo está sendo gasto. Consultas demoradas? Excesso de consultas? Cálculos lentos? Renderização complexa? Use essas informações para concentrar seus esforços na direção certa.
- As recomendações neste documento são apenas isso: recomendações. Embora possam ser consideradas práticas recomendadas, é necessário testá-las para saber se elas de fato melhorarão o desempenho em seu caso específico. Muitas dessas recomendações podem variar de acordo com a estrutura dos seus dados e a fonte de dados que você está usando (por exemplo: arquivos simples, um RDBMS ou uma extração de dados).
- As extrações são uma forma rápida e fácil de agilizar a execução de pastas de trabalho.
- Quanto mais organizados seus dados estiverem e melhor corresponderem à estrutura de suas perguntas, exigindo o mínimo de preparação e manuseio, por exemplo, mais rápido suas pastas de trabalho serão executadas.
- Na maioria dos casos, a lentidão dos painéis se deve a um design ruim, especialmente quando há gráficos demais em um único painel ou ao tentar exibir muitos dados de uma só vez. Prefira a simplicidade. Permita que os usuários do seu painel detalhem os gráficos incrementalmente, em vez de tentar mostrar tudo ao mesmo tempo e depois filtrar os dados.
- Trabalhe apenas com os dados necessários, nada mais. Isso se aplica aos campos que você consulta e à granularidade dos registros retornados. Dessa forma, o Tableau é capaz de gerar consultas melhores, mais rápidas e em menor número, reduzindo a quantidade de dados que precisa ser movida da fonte de dados para o processador do Tableau. Isso também reduz o tamanho das suas pastas de trabalho, tornando-as mais rápidas de abrir e mais fáceis de compartilhar.
- Ao reduzir a quantidade de dados, verifique se os filtros estão sendo usados de forma eficaz.
- Strings e datas são lentas, números e booleanos são rápidos.

Algumas das recomendações fornecidas neste documento só terão um impacto significativo se você estiver trabalhando com conjuntos de dados grandes e/ou complexos. O que é um conjunto de dados grande ou complexo? Bem, isso depende... mas não custa nada seguir essas recomendações para todas as suas pastas de trabalho, porque nunca se sabe quando o volume de dados crescerá. A prática leva à perfeição.

Sumário

Sobre este documento.....	2
Muito longo, não li.....	3
Introdução.....	6
O que o Tableau faz bem?	6
O que o Tableau não faz bem?	6
Entendendo o conceito de eficiência.....	8
O que é uma pasta de trabalho “eficiente”?	8
Por que você deveria se preocupar com a eficiência?.....	8
As leis da física	9
Ferramentas úteis	11
Registrador de desempenho.....	11
Logs	13
Exibições de desempenho do Tableau Server	14
Monitoramento e testes	15
Outras ferramentas.....	15
O problema é o design da minha pasta de trabalho?.....	18
Um bom design de painel	18
Adapte seu painel para garantir um bom desempenho	21
Um bom design de planilha	27
Filtros eficientes.....	33
O problema são os meus cálculos?.....	43
Tipos de cálculo.....	44
Análises	48
Cálculos x recursos nativos	48
Impacto dos tipos de dados	49
Técnicas de desempenho.....	49
O problema são as minhas consultas?.....	54
Otimizações automáticas.....	54
Unições	61
Combinação	61
Integração de dados.....	65
SQL personalizado.....	67
Alternativas ao SQL personalizado	68
O problema são os meus dados?	70
Conselho geral	70

Fontes de dados	71
Preparação de dados	79
Extrações de dados	80
Governança de dados	87
O problema é o ambiente?	88
Atualização	88
Teste o Tableau Desktop no Tableau Server.....	88
Separe atualizações e cargas de trabalho interativas.....	88
Monitore e ajuste as configurações do Tableau Server.....	89
Infraestrutura.....	89
Conclusão.....	91

Introdução

O que o Tableau faz bem?

Na Tableau, buscamos mudar a forma como as pessoas veem, entendem e interagem com os dados. Consequentemente, não tentamos fornecer o mesmo tipo de experiência que as plataformas de BI empresariais tradicionais. O Tableau é uma excelente ferramenta para criar pastas de trabalho que são:

- **Visuais** – diversos estudos comprovam que a maneira mais eficiente de nós humanos entendermos conjuntos de dados grandes e complexos é por meio da representação visual. O comportamento padrão do Tableau é apresentar dados usando gráficos, diagramas e painéis. Ainda assim, tabelas e as tabelas de referência cruzada têm seu lugar (e são compatíveis). Falaremos mais sobre a melhor forma de usá-las posteriormente.
- **Interativas** – os documentos do Tableau são criados para oferecer interatividade aos usuários, seja em desktops, na Web ou em um dispositivo móvel. Diferentemente das outras ferramentas de BI, que geram um resultado para impressão (em papel ou em um formato eletrônico, como o PDF), o objetivo é criar experiências ricas e interativas para que os usuários possam explorar os dados e obter ajuda para responder às suas perguntas comerciais.
- **Iterativas** – a descoberta é um processo essencialmente cíclico. O Tableau foi criado para agilizar o ciclo pergunta-resposta-pergunta, para que os usuários possam rapidamente desenvolver uma hipótese, testá-la com os dados disponíveis, revisar essa hipótese, testá-la novamente, e assim por diante.
- **Rápidas** – o processo de BI é historicamente lento. Lento para instalar e configurar o software, lento para disponibilizar dados para análise e lento para criar e implementar documentos, relatórios, painéis, etc. Com o Tableau, os usuários podem instalar a ferramenta, se conectar e gerar documentos mais rápido do que nunca, em muitos casos reduzindo o tempo de resposta de meses ou semanas para horas ou minutos.
- **Simples** – as ferramentas corporativas tradicionais de BI muitas vezes estão além dos recursos da maioria dos usuários comerciais, em termos de custo ou complexidade. Com frequência, os usuários precisam da ajuda da equipe de TI ou de um usuário avançado para criar as consultas e os documentos de que precisam. O Tableau oferece uma interface intuitiva para que usuários sem conhecimentos técnicos possam consultar e analisar dados complexos sem que precisem se tornar especialistas em bancos de dados ou planilhas.
- **Bonitas** – dizem que a beleza está nos olhos de quem vê, mas, quando se trata de comunicação visual, é preciso seguir as práticas recomendadas. Com recursos como o “Mostre-me”, o Tableau orienta usuários sem conhecimentos técnicos na criação de gráficos inteligíveis e eficazes com base nos dados que estão sendo usados.
- **Onipresentes** – cada vez mais os usuários estão criando documentos para várias plataformas de distribuição. Os usuários precisam exibir dados e interagir com eles em seus computadores, na Web, em dispositivos móveis, incorporados em outros aplicativos e documentos, etc. O Tableau permite que um único documento seja publicado e, em seguida, utilizado em todas essas plataformas sem a necessidade de migração ou reformulação.

O que o Tableau não faz bem?

O Tableau é uma ferramenta poderosa e com muitos recursos, mas é importante entender primeiro que, para alguns problemas, ele talvez não seja a melhor solução. Isso não significa que ele não possa resolver esses problemas; o Tableau pode ser utilizado para realizar muitas tarefas que não fazem parte de sua especificação de design original. Quero dizer que esses não são os tipos de problemas que o Tableau foi desenvolvido para resolver. Por isso, se você escolher usá-lo, a proporção entre esforço e benefício provavelmente será desfavorável, e a solução pode ter um desempenho ruim ou não ser flexível.

Sugerimos que você reveja seus requisitos ou considere outra abordagem se:

- Você precisar de um documento que tenha sido criado para ser impresso, e não exibido na tela. Por exemplo, se você precisa controlar layouts de página complexos, se precisa de recursos como cabeçalhos/rodapés de página, de seção e de grupos ou de uma formatação WYSIWYG (o formato exibido é o resultado final). O Tableau pode gerar relatórios com várias páginas, mas eles não terão a formatação em faixas oferecida pelas ferramentas de relatórios dedicadas.
- Você precisar de um mecanismo de envio e entrega complexo para os documentos personalizados (também chamado de “bursting”) enviados por meio de vários modos de entrega. O Tableau Server inclui o conceito de assinaturas de relatório, permitindo que um usuário assine um relatório para recebê-lo por e-mail (no Tableau 10, também é possível fazer uma assinatura para outras pessoas). No entanto, há casos em que os clientes preferem uma solução mais flexível. O Tableau pode ser usado para criar outros tipos de sistemas de envio e entrega, mas esse não é um recurso nativo do Tableau. É necessário desenvolver uma solução personalizada com o utilitário TABCMD, ou introduzir soluções de terceiros, como o *VizAlerts* (<http://tabsoft.co/1stldFh>) ou o *Push Intelligence for Tableau* da Metric Insights (<http://bit.ly/1HACxul>).
- O principal caso de uso para o leitor é exportar os dados para outro formato (geralmente um arquivo .csv ou do Excel). Isso normalmente gera um relatório em forma de tabela com muitas linhas de dados detalhados. Ou seja, o Tableau permite que os usuários exportem dados de uma exibição ou de um painel para o Excel, estejam eles resumidos ou detalhados. No entanto, quando o principal caso de uso é a exportação de dados, isso significa que a exportação é um substituto para o processo de ETL (extração, transformação e carregamento). Há outras soluções muito mais eficientes do que uma ferramenta de relatórios para executar essa operação.
- Você precisa de documentos de tabela de referência cruzada extremamente complexos que espelhem os relatórios de planilha existentes com cálculos complexos de subtotal, referência cruzada, etc. Exemplos comuns disso são: relatórios financeiros, como lucros e perdas, balanço, etc. Em algumas situações, pode ser necessário fazer uma modelagem de cenário, uma análise de cenários hipotéticos (what-if) e um write-back de dados presumidos. Se os dados granulares subjacentes não estiverem disponíveis, ou se a lógica do relatório estiver baseada em “referências de célula”, e não em acumular os totais dos registros, talvez o melhor seja continuar usando uma planilha para esse tipo de relatório.

Entendendo o conceito de eficiência

O que é uma pasta de trabalho “eficiente”?

São vários os fatores que tornam uma pasta de trabalho “eficiente”. Alguns deles são técnicos e outros mais direcionados aos usuários, mas, geralmente, uma pasta de trabalho eficiente é:

- **Simples** – a pasta de trabalho é fácil de criar e será fácil de manter no futuro? Ela aproveita os princípios da análise visual para comunicar claramente a mensagem de seu autor e dos dados?
- **Flexível** – a pasta de trabalho é capaz de responder às várias perguntas que os usuários desejam fazer ou apenas a uma pergunta? Ela envolve o usuário em uma experiência interativa ou é apenas um relatório estático?
- **Rápida** – a pasta de trabalho responde às perguntas dos usuários com a agilidade necessária? Isso inclui o tempo de abertura, de atualização e de resposta à interação. Essa avaliação pode ser um pouco subjetiva, mas, em geral, queremos que a pasta de trabalho forneça uma exibição de informações inicial e responda às interações do usuário em segundos.

O desempenho de um painel é afetado pelo seguinte:

- O design visual do painel e da planilha. Por exemplo, a quantidade de elementos, a quantidade de pontos de dados, o uso de filtros e ações, etc.
- Os tipos de cálculos, onde eles são executados, etc.
- As consultas, incluindo a quantidade de dados retornada, se são um SQL personalizado.
- As conexões de dados e as fontes de dados subjacentes.
- Algumas diferenças entre o Tableau Desktop e o Tableau Server.
- Outros fatores do ambiente, como a configuração e a capacidade do hardware.

Por que você deveria se preocupar com a eficiência?

Por diversos motivos:

- Poder trabalhar de forma eficiente como um analista ou autor de pasta de trabalho permite que você obtenha respostas com mais rapidez.
- Trabalhar de forma eficiente ajuda a manter você no “fluxo” de análise. Isso significa que você está concentrado nas perguntas e nos dados, em vez de se preocupar em aprender a usar a ferramenta para obter um resultado.
- Criar uma pasta de trabalho com um design flexível reduz a necessidade de criar e manter diversas pastas de trabalho que atendem a requisitos semelhantes.
- Criar uma pasta de trabalho com um design simples permite que outras pessoas a utilizem com facilidade para fazer novas iterações com os dados.
- A percepção de responsividade é um fator de sucesso importante para os usuários finais quando eles visualizam relatórios e painéis. Por isso, agilizar ao máximo a execução de suas pastas de trabalho deixa os usuários mais satisfeitos.

Nossa vasta experiência nos mostra que a maioria dos problemas de desempenho encontrados por clientes são consequência de erros no design da pasta de trabalho. Se pudermos corrigir esses erros, ou melhor, evitar que eles aconteçam oferecendo um treinamento adequado, então poderemos resolver esses problemas.

Se você estiver trabalhando com pequenos volumes de dados, muitas dessas recomendações não serão essenciais. Você poderá resolver o problema pontualmente. Porém, quando trabalhamos com centenas de milhões de registros, várias pastas de trabalho ou diversos autores, os efeitos de uma

pastas de trabalho mal feitas são potencializadas, e esse é um bom motivo para você considerar as orientações fornecidas neste whitepaper.

Mas, naturalmente, a prática leva à perfeição e é recomendável seguir essas orientações ao criar todas as pastas de trabalho. Lembre-se de que seu design não estará pronto até você testar sua pasta de trabalho com os volumes de dados de produção esperados.

Uma observação importante: neste documento, nos referimos ao Tableau Server, mas, em muitas ocasiões, as orientações fornecidas aqui também se aplicarão ao Tableau Online, caso você prefira usar a nossa solução hospedada no lugar de uma implantação local. As exceções óbvias são os pontos em que abordamos as adaptações ou os ajustes dos parâmetros de configuração do Tableau Server e a instalação/atualização do software na camada do servidor. No mundo do SaaS, você não precisa se preocupar com essas coisas!

As leis da física

Antes de entrarmos nos detalhes técnicos de como vários recursos afetam o desempenho das pastas de trabalho, veremos alguns princípios básicos que ajudarão você a criar painéis e relatórios eficientes:

Se é lento na fonte de dados, será lento no Tableau

Se sua pasta de trabalho do Tableau for baseada em uma consulta de execução lenta, sua pasta de trabalho também será lenta. Nas seções a seguir, identificaremos dicas de ajustes para seus bancos de dados que podem melhorar o tempo de execução das consultas. Além disso, discutiremos como o processador de dados rápido do Tableau pode ser usado para melhorar o desempenho das consultas.

Se é lento no Tableau Desktop, será (quase sempre) lento no Tableau Server

Uma pasta de trabalho com desempenho ruim no Tableau Desktop não será mais rápida depois de publicada no Tableau Server. Os usuários costumam achar que suas pastas de trabalho serão executadas com mais rapidez no Tableau Server, porque ele tem mais CPU, RAM, etc. do que seus computadores locais. Em geral, as pastas de trabalho são um pouco mais lentas no Tableau Server, porque:

- Há muitos usuários compartilhando os recursos do Tableau Server para gerar pastas de trabalho simultaneamente (embora você acredite que sua pasta de trabalho terá um tempo de resposta menor depois de compartilhá-la, devido aos mecanismos de armazenamento em cache do Tableau Server, não é isso o que acontece).
- O Tableau Server precisa trabalhar mais para renderizar os painéis e gráficos do que uma estação de trabalho cliente.

Seus esforços iniciais devem se concentrar em ajustar a pasta de trabalho no Tableau Desktop antes de ajustar seu desempenho no Tableau Server.

A exceção a essa regra ocorre quando o Tableau Desktop encontra limitações devido a recursos ausentes no Tableau Server. Por exemplo, se o computador não tiver memória RAM suficiente para suportar o volume de dados que você está analisando ou se o Tableau Server tiver uma conexão de latência mais rápida/lenta com a fonte de dados. Alguns usuários enfrentam problemas de lentidão e até mesmo recebem a mensagem de erro “memória insuficiente” quando utilizam um conjunto de dados em suas estações de trabalho com configurações baixas e 2 GB de RAM, mas consideram o desempenho da pasta de trabalho publicada aceitável, porque o servidor tem muito mais memória e capacidade de processamento.

Quanto mais novo melhor

A equipe de desenvolvimento da Tableau trabalha constantemente para melhorar o desempenho e a usabilidade dos softwares da empresa. Atualizar para a versão mais recente do Tableau Desktop

e do Tableau Server pode resultar em melhorias significativas no desempenho do software, eliminando a necessidade de fazer alterações na pasta de trabalho. Muitos de nossos clientes relataram que suas pastas de trabalho apresentaram um desempenho três vezes melhor, ou até superior a isso, depois que atualizaram da versão 8 para a versão 9. O aprimoramento do desempenho é o foco do Tableau 10 e continuará sendo em suas próximas versões. Se você usa o Tableau Online, não precisa se preocupar com isso, uma vez que ele está sempre sendo atualizado para as versões mais recentes assim que elas são disponibilizadas.

Isso se aplica às versões de manutenção e às versões principais e secundárias do software. Para obter mais informações sobre o ciclo de versões do Tableau e detalhes específicos de cada uma delas, visite a página Notas de versão em:

<http://www.tableau.com/pt-br/support/releases>

Além disso, fornecedores de bancos de dados estão trabalhando para aperfeiçoar seus produtos. Certifique-se de que você sempre esteja usando a versão mais recente do driver apropriado para sua fonte de dados, conforme listado nesta página da Web:

<http://www.tableau.com/pt-br/support/drivers>

Menos é mais:

Como tudo na vida, em excesso, até mesmo o que é bom se torna ruim. Não tente colocar tudo em uma única pasta de trabalho. Embora seja *possível ter* 50 painéis, cada um com 20 objetos de gráfico, conectados a 50 fontes de dados diferentes em uma pasta de trabalho do Tableau, o desempenho dela certamente será muito ruim.

Se você tiver uma pasta de trabalho como essa, transforme-a em vários arquivos separados. É muito fácil. Você só precisa copiar os painéis de uma pasta de trabalho para outra, e o Tableau exibirá todas as pastas de trabalho e fontes de dados associadas. Se seus painéis forem muito complexos, simplifique-os usando interações para orientar os usuários finais de um relatório para o outro. Lembre-se de que o preço do software não é baseado no número de documentos, então, espalhe um pouco os dados.

Escalabilidade não é o mesmo que desempenho

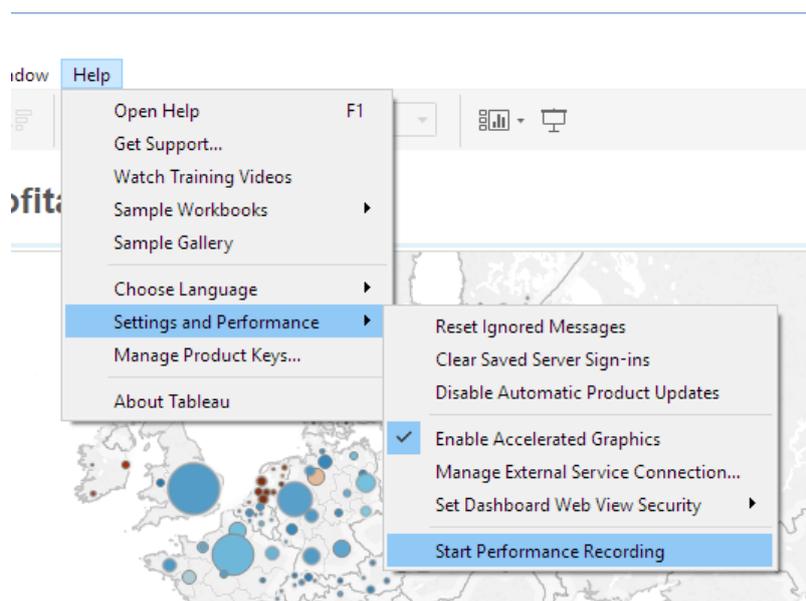
A escalabilidade consiste na nossa capacidade de permitir que diversos usuários exibam pastas de trabalho compartilhadas ao mesmo tempo. Já o desempenho consiste na nossa capacidade de garantir que uma pasta de trabalho tenha o tempo de execução mais rápido possível. Apesar de muitas das recomendações fornecidas neste documento influenciarem de forma positiva a escalabilidade das pastas de trabalho publicadas no Tableau Server, o objetivo principal deste documento é ajudar você a melhorar o desempenho do software.

Ferramentas úteis

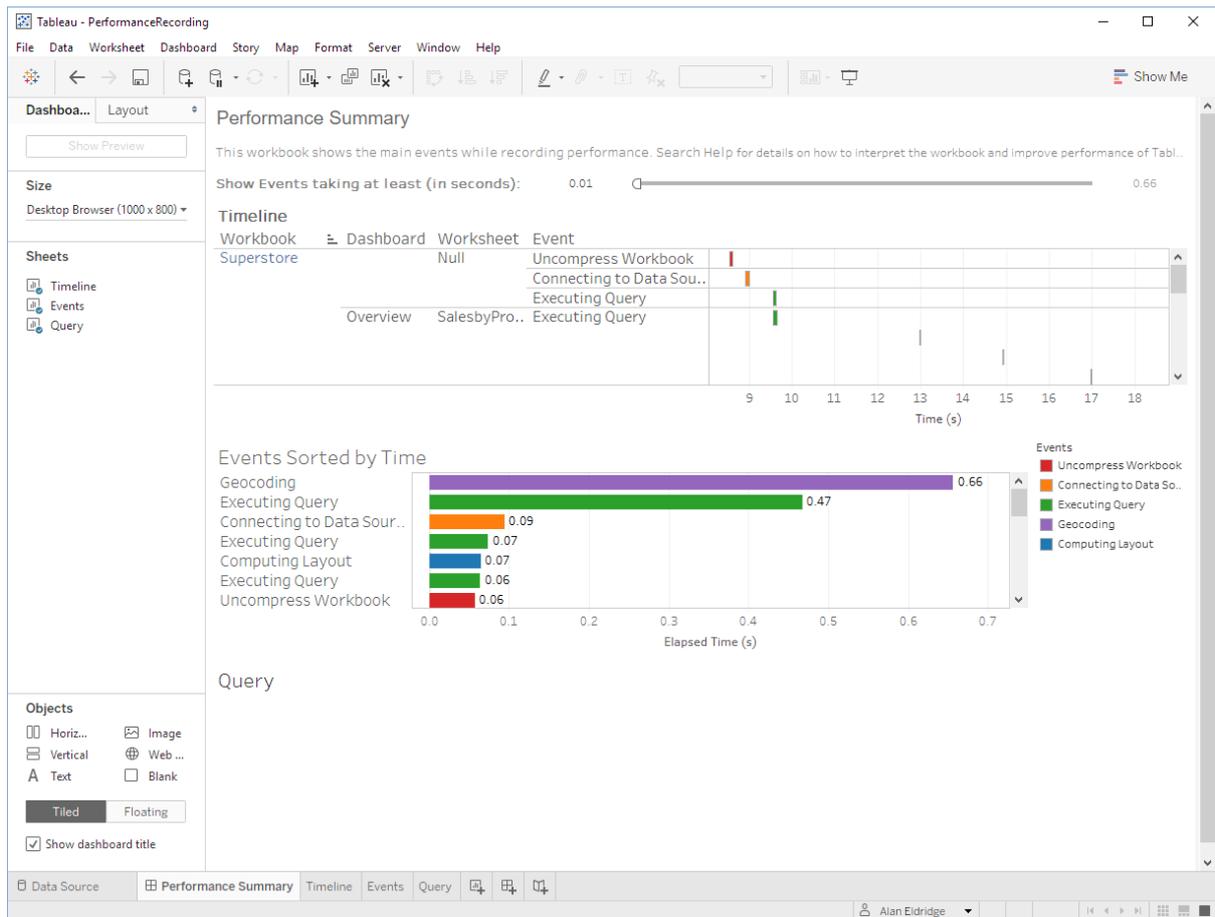
Para entender melhor o desempenho de suas pastas de trabalho, você precisa saber: o que está acontecendo e quanto tempo isso está tomando. Você pode encontrar essas informações em vários lugares do software, dependendo de onde estiver executando sua pasta de trabalho (no Tableau Desktop ou no Tableau Server) e com diversos níveis de detalhamento. Esta seção descreve todas as opções disponíveis.

Registrador de desempenho

O primeiro lugar em que você deve procurar informações de desempenho é no recurso Registrador de desempenho do Tableau Desktop e do Tableau Server. No Tableau Desktop, esse recurso pode ser habilitado no menu Ajuda:



Abra o Tableau e inicie a gravação de desempenho. Em seguida, abra sua pasta de trabalho (é uma prática recomendada não ter outras pastas de trabalho abertas enquanto você estiver registrando o desempenho para não haja uma competição desnecessária por recursos). Interaja com a pasta de trabalho como se você fosse um usuário final e, quando achar que coletou dados suficientes, volte ao menu Ajuda e pare a gravação. Outra janela do Tableau Desktop será aberta e mostrará os dados capturados.



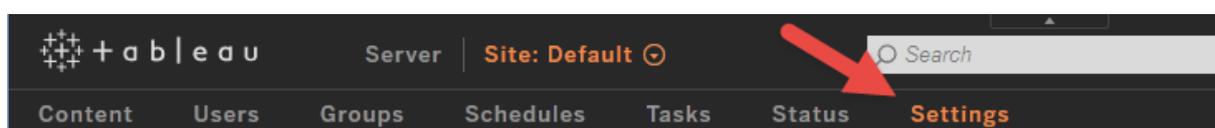
Agora, você pode identificar as ações mais demoradas na pasta de trabalho. Por exemplo, na imagem acima, a consulta selecionada na planilha “Timeline” demorou 30,66 segundos para ser concluída. Clicar na barra mostra o texto da consulta que está sendo executada. Como resultado do registro de desempenho, uma pasta de trabalho do Tableau é criada, e você pode adicionar outras exibições a ela para explorar essas informações de outras formas.

Observação: por padrão, eventos com duração inferior a 0,1 segundo não são mostrados. Você pode alterar a configuração para incluir esses eventos ajustando o filtro na parte superior do painel, mas seria mais interessante se concentrar nas tarefas com execuções mais demoradas. É uma boa prática definir o filtro para 1 segundo.

Também é possível criar registros de desempenho no Tableau Server para ajudar a identificar problemas que surgem no momento da publicação da pasta de trabalho. Por padrão, o registro de desempenho não está habilitado no Tableau Server. Esse recurso pode ser gerenciado em cada site.

O administrador do sistema pode habilitar o registro de desempenho site por site.

1. Acesse o site no qual você deseja habilitar o registro de desempenho.
2. Clique em **Configurações**:



3. Em Métricas de desempenho da pasta de trabalho, selecione **Gravar métricas de desempenho da pasta de trabalho**.

4. Clique em **Salvar**.

Se você quiser criar um registro de desempenho:

1. Abra a exibição da qual deseja registrar o desempenho. Quando você abre uma exibição, o Tableau Server adiciona ":iid=<n>" ao final da URL, que indica a ID da sessão. Por exemplo:

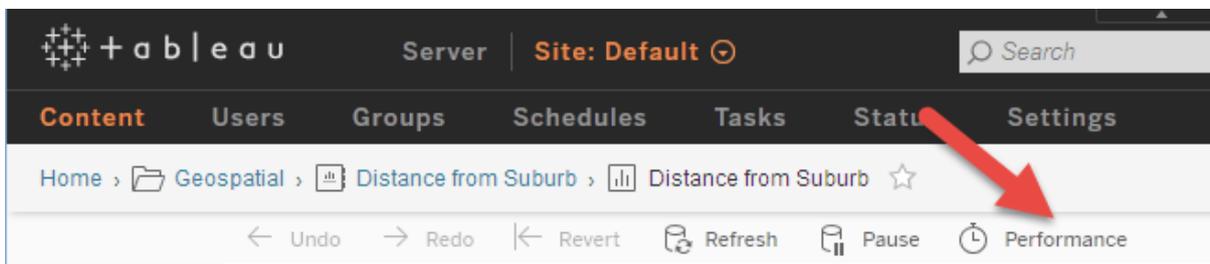
```
http://<tableau_server>/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?iid=1
```

2. Digite `:record_performance=yes&` ao final da URL da exibição, imediatamente antes da ID da sessão. Por exemplo:

```
http://<tableau_server>/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?record_performance=yes&iid=1
```

3. Carregue a exibição.

Uma indicação visual de que o registro de desempenho foi iniciado é a opção **Desempenho** mostrada na barra de ferramentas da exibição:



Quando terminar e quiser ver o registro de desempenho, faça o seguinte:

1. Clique em **Desempenho** para abrir a pasta de trabalho de desempenho. Essa pasta de trabalho exibe um instantâneo dos dados de desempenho atuais. Você pode capturar outros instantâneos enquanto estiver trabalhando com a exibição. Os dados de desempenho são cumulativos.
2. Vá para outra página ou remova `:record_performance=yes` da URL para parar a gravação.

Use essas informações para identificar as seções de uma pasta de trabalho que devem ser revistas. Por exemplo, onde você pode melhorar ao máximo o tempo gasto? Para obter mais informações sobre como interpretar esses registros, clique no link abaixo:

http://onlinehelp.tableau.com/current/pro/desktop/pt-br/help.htm#perf_record_create_desktop.html

Logs

No Tableau, você pode encontrar o texto completo da consulta pesquisando no arquivo de log.

O local padrão é `C:\Users\<username>\Documents\My Tableau Repository\Logs\log.txt`.

Esse arquivo é bastante extenso e está escrito em texto codificado JSON. Use um bom editor de texto, como o Notepad++ ou o Sublime, para abri-lo. Se procurar por "begin-query" ou "end-query", você encontrará a string da consulta que foi transmitida para a fonte de dados.

O registro "end-query" do log também mostra o tempo gasto com a execução da consulta e quantos registros foram retornados para o Tableau:

```
{ "ts": "2015-05-24T12:25:41.226", "pid": 6460, "tid": "1674", "sev": "info", "req": "-", "sess": "-", "site": "-", "user": "-", "k": "end-query", "v": { "protocol": "4308fb0", "cols": 4, "query": "SELECT
```

```

[DimProductCategory].[ProductCategoryName] AS
[none:ProductCategoryName:nk],\n
[DimProductSubcategory].[ProductSubcategoryName] AS
[none:ProductSubcategoryName:nk],\n SUM(CAST(([FactSales].[ReturnQuantity])
as BIGINT)) AS [sum:ReturnQuantity:ok],\n SUM([FactSales].[SalesAmount])AS
[sum:SalesAmount:ok]\nFROM [dbo].[FactSales] [FactSales]\n INNER JOIN
[dbo].[DimProduct] [DimProduct] ON ([FactSales].[ProductKey] =
[DimProduct].[ProductKey])\n INNER JOIN [dbo].[DimProductSubcategory]
[DimProductSubcategory] ON ([DimProduct].[ProductSubcategoryKey] =
[DimProductSubcategory].[ProductSubcategoryKey])\n INNER JOIN
[dbo].[DimProductCategory] [DimProductCategory] ON
([DimProductSubcategory].[ProductCategoryKey] =
[DimProductCategory].[ProductCategoryKey])\nGROUP BY
[DimProductCategory].[ProductCategoryName],\n
[DimProductSubcategory].[ProductSubcategoryName]", "rows":32,"elapsed":0.951}
}

```

Se você estiver usando o Tableau Server, os logs estarão em C:\ProgramData\Tableau\Tableau Server\data\tabsvc\vizqlserver\Loggs.

Exibições de desempenho do Tableau Server

O Tableau Server é fornecido com várias exibições para administradores, criadas para ajudar no monitoramento da atividade do software. As exibições estão localizadas na tabela Análise da página Manutenção do Tableau Server:

Analysis	
Dashboards that monitor Tableau Server activity.	
Dashboard	Analysis
Traffic to Sheets	Usage and users for published sheets.
Traffic to Data Sources	Usage and users for published data sources.
Actions by All Users	Actions for all users.
Actions by Specific User	Actions for a specific user, including items used.
Actions by Recent Users	Recent actions by users, including last action time and idle time.
Background Tasks for Extracts	Completed and pending extract task details.
Background Tasks for Non Extracts	Completed and pending background task details (non-extract).
Stats for Load Times	Sheet load times and performance history.
Stats for Space Usage	Space used by published workbooks and data sources, including extracts and live connections.
Server Disk Space	Current and historical disk space usage, by server node.
Tableau Desktop License Usage	Summary of usage for Tableau Desktop licenses
Tableau Desktop License Expirations	Expiration information for Tableau Desktop licenses

Para obter mais informações sobre essas exibições, clique no link abaixo:

<http://onlinehelp.tableau.com/current/server/pt-br/adminview.htm>

Também é possível criar exibições administrativas conectando o banco de dados PostgreSQL que faz parte do repositório do Tableau. As instruções podem ser encontradas aqui:

http://onlinehelp.tableau.com/current/server/pt-br/adminview_postgres.htm

Monitoramento e testes

TabMon

O TabMon é um programa de monitoramento de cluster de código-fonte aberto desenvolvido para o Tableau Server, que permite coletar estatísticas de desempenho ao longo do tempo. O suporte do TabMon é fornecido pela comunidade, e estamos disponibilizando o código-fonte completo por meio da licença de software livre do MIT.

O TabMon registra a integridade do sistema e as métricas de aplicativo sem necessidade de configuração. Ele coleta métricas internas da mesma forma que o Monitor de Desempenho (Perfmon) do Windows, o Java Health e os contadores do Java Mbean (JMX) nos computadores com o Tableau Server instalado pertencentes a uma mesma rede. Você pode usar o TabMon para monitorar o uso de recursos físicos (CPU, RAM), da rede e da unidade de disco rígido, bem como para rastrear a taxa de acertos do cache, a latência da solicitação, as sessões ativas e muito mais. Ele mostra os dados em uma estrutura clara e unificada, facilitando a visualização dos dados no Tableau Desktop.

O TabMon permite que você tenha controle total sobre quais métricas coletar e quais computadores monitorar, sem precisar escrever scripts ou códigos. Você só precisa saber o nome da máquina e da métrica. O TabMon pode ser executado remotamente e independentemente do seu cluster. Você pode monitorar, agregar e analisar a integridade dos seus clusters em qualquer computador da sua rede sem adicionar praticamente nenhuma carga aos computadores de produção.

Encontre mais informações sobre o TabMon aqui:

<http://bit.ly/1ULFelf>

TabJolt

O TabJolt é uma ferramenta de testes de desempenho e carga “apontar-e-executar”, desenvolvida especificamente para funcionar de forma integrada com o Tableau Server. Diferentemente das ferramentas de testes de carga tradicionais, o TabJolt direciona a carga automaticamente para a sua instância do Tableau Server, sem que seja necessário criar ou fazer a manutenção de scripts. Como o TabJolt conhece o modelo de apresentação do Tableau, ele consegue carregar visualizações automaticamente e interpretar possíveis interações durante a execução do teste.

Tudo o que você precisa fazer é apontar o TabJolt para uma ou mais pastas de trabalho no Tableau Server e carregar e executar automaticamente as interações nas exibições do Tableau. O TabJolt também coleta as métricas principais, como o tempo médio de resposta, a taxa de transferência e o tempo de resposta do 95º percentil, e ainda captura as métricas de desempenho do Windows para correlacioná-las.

Mesmo com a ajuda do TabJolt, é preciso que os usuários tenham conhecimentos suficientes sobre a arquitetura do Tableau Server. Não é recomendável tratar o Tableau Server como uma caixa preta de testes de carga. Isso pode produzir resultados que frustrarão suas expectativas. O TabJolt é uma ferramenta automatizada e não pode replicar com facilidade todas as interações humanas possíveis. Por isso, avalie cuidadosamente se os resultados do TabJolt refletem a realidade.

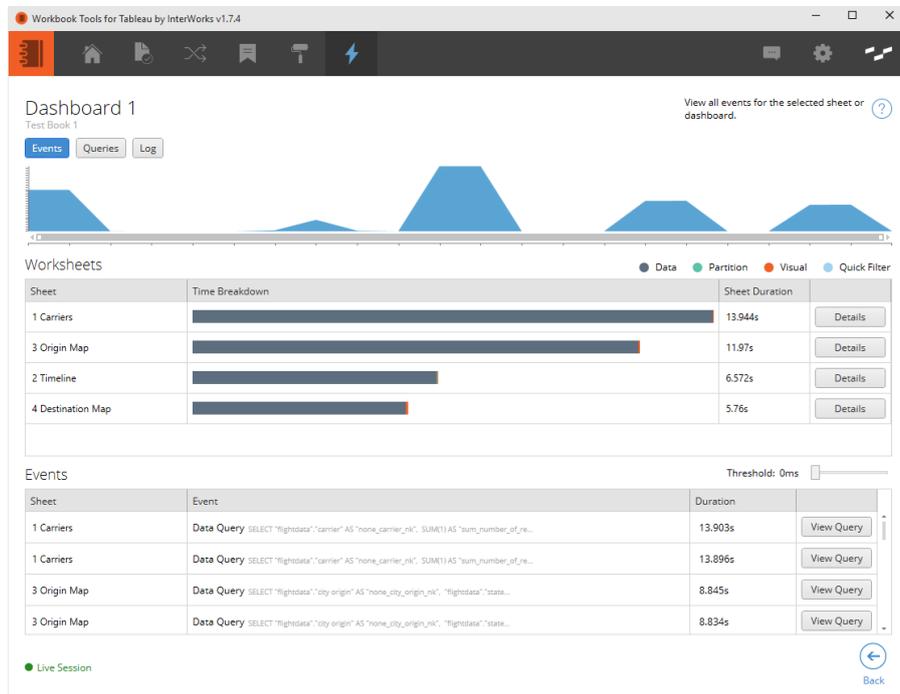
Encontre mais informações sobre o TabJolt aqui:

<http://bit.ly/1ULFtgi>

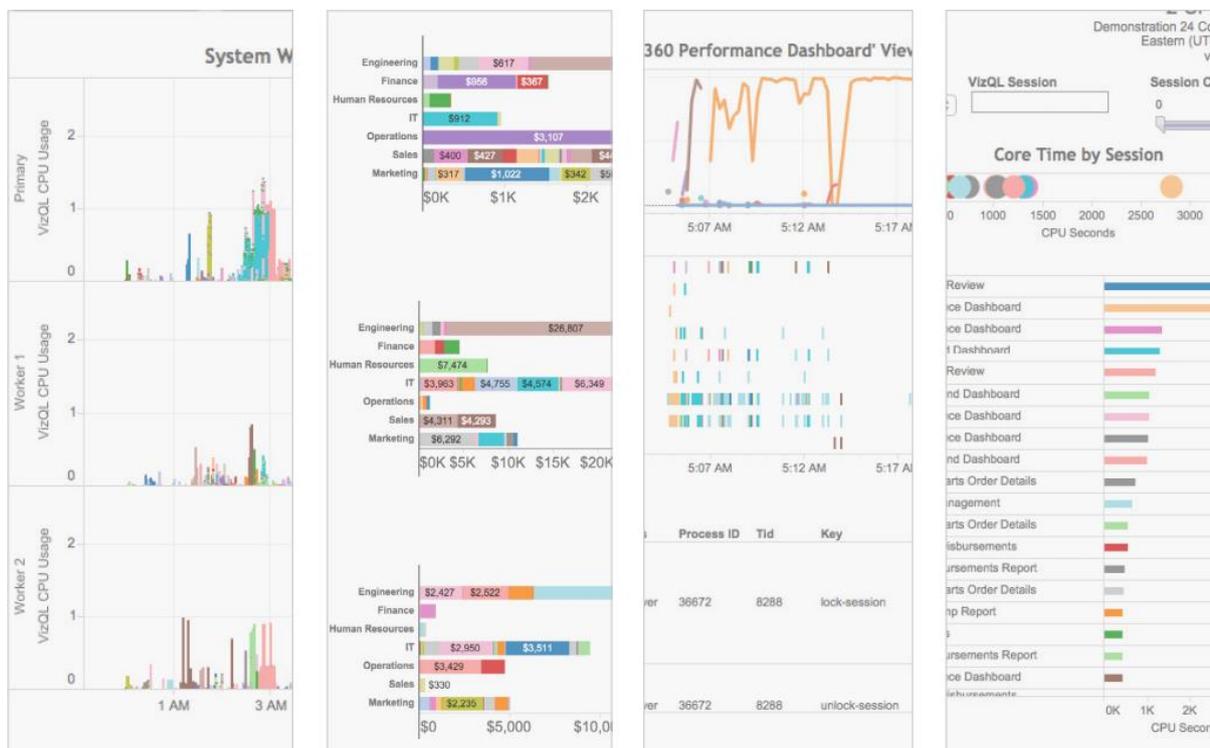
Outras ferramentas

Existem outras ferramentas de terceiros disponíveis que podem ajudar você a identificar características de desempenho de suas pastas de trabalho. Uma delas é o pacote “Power Tools for Tableau”,

desenvolvido pela Interworks, que inclui um analisador de desempenho (semelhante ao registro de desempenho integrado do Tableau) que permite detalhar e determinar quais planilhas e consultas têm o maior tempo de resposta:



A Palette Software também tem um produto chamado Palette Insight que captura informações de desempenho do Tableau Server e permite criar um planejamento de capacidade, identificar usuários e pastas de trabalho que utilizam muitos recursos, fazer auditoria do acesso dos usuários e criar modelos de reembolso.



Além disso, a maioria das plataformas de DBMS modernas oferecem ferramentas administrativas que permitem rastrear e analisar consultas em execução. Seu DBA pode ser de grande ajuda se seus registros de desempenho indicarem o tempo de execução da consulta como um fator principal.

Se você acredita que o problema está na interação do navegador do cliente com o Tableau Server, use ferramentas como o Telerik Fiddler ou as ferramentas de desenvolvedor do seu navegador para analisar mais a fundo o tráfego entre o cliente e o Tableau Server.

O problema é o design da minha pasta de trabalho?

Trabalhar no Tableau é uma experiência nova para muitos usuários, e é preciso conhecer técnicas e práticas recomendadas de design para criar pastas de trabalho eficientes. Ainda encontramos muitos usuários que tentam aplicar abordagens antigas de design com o Tableau e, portanto, acabam por obter resultados não muito bons. O objetivo desta seção é abordar alguns princípios de design que refletem as práticas recomendadas.

Um bom design de painel

Com o Tableau, você cria uma experiência interativa para seus usuários. O resultado final fornecido pelo Tableau Server é um aplicativo interativo que permite aos usuários explorar os dados, e não apenas exibi-los. Então, para criar um painel eficiente do Tableau, você precisa parar de pensar que está desenvolvendo um relatório estático.

Este é um exemplo de painel que muitos dos novos autores criam, principalmente se eles costumavam trabalhar com ferramentas como o Excel ou o Access, ou usar ferramentas de geração de relatório “tradicionais”. Temos aqui um relatório no formato de tabela que mostra “tudo” e diversos filtros para os usuários refinarem a tabela até que ela exiba os registros nos quais os usuários estão interessados:

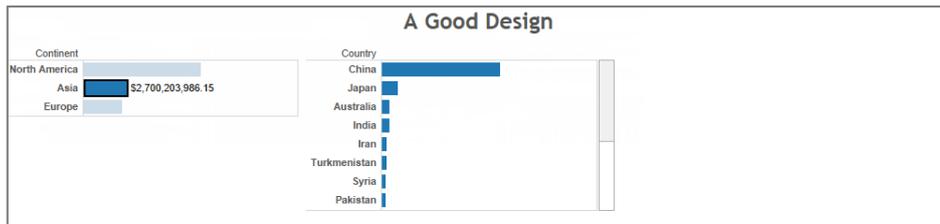
A Bad Design									
Continent	Country	State/Province	Product Category	Product Subcate..	Product Name	Sales Qty	Total Cost	Sales Amou..	
Asia	Turkmenistan	Ahal Province	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	64	\$5,547.52	\$11,790.68	
North America	United States	Alaska	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	62	\$5,200.80	\$11,536.20	
North America	Canada	Alberta	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	40	\$3,467.20	\$7,540.00	
Europe	France	Alpes-Maritim.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	31	\$2,687.08	\$5,734.17	
Asia	Armenia	Armenia	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	19	\$1,560.24	\$3,468.40	
Europe	France	Bas-Rhin	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	36	\$3,033.80	\$6,710.60	
Europe	Germany	Bavaria	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	60	\$5,114.12	\$10,819.90	
Asia	China	Beijing	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	2,276	\$194,683.28	\$421,825.30	
Europe	Germany	Berlin	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	1,193	\$102,022.36	\$220,330.11	
Europe	Switzerland	Bern	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	38	\$3,207.16	\$6,936.80	
North America	Canada	British Colum.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	107	\$9,274.76	\$20,075.25	
Europe	Romania	Bucuresti	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	10	\$866.80	\$1,885.00	
Europe	Greece	Central Greec.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	18	\$1,560.24	\$3,317.60	
Asia	Japan	Chubu	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	13	\$1,040.16	\$2,337.40	
Asia	Kyrgyzstan	Chuy Province	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	66	\$5,547.52	\$12,280.78	
North America	United States	Colorado	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	699	\$59,635.84	\$130,093.28	
North America	United States	Connecticut	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	293	\$25,050.52	\$54,533.05	
Asia	Syria	Damascus	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	104	\$8,928.04	\$19,311.83	
Europe	United Kingdo.	England	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	478	\$40,912.96	\$88,702.45	
North America	United States	Florida	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	423	\$35,885.52	\$78,745.88	
Europe	Germany	Hesse	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	26	\$2,253.68	\$4,674.80	
Asia	Japan	Hokkaido	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	22	\$1,906.96	\$4,109.30	
Asia	China	Hong Kong	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	111	\$9,448.12	\$20,574.78	
Asia	Pakistan	Islamabad Ca.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	63	\$5,287.48	\$11,724.70	
Asia	Japan	Kansai	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	119	\$10,228.24	\$22,158.18	
Asia	Japan	Kanto	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	176	\$15,255.68	\$32,648.20	
Asia	Thailand	Krung Thep	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	89	\$7,541.16	\$16,522.03	
Europe	Ireland	Leinster	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	31	\$2,687.08	\$5,824.65	

Este não é um “bom” painel do Tableau (na verdade, não é um “bom” painel de jeito algum). Na pior das hipóteses, ele é um processo consagrado de extração de dados, porque o usuário quer usar os dados em outra ferramenta, como o Excel, para fazer análises e gerar gráficos. Na melhor das hipóteses, ele indica que não sabemos ao certo como o usuário final deseja explorar os dados, então, adotamos a abordagem “com base em seus critérios iniciais, todos os dados estão aqui... incluí alguns filtros para que você possa refinar os dados e encontrar o que deseja”.

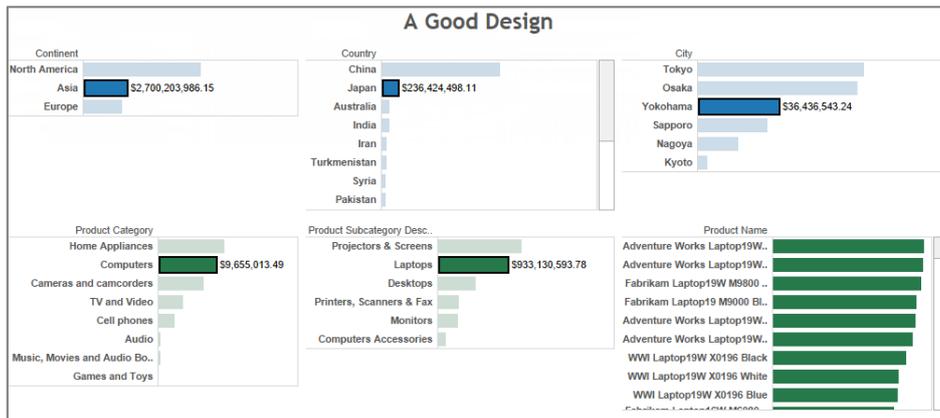
Veja agora a nova exibição, com exatamente os mesmos dados. Começamos aqui no nível mais alto de agregação:



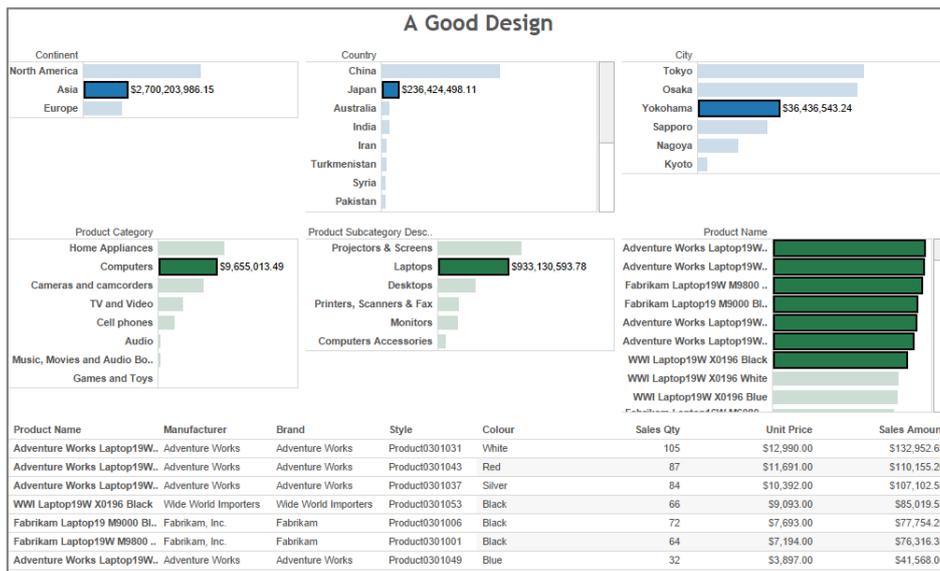
Se selecionar um ou mais elementos, você poderá ver os detalhes:



Podemos fazer isso várias vezes para detalhar ainda mais:



Até chegarmos ao último nível: os mesmos dados mostrados no painel da tabela de referência cruzada acima.



Não se concentre agora na apresentação dos dados (esse é um tópico importante, mas será abordado depois). Em vez disso, pense na experiência de usar esse painel. Observe como ele segue um caminho natural, da esquerda para a direita e de cima para baixo. Mesmo com muitos dados subjacentes nesse exemplo, o painel orienta o usuário final a detalhar gradualmente até encontrar o conjunto de registros detalhados que deseja.

A principal diferença entre os dois exemplos é como eles orientam o usuário final pelo processo de análise. O primeiro exemplo começa amplo (mostrando todos os registros que podem ser analisados) e permite que o usuário final reduza o número de registros exibidos por meio da utilização dos filtros. Essa técnica apresenta alguns problemas:

- A consulta inicial que deve ser executada antes de tudo é mostrada ao usuário final como a maior consulta possível: “mostre todos os registros”. Em qualquer conjunto de dados real, essa consulta demoraria muito tempo para ser executada e transmitida de volta ao processador do Tableau. A experiência do “primeiro contato” é essencial para definir a percepção da solução pelo usuário final e, se ela demorar muito para processar algo, a percepção será negativa.
- Criar uma exibição com centenas de milhares de marcas (cada célula de uma tabela de referência cruzada é chamada de marca) exige muito da CPU e da memória. E também leva tempo, aumentando a percepção negativa sobre a capacidade de resposta do sistema. No Tableau Server, ter muitas pessoas gerando tabelas de referência cruzada pode resultar em um desempenho ruim e, até mesmo, deixar o sistema sem memória. Isso pode resultar em problemas de estabilidade no servidor, erros e diversos tipos de experiências desagradáveis para os usuários finais. É claro que você pode adicionar mais memória ao servidor para minimizar isso, mas estaria tratando o sintoma, não a causa.
- Além disso tudo, os usuários não conseguem saber pelo contexto se o conjunto de filtros inicial será muito abrangente ou muito restrito. Como um usuário do relatório saberá se, ao marcar todas as categorias disponíveis, sua consulta retornará dezenas de milhares de registros e consumirá toda a memória RAM disponível no servidor? É impossível, ele só saberá se passar por essa má experiência.

Compare isso com a segunda abordagem, em que a consulta inicial mostra apenas o nível mais alto de agregação:

- A consulta inicial que deve ser executada está extremamente agregada e retorna apenas alguns registros. Para um banco de dados bem feito, essa é uma atividade bastante eficiente, e a resposta do “primeiro contato” é muito rápida, proporcionando uma experiência positiva com o sistema. À medida que detalhamos, cada consulta subsequente é agregada e restringida pelas seleções do nível mais alto. As consultas continuam sendo rápidas e retornam com agilidade os resultados para o Tableau.
- Embora mais exibições sejam mostradas quando o painel estiver completo, cada exibição apresenta apenas algumas marcas. Os recursos necessários para gerar cada uma dessas exibições, mesmo quando houver muitos usuários ativos no sistema, são comuns e, por isso, é pouco provável que o sistema fique sem memória.
- Por fim, você pode ver que, para os níveis de “navegação” mais altos, aproveitamos a oportunidade de mostrar o volume de vendas de cada categoria. Isso permite que usuário saiba se essa seleção contém muitos ou poucos registros. Também usamos cores para indicar a lucratividade de cada categoria. Isso agora é muito relevante, porque você pode ver quais áreas específicas exigem atenção, em vez de ficar navegando sem rumo.

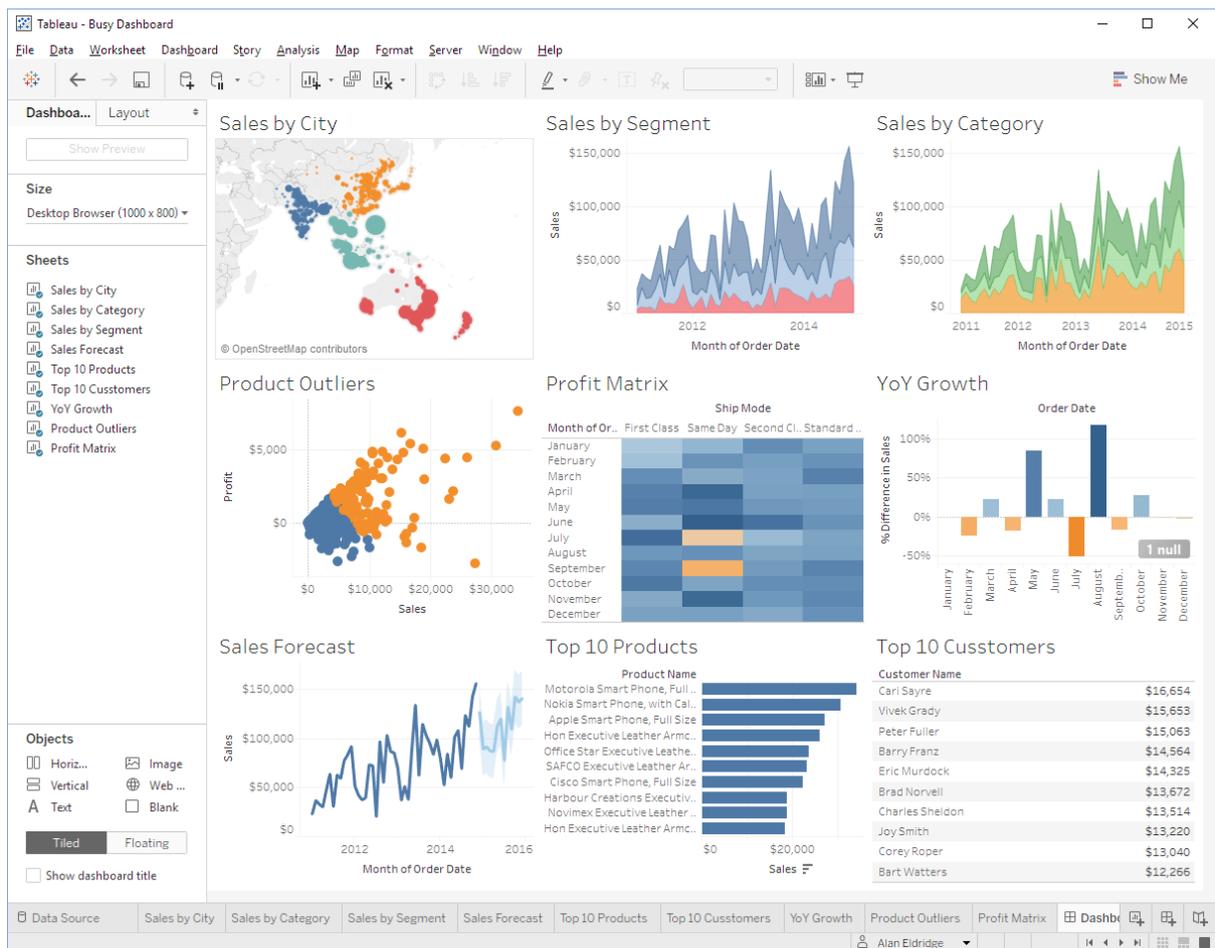
Prefira a simplicidade

Um erro comum dos novos usuários é a criação de painéis excessivamente “complexos”. Talvez eles estejam tentando recriar um documento que já usaram antes em outra ferramenta ou elaborar algo especificamente criado para ser um relatório impresso. O resultado final é uma pasta de trabalho lenta e ineficiente.

Os pontos mencionados a seguir contribuem para a complexidade:

Muitas planilhas por painel

É muito comum os novos usuários adicionarem vários gráficos/planilhas em um único painel.



Saiba que cada planilha executará uma ou, muito provavelmente, mais consultas nas fontes de dados. Portanto, quanto mais planilhas você tiver, mais demorada será a renderização do painel. Aproveite que o Tableau foi projetado para criar painéis interativos para usuários finais e espalhe os dados em vários painéis ou páginas.

Muitos cartões de filtro

Os cartões de filtro são um recurso poderoso do Tableau que permite criar painéis ricos e interativos para usuários finais. No entanto, cada filtro pode exigir a execução de uma determinada consulta para mostrar as opções. Assim, adicionar muitos filtros em seu painel pode aumentar inesperadamente o tempo de renderização do seu painel. Além disso, quando você usa a opção “mostrar valores relevantes” em um filtro, será necessário executar uma consulta para atualizar os valores mostrados sempre que os outros filtros forem alterados. Use esse recurso com moderação.

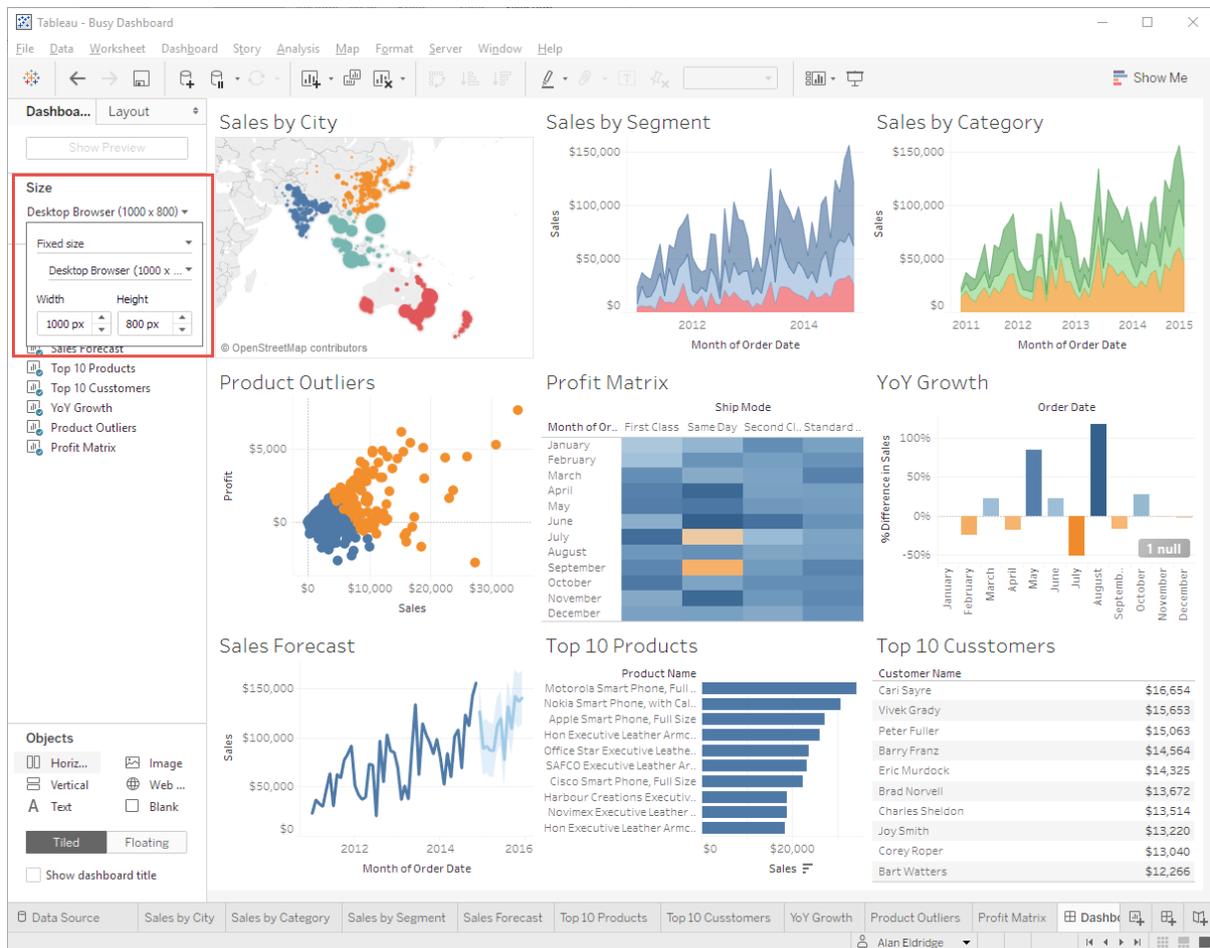
Se você também aplicou filtros a várias planilhas, saiba que cada alteração feita acionará diversas consultas, visto que todas as planilhas visíveis afetadas serão atualizadas (as consultas não serão executadas nas planilhas que não estiverem visíveis). Se essa operação demorar vários segundos para ser concluída, isso pode resultar em uma experiência de usuário ruim. Se você espera que seus usuários façam diversas alterações em um tipo de filtro multisseleção, considere disponibilizar o botão “Aplicar”, para que os usuários possam acionar a atualização depois que fizerem todas as suas seleções.

Adapte seu painel para garantir um bom desempenho

Depois de se certificar de que seu painel está o mais simples possível, podemos adaptar seu design para aproveitar o armazenamento em cache e melhorar ainda mais seu desempenho.

Painéis com tamanho fixo

Uma das medidas mais simples que podemos tomar para melhorar o desempenho de um painel é verificar se seu tamanho é fixo.



Parte do processo de renderização do Tableau é criar um layout, ou seja, especificar quantas linhas/colunas exibir para múltiplos pequenos e tabelas de referência cruzada, o número e o intervalo das escalas de eixo/linhas de grade a serem representados no gráfico, o número e o local dos rótulos de marca que serão mostrados, etc. Tudo isso é determinado pelo tamanho da janela na qual o painel será exibido.

Se tivermos várias solicitações para o mesmo painel, mas em janelas com tamanhos diferentes, precisaríamos gerar um layout para cada solicitação. Ao definir o layout do painel com um tamanho fixo, precisaremos criar apenas um layout que poderá ser reutilizado para todas as solicitações. Isso é ainda mais importante quando a renderização for feita no Tableau Server, visto que ter painéis de tamanho fixo significa que podemos armazenar em cache e compartilhar os bitmaps renderizados nele, otimizando seu desempenho e escalabilidade.

Painéis específicos para cada dispositivo

No Tableau 10, incluímos um novo recurso chamado Painéis específicos para cada dispositivo. Esse recurso permite criar layouts de painel personalizados, que são automaticamente selecionados com base no dispositivo em uso.

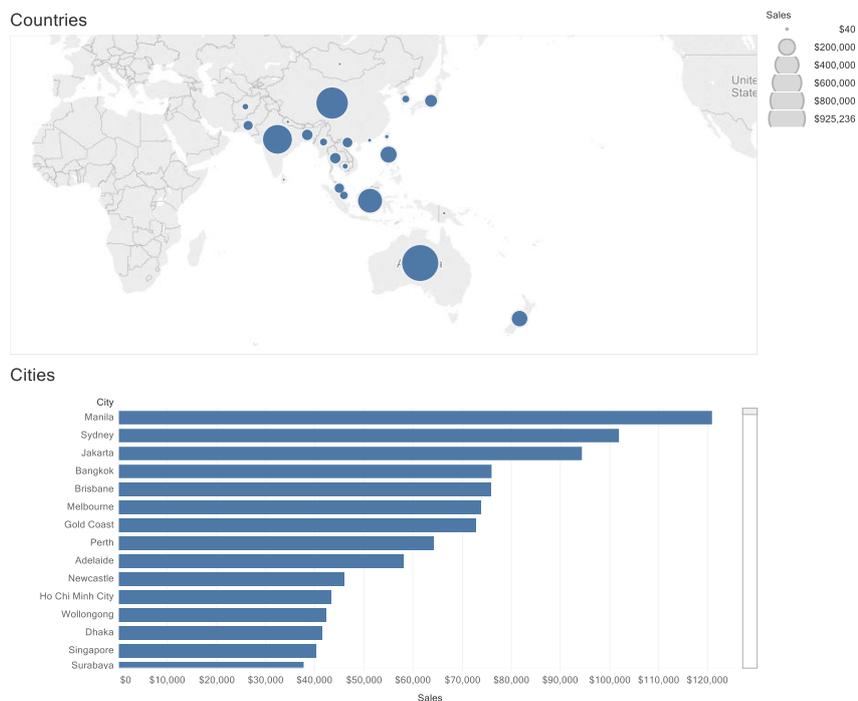
Escolhemos o layout a ser usado considerando o tamanho da tela:

- <= 500 px no eixo menor – telefone
- <= 800 px no eixo menor – tablet
- > 800 px – desktop

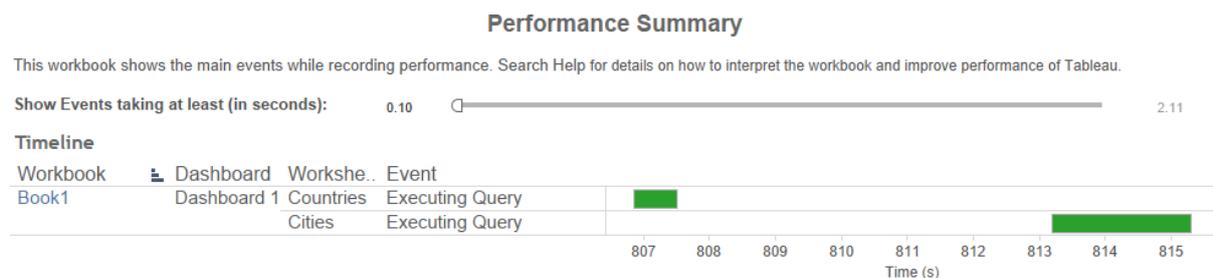
Dentro dessas especificações, como esses dispositivos terão tamanhos de tela diferentes e podem ser rotacionados, provavelmente você definirá os layouts para telefone/tablet com o redimensionamento automático. Isso proporcionará uma experiência de visualização melhor nos dispositivos, embora possa inviabilizar a reutilização do cache (tanto do cache de modelos de apresentação quanto do cache de blocos de imagem para renderização no servidor). De forma geral, dimensionar adequadamente um layout para um dispositivo compensa a não utilização do cache, mas é preciso avaliar bem isso. Assim que os usuários começarem a utilizar a pasta de trabalho, você disponibilizará os modelos e os bitmaps para os tamanhos de tela mais comuns, e o desempenho será aprimorado.

Usando o nível de detalhe da visualização para reduzir as consultas

Embora a prática recomendada seja adicionar apenas os campos necessários em cada planilha, há situações em que é possível melhorar o desempenho adicionando mais informações em uma pasta de trabalho para evitar a execução de consultas em outra pasta de trabalho. Veja o painel abaixo:



Se criarmos esse painel da forma esperada, o plano de execução resultará em duas consultas, uma em cada planilha:



```
SELECT [Superstore APAC].[City] AS [City],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[City]
```

```
SELECT [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Country]
```

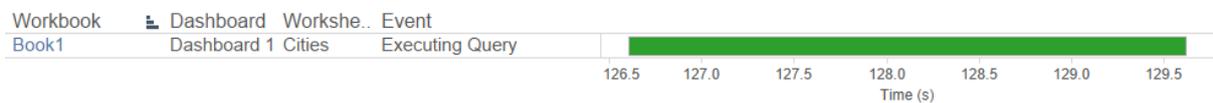
Se alterarmos o design do painel e adicionarmos Country (País) à planilha Cities (Cidades) (na divisória Detalhe), o Tableau poderá concluir o painel executando apenas uma consulta. O Tableau é inteligente o suficiente para executar a consulta na planilha Cities (Cidades) primeiro e utilizar o cache de resultados de consultas para fornecer os dados relevantes para a planilha Countries (Países). Esse recurso é chamado de “execução de consultas em lote”.

Performance Summary

This workbook shows the main events while recording performance. Search Help for details on how to interpret the workbook and improve performance of Tableau.

Show Events taking at least (in seconds): 0.10 3.02

Timeline

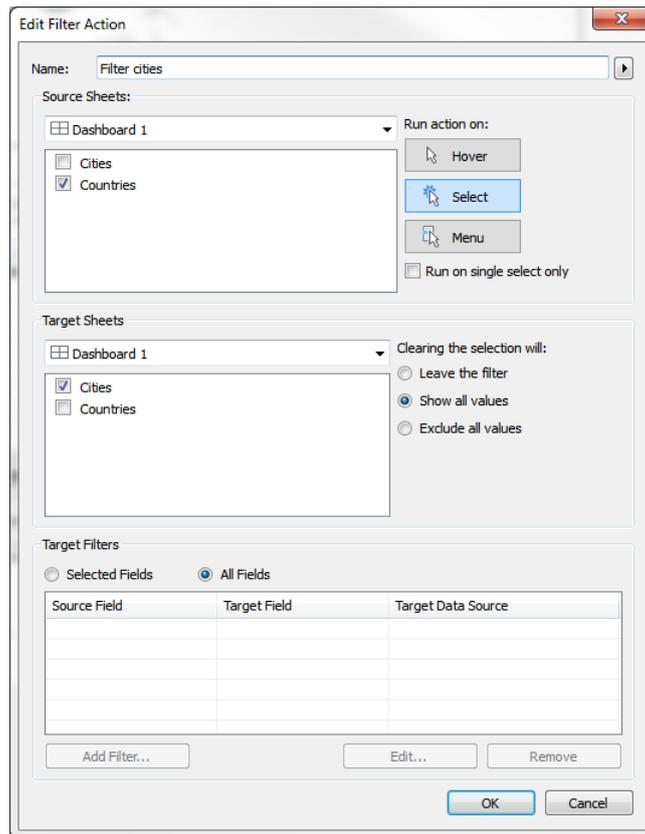


```
SELECT [Superstore APAC].[City] AS [City],
       [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[City],
         [Superstore APAC].[Country]
```

É claro que nem sempre poderemos fazer isso, visto que adicionar uma dimensão à exibição altera o nível de detalhe, podendo resultar na exibição de mais marcas. Porém, se você tiver uma relação hierárquica em seus dados, como mostrado no exemplo acima, essa é uma técnica muito útil, porque não afeta o nível de detalhe visível.

Usando o nível de detalhe da visualização para otimizar ações

Podemos usar uma abordagem semelhante com as ações para reduzir o número de consultas que precisamos executar. Considere o mesmo painel acima (antes da otimização que fizemos), mas agora adicionaremos uma ação de filtro da planilha Countries (Países) à planilha Cities (Cidades).



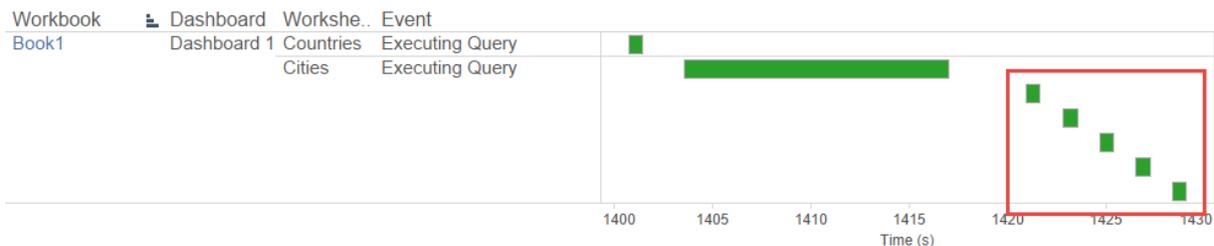
Quando dispararmos essa ação clicando em uma marca no mapa, veremos que o Tableau precisará executar uma consulta para determinar os valores na planilha Cities (Cidades). Isso acontece porque não há dados no cache de resultados de consultas para a relação cidade-país.

Performance Summary

This workbook shows the main events while recording performance. Search Help for details on how to interpret the workbook and improve performance of Tableau.

Show Events taking at least (in seconds): 0.10 13.47

Timeline



```
SELECT [Superstore APAC].[City] AS [City],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE ([Superstore APAC].[Country] = 'Australia')
GROUP BY [Superstore APAC].[City]
```

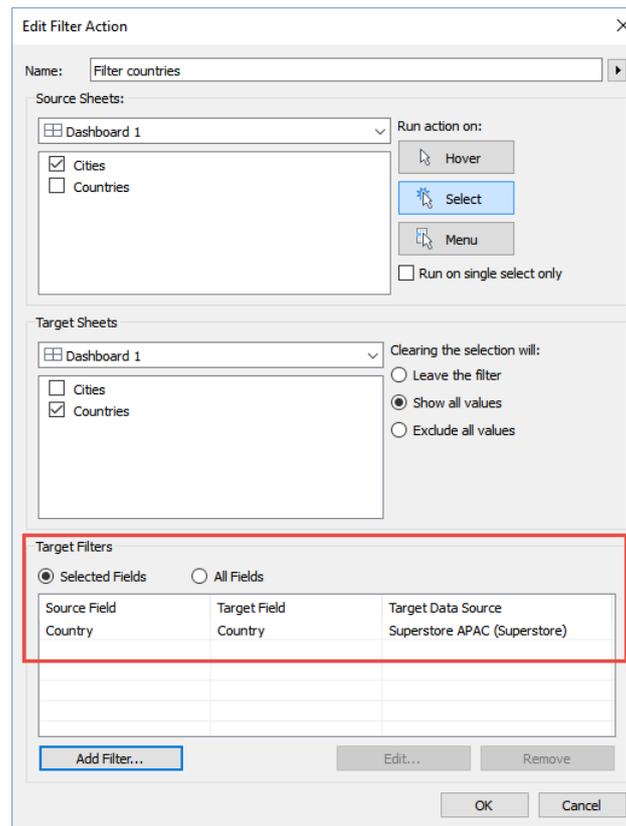
Se adicionarmos Country (País) à planilha Cities (Cidades), teremos informações suficientes no cache de resultados de consultas para podermos usar esses filtros sem precisar acessar a fonte de dados.

Também é possível otimizar o desempenho de forma semelhante quando a pasta de trabalho de origem for mais detalhada do que a de destino. Se utilizarmos a definição de ação padrão onde filtramos usando “todos os campos”:

Isso faz com que a pasta de trabalho execute uma consulta para cada ação, porque a instrução do filtro consulta os dados de Country (País) e City (Cidade), que não podem ser obtidos no cache de resultados de consultas da planilha Countries (Países).

```
SELECT [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE (([Superstore APAC].[City] = 'Sydney') AND ([Superstore APAC].[Country] =
'Australia'))
GROUP BY [Superstore APAC].[Country]
```

Se alterarmos a ação para filtrar apenas por Country (País):



Agora podemos satisfazer esse filtro no cache de resultados de consultas para não precisarmos acessar a fonte de dados. Conforme mencionado anteriormente, é necessário avaliar se alterar o nível de detalhe afetará o design da sua planilha. Se não alterar, essa pode ser uma técnica interessante.

Um bom design de planilha

O nível logo abaixo do painel é a planilha. No Tableau, o design da planilha está intrinsecamente relacionado às consultas que ela executa. Cada planilha gera uma ou mais consultas, por isso, neste nível, é importante garantir que as melhores consultas possíveis sejam geradas.

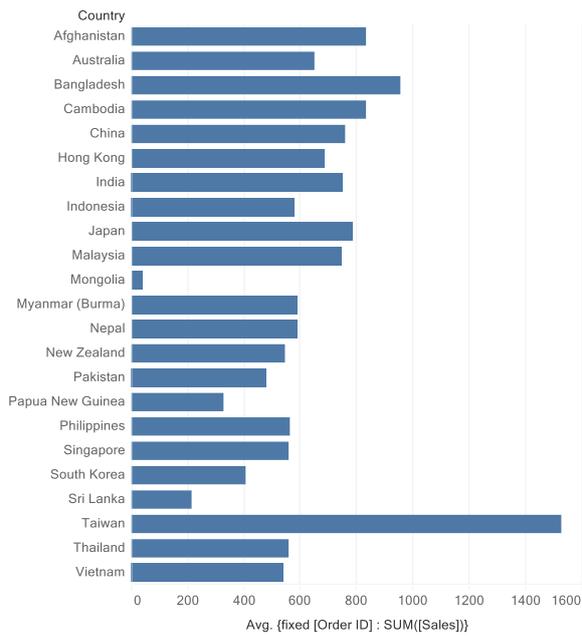
Inclua apenas os campos necessários.

Remova da divisória Detalhe todos os campos que não são usados diretamente pela visualização e que não são necessários na dica de ferramenta ou para promover o nível de detalhe de marca desejado. Isso agiliza a execução da consulta na fonte de dados e exige que menos dados sejam retornados nos resultados da consulta. Já vimos que existem algumas exceções a essa regra, como ajudar a execução de consultas em lote a eliminar consultas semelhantes em outras planilhas. No entanto, elas são menos comuns e precisam que o nível de detalhe visual da planilha seja mantido.

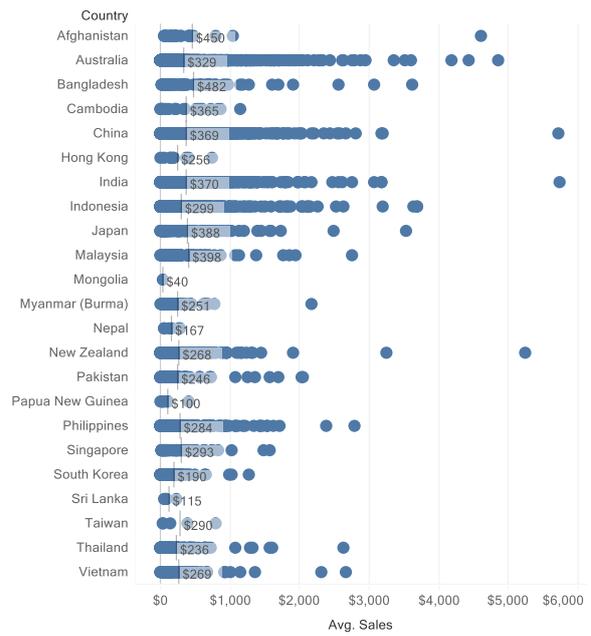
Mostre o número mínimo de marcas para responder a uma pergunta

No Tableau, geralmente é possível fazer o mesmo cálculo de várias formas. Veja o painel abaixo. As duas planilhas respondem à pergunta “qual é o tamanho de pedido médio por país?”

Avg Order Size 1



Avg Order Size 2

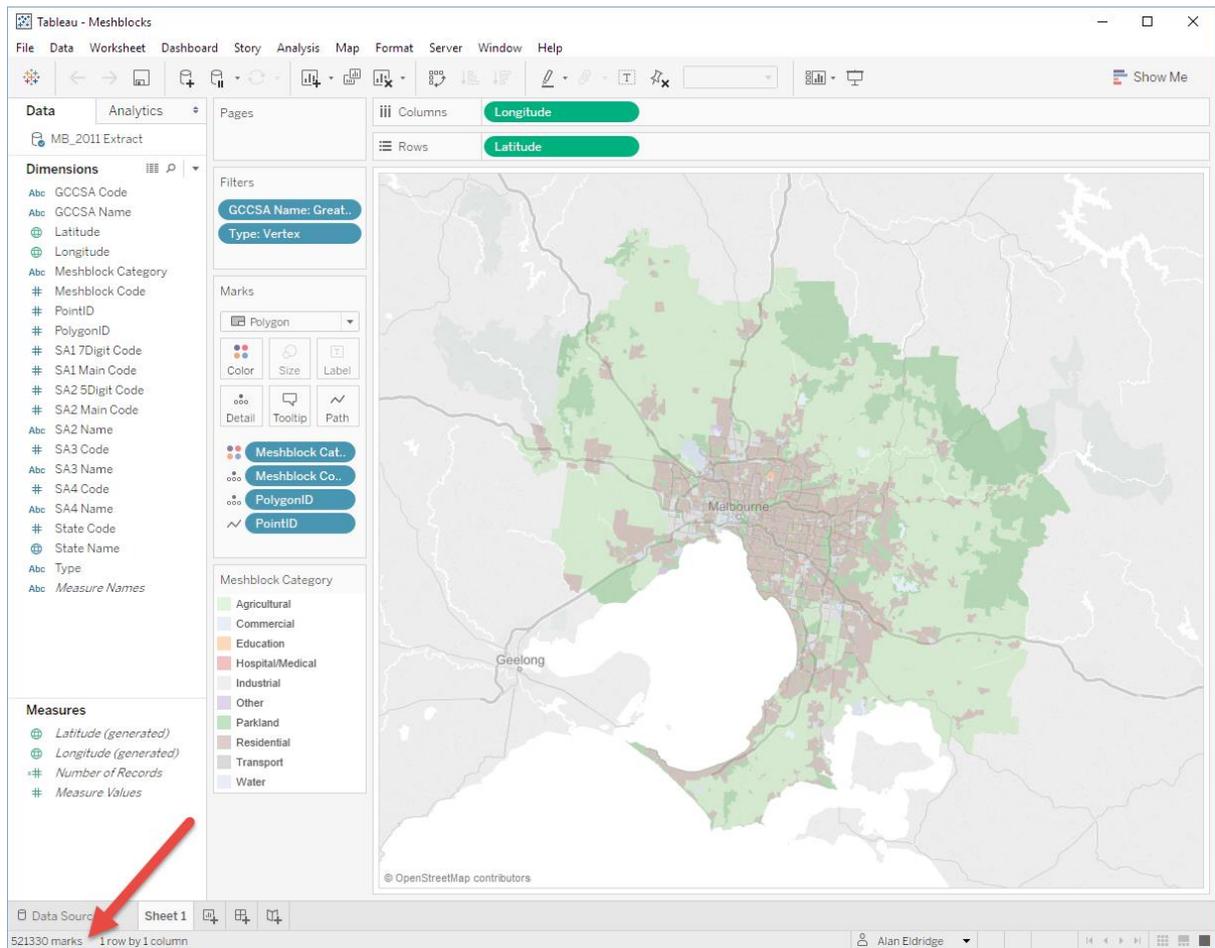


A Planilha 1 mostra apenas uma única marca que representa o tamanho de pedido médio para cada país. Por isso, apenas 23 registros são retornados da fonte de dados. Já a Planilha 2 mostra uma marca para cada pedido em cada país e, em seguida, calcula a média como uma linha de referência. Esse cálculo retorna 5.436 registros da fonte de dados.

Se estivermos interessados apenas na pergunta original, “qual é o tamanho de pedido médio por país?”, a Planilha 1 é a melhor opção. A Planilha 2 também responde a essa pergunta, mas oferece mais detalhes sobre as variações de tamanho dos pedidos, permitindo identificar exceções.

Evite visualizações muito complexas

Uma métrica importante a ser observada é quantos pontos de dados estão sendo renderizados em cada visualização. Para descobrir isso, basta consultar a barra de status no canto inferior esquerdo da janela do Tableau Desktop:

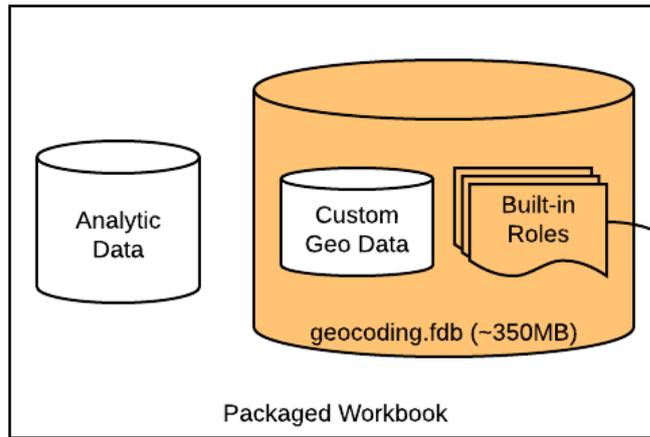


Embora não haja uma definição padrão para “excesso de marcas”, quanto maior for o número de marcas, mais CPU e RAM serão necessárias para renderizá-las. Tenha cuidado com tabelas de referência cruzada grandes, gráficos de dispersão e/ou mapas com polígonos personalizados complexos.

Mapas

Codificação geográfica personalizada

Quando você importa uma função de codificação geográfica personalizada, ela é gravada no banco de dados de codificação geográfica, que é um arquivo de banco de dados do Firebird armazenado por padrão em C:\Arquivos de programa\Tableau\Tableau 10.0\Local\data\geocoding.fdb. Se você usar a função em uma planilha e salvá-la como uma pasta de trabalho em pacote, o arquivo do banco de dados inteiro será compactado em um arquivo TWBX com todos os seus 350 MB de dados.

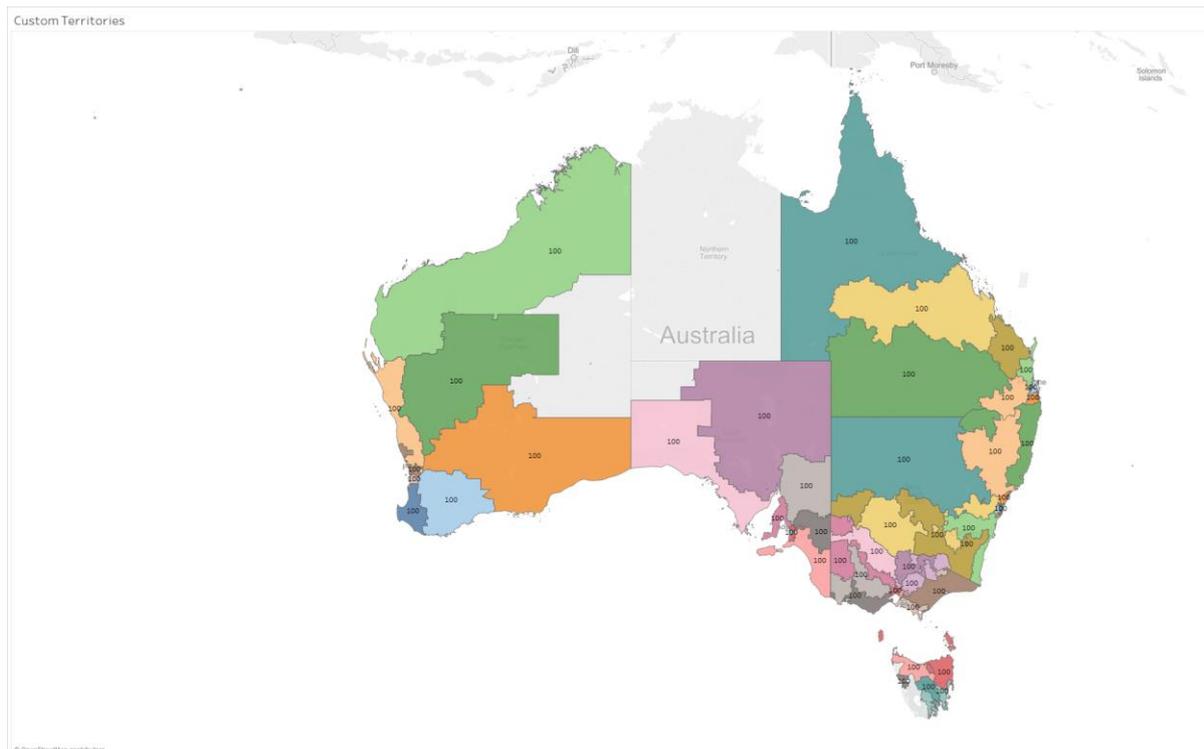


Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
AreaCode.tds	Tableau Datasource	2 KB	No	9 KB	82%	9/06/2016 1:29 PM
City.tds	Tableau Datasource	2 KB	No	13 KB	85%	9/06/2016 1:29 PM
CMSA.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
Congress.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
Country.tds	Tableau Datasource	3 KB	No	17 KB	87%	9/06/2016 1:29 PM
County.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
GEOCODING.FDB	FDB File	118,407 KB	No	359,280 KB	68%	9/06/2016 1:28 PM
State.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
Suburb.tds	Tableau Datasource	3 KB	No	15 KB	86%	9/06/2016 1:29 PM
ZipCode.tds	Tableau Datasource	2 KB	No	8 KB	81%	9/06/2016 1:29 PM

Isso torna o arquivo TWBX a) muito grande, visto que a carga de trabalho da codificação geográfica é de cerca de 110 MB compactados e b) muito lento de abrir, já que a descompactação inicial precisa ser feita com um conjunto de dados muito maior. Seria mais eficiente não importar os dados como uma função de codificação geográfica personalizada, e sim usar a combinação na pasta de trabalho para combinar os dados de análise com os dados geoespaciais. Com essa abordagem, o arquivo geocoding.fdb não é incorporado ao TWBX e contém apenas os dados de análise e de codificação geográfica personalizada.

Áreas personalizadas

Esse novo recurso do Tableau 10 permite que os usuários combinem áreas dos bancos de dados de codificação geográfica interna para criar regiões agregadas.



A renderização inicial de áreas personalizadas baseada em várias regiões de nível inferior pode ser muito lenta, por isso, use esse recurso com cuidado. Depois de definidas, as áreas personalizadas são armazenadas em cache, e o desempenho pode melhorar.

Mapas preenchidos x Mapas de ponto

As marcas de um mapa preenchido (utilizadas em um mapa ou como marcas em outro tipo de gráfico) exigem um processamento intenso quando a renderização no lado do cliente é usada, visto que precisamos enviar os dados de polígono para a forma e isso pode ser bastante complexo. Se você estiver observando um desempenho de renderização ruim, considere usar um mapa de símbolos.

Marcas de polígono

Todas as visualizações que usam marcas de polígono forçarão o Tableau Server a fazer a [renderização no lado do servidor](#), o que pode afetar a experiência do usuário final. Use esse recurso com moderação.

Outros fatores

Tabelas de referência cruzada grandes

Nas versões anteriores deste documento, recomendamos que você evitasse criar tabelas de referência cruzada grandes, porque sua renderização é muito lenta. O mecanismo subjacente desse tipo de visualização foi aprimorado em versões mais recentes, e agora as tabelas de referência cruzada são renderizadas com a mesma rapidez do que outros tipos de gráficos de múltiplos pequenos. Ainda assim, recomendamos que você avalie bem o uso de tabelas de referência cruzada grandes, pois elas exigem que uma grande quantidade dos dados da fonte de dados subjacente seja lida e não são muito úteis para análises.

Dicas de ferramenta

Por padrão, quando você adiciona uma dimensão à divisória Dica de ferramenta, a dimensão é agregada com o uso da função de atributo ATTR(). Essa operação exige que duas agregações sejam feitas na fonte de dados subjacente, MIN() e MAX(), e que ambos os resultados sejam enviados para

o conjunto de resultados. Consulte a seção sobre o [uso da função ATTR\(\)](#) mais adiante neste documento para obter mais informações.

Se você não estiver preocupado com a possibilidade de ter diversos valores de dimensão, a melhor opção é usar apenas uma agregação, em vez da função ATTR() padrão. Escolha MIN() ou MAX() e continue usando a função que você escolheu para maximizar suas chances de obter resultados no cache.

Outra opção, se você tiver certeza de que isso não afetará o nível de detalhe visual da sua visualização, é colocar a dimensão na divisória Nível de detalhe, em vez de colocá-la na divisória Dica de ferramenta. Com isso, o campo da dimensão será usado diretamente nas instruções SELECT e GROUP BY da consulta. Recomendamos que você faça um teste para saber se o desempenho foi melhor do que com uma única agregação, visto que isso pode variar de acordo com o desempenho da sua plataforma de dados.

Legendas

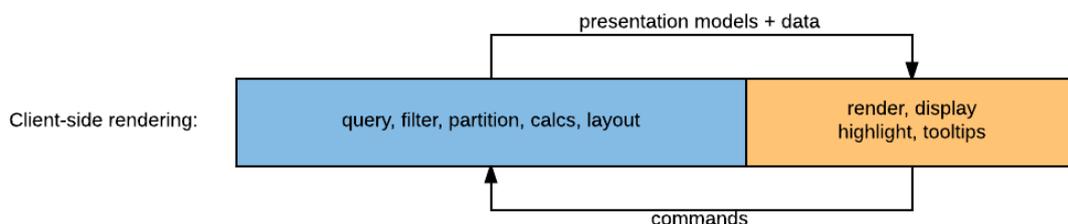
As legendas não costumam causar problemas de desempenho, pois seus domínios são preenchidos pelo cache de resultados de consultas. No entanto, elas podem prejudicar a renderização se o domínio enumerado for grande, porque os dados precisam ser transmitidos para o navegador do cliente. Se esse for o caso, a legenda não terá muita utilidade e deve ser removida.

Divisória Páginas

Alguns usuários acreditam que a divisória Páginas funciona da mesma forma que a divisória Filtros, reduzindo o número de registros retornados da fonte de dados. Isso não é verdade. A consulta da planilha retornará os registros para todas as marcas em todas as páginas. Se houver uma dimensão de página com um alto grau de cardinalidade (muitos valores exclusivos, por exemplo), isso pode aumentar significativamente o tamanho da planilha e afetar o desempenho. Use esse recurso com moderação.

Renderização no lado do cliente x Renderização no lado do servidor

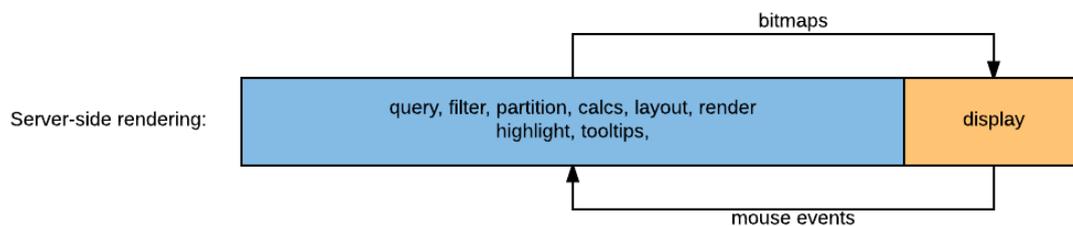
Antes que as marcas e os dados sejam mostrados em uma exibição no navegador da Web do cliente, eles são recuperados, interpretados e renderizados. O Tableau Server pode realizar esse processo no navegador da Web do cliente ou no servidor. A renderização no lado do cliente é o modo padrão, porque fazer a renderização e todas as interações no servidor pode resultar em mais transferências de dados na rede e atrasos em todo o processo. Com a renderização no lado do cliente, muitas interações na exibição são mais rápidas porque a interpretação e a renderização dessas interações são feitas diretamente no navegador.



(em azul – executado no servidor; em laranja – executado no navegador do cliente)

No entanto, algumas exibições são renderizadas com mais eficiência no lado do servidor, pois ele tem mais capacidade de processamento. A renderização no servidor é interessante para exibições complexas, porque os arquivos de imagem utilizam muito menos largura de banda do que os dados usados para criar imagens. Ela também é indicada para os usuários de tablets, que têm um desempenho inferior ao dos computadores e,

por isso, não podem lidar com exibições muito complexas. O problema aqui é que interações simples, como o uso das dicas de ferramentas e o destaque, podem ser lentas quando renderizadas no servidor, pois exigem um processamento completo no servidor.



(em azul – executado no servidor; em laranja – executado no navegador do cliente)

O Tableau Server é configurado para lidar automaticamente com todas essas situações usando um limite de complexidade como disparador para renderizar uma exibição no servidor, e não em um navegador da Web. Esse limite é diferente para computadores e dispositivos móveis. Por isso, há situações em que uma exibição aberta em um navegador da Web de um computador pode ser renderizada no cliente, mas a mesma exibição aberta em um navegador de um tablet é renderizada pelo servidor. A filtragem também pode alterar o comportamento de renderização. Uma pasta de trabalho pode ser aberta inicialmente com a renderização no lado do servidor e depois mudar para a renderização no lado do cliente após a aplicação de um filtro. Além disso, se uma visualização usar marcas de polígono ou a divisória Páginas, ela utilizará apenas a renderização no lado do servidor, mesmo se atender aos critérios para a renderização no lado do cliente. Por isso, use esses recursos com cuidado.

Como administrador, você pode testar ou ajustar essa configuração para computadores e tablets. Consulte o link a seguir para obter mais detalhes:

https://onlinehelp.tableau.com/current/server/pt-br/browser_rendering.htm

Filtros eficientes

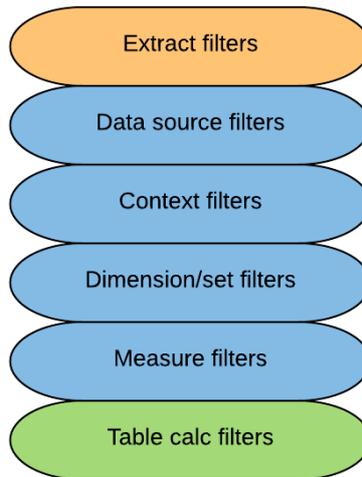
A filtragem no Tableau é muito poderosa e expressiva. No entanto, filtros ineficientes são uma das causas mais comuns de pastas de trabalho e painéis com desempenho ruim. As seções a seguir apresentam várias práticas recomendadas para trabalhar com filtros.

Nota: a eficiência dos filtros é significativamente afetada pela presença e manutenção de índices na fonte de dados. Consulte a seção sobre índices para obter mais detalhes.

Tipos de filtro

Prioridade dos filtros

No Tableau, os filtros são aplicados na seguinte ordem:



Filtros de extração

Esses filtros só podem ser aplicados quando extrações de dados forem usadas e, nesse caso, são logicamente aplicados antes de todos os outros filtros. Eles limitam os dados coletados da fonte de dados subjacente e podem ser filtros de dimensão ou de medida. Além disso, eles podem executar uma função TOP ou SAMPLE para reduzir o número de registros retornados, dependendo da plataforma dos dados de origem.

Filtros da fontes de dados

Os filtros da fonte de dados são os filtros de nível mais alto disponíveis em conexões em tempo real. A principal diferença entre os filtros da fonte de dados e os filtros de contexto é que os filtros da fonte de dados abrangem toda a fonte de dados, enquanto os filtros de contexto são definidos para cada planilha. Isso significa que os filtros de fontes de dados podem ser usados em uma fonte de dados publicada, enquanto os filtros de contexto são aplicados no nível da planilha.

Os filtros da fonte de dados podem ser uma forma eficaz de restringir uma fonte de dados para evitar que usuários finais acidentalmente executem uma consulta muito grande. Por exemplo, você pode adicionar um filtro de fonte de dados para limitar as consultas em uma tabela de transações apenas aos dados dos últimos seis meses.

Filtros de contexto

Por padrão, todos os filtros definidos no Tableau são calculados de modo independente. Ou seja, cada filtro acessa todas as linhas na sua fonte de dados, sem considerar outros filtros. No entanto, ao especificar um filtro de contexto, você pode tornar outros filtros dependentes, porque eles processarão apenas os dados que passarem pelo filtro de contexto.

Use os filtros de contexto quando precisar que eles obtenham uma resposta correta (por exemplo, N superior(es) filtrados). Por exemplo, você tem uma exibição que mostra os 10 principais produtos por SUM(Sales), filtrados por região. Sem um filtro de contexto, a seguinte consulta é executada:

```
SELECT [Superstore APAC].[Product Name] AS [Product Name],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
INNER JOIN (
  SELECT TOP 10 [Superstore APAC].[Product Name] AS [Product Name],
               SUM([Superstore APAC].[Sales]) AS [$_alias_0]
  FROM [dbo].[Superstore APAC] [Superstore APAC]
  GROUP BY [Superstore APAC].[Product Name]
  ORDER BY 2 DESC
) [t0] ON ([Superstore APAC].[Product Name] = [t0].[Product Name])
```

```
WHERE ([Superstore APAC].[Region] = 'Oceania')
GROUP BY [Superstore APAC].[Product Name]
```

Isso retorna a contribuição da Oceania para os 10 principais produtos globais. Se o seu objetivo é saber quais foram os 10 principais produtos na Oceania, adicione o filtro Região ao contexto, e a seguinte consulta será executada:

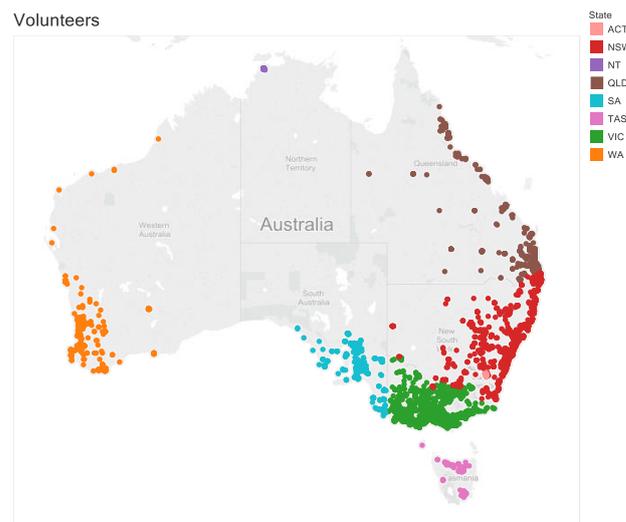
```
SELECT TOP 10 [Superstore APAC].[Product Name] AS [Product Name],
SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok],
SUM([Superstore APAC].[Sales]) AS [$__alias__0]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE ([Superstore APAC].[Region] = 'Oceania')
GROUP BY [Superstore APAC].[Product Name]
ORDER BY 3 DESC
```

Antes, os filtros de contexto eram implementados como uma tabela temporária na fonte de dados. Isso não acontece mais. Na maioria dos casos, os filtros de contexto são implementados como parte da consulta da fonte de dados (conforme mostrado acima) ou processados localmente no Processador de dados.

Você não deve usar os filtros de contexto como um mecanismo para melhorar o desempenho da consulta.

Filtragem de dimensões categóricas

Considere a visualização abaixo: um mapa da Austrália com marcas por código postal.



Há várias formas de filtrar o mapa para mostrar apenas os códigos postais do Oeste da Austrália (os pontos laranjas):

- Poderíamos selecionar todas as marcas dessa região e manter apenas a seleção.
- Poderíamos selecionar todas as marcas fora dessa região e excluir a seleção.
- Poderíamos manter apenas outro atributo, como a dimensão State (Estado).
- Poderíamos filtrar por intervalo, seja pelos valores de código postal ou pelos valores de latitude/longitude.

Discreto

Com as duas primeiras opções, o desempenho das opções Manter apenas e Excluir seria ruim. Na verdade, essas opções geralmente são mais lentas do que o conjunto de dados não filtrado. Isso acontece porque elas são expressas como uma lista discreta de valores de código postal, que são exibidos ou ocultados de acordo com o filtro pelo DBMS. Você também pode usar uma instrução

WHERE IN ou, se houver muitos valores, criar uma tabela temporária com os valores selecionados e usar uma UNIÃO INTERNA com ela e com as tabelas principais. Para um conjunto maior de marcas, isso pode resultar em uma consulta muito complexa de avaliar.

A terceira opção é rápida nesse exemplo porque o filtro resultante (`WHERE STATE="Western Australia"`) é muito simples e pode ser processado com eficiência pelo banco de dados. No entanto, essa abordagem é cada vez menos eficiente à medida que aumenta do número de membros da dimensão necessários para expressar o filtro, e pode ser semelhante ao desempenho dos recursos Seleção e Manter apenas.

Intervalo de valores

Usar a abordagem do filtro de intervalo de valores também permite que o banco de dados avalie uma instrução de filtragem simples (`WHERE POSTCODE >= 6000 AND POSTCODE` ou `WHERE LONGITUDE < 129`), resultando em uma execução rápida. No entanto, essa abordagem, diferentemente de um filtro em uma dimensão relacionada, não se torna mais complexa com o aumento do número de dimensões.

Com isso, podemos concluir que filtros com intervalo de valores geralmente são mais rápidos de avaliar do que grandes listas detalhadas de valores discretos. Por isso, seu uso deve ter preferência em relação às opções Manter apenas e Excluir para grandes conjuntos de marcas, se possível.

Filtros de divisão

Os filtros de divisão são os filtros das dimensões que não estão em uso na visualização (ou seja, que não fazem parte do nível de detalhe da visualização). Por exemplo, você tem uma visualização que mostra o total de vendas por país, mas a visualização está filtrada por região. Neste exemplo, a execução da consulta é feita da seguinte forma:

```
SELECT [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE ([Superstore APAC].[Region] = 'Oceania')
GROUP BY [Superstore APAC].[Country]
```

Esses filtros tornam-se mais complexos se dividirmos o resultado de uma agregação. Por exemplo, se filtrarmos a visualização acima para mostrar as vendas dos 10 principais produtos mais lucrativos, e não por região, o Tableau precisará executar duas consultas: uma no nível do produto, para isolar os 10 principais produtos mais lucrativos, e outra no nível do país, limitada pelos resultados da primeira consulta:

```
SELECT [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
INNER JOIN (
  SELECT TOP 10 [Superstore APAC].[Product Name] AS [Product Name],
              SUM([Superstore APAC].[Profit]) AS [$_alias_0]
  FROM [dbo].[Superstore APAC] [Superstore APAC]
  GROUP BY [Superstore APAC].[Product Name]
  ORDER BY 2 DESC
) [t0] ON ([Superstore APAC].[Product Name] = [t0].[Product Name])
GROUP BY [Superstore APAC].[Country]
```

Tenha cuidado ao usar os filtros de divisão, porque sua avaliação pode ser trabalhosa. Observe também que, como a dimensão não faz parte do cache de resultados de consultas, não podemos agilizar a filtragem no navegador com os filtros de divisão (consulte a [seção anterior](#) que compara a renderização no lado do cliente e no lado do servidor).

Filtros entre fontes de dados

A filtragem entre fontes de dados é um recurso novo do Tableau 10. Ela permite a aplicação de um filtro a diversas fontes de dados que tenham um ou mais filtros em comum. As relações são definidas como em uma combinação: automaticamente, com base nas correspondências de nome/tipo, ou manualmente, por meio de relações personalizadas definidas no menu Dados.

Os filtros entre fontes de dados têm as mesmas implicações de desempenho que os filtros rápidos de um painel. Quando eles são atualizados, diversas áreas também são atualizadas, o que possivelmente exige a execução de várias consultas. Use-os com sabedoria e, se você espera que seus usuários façam várias alterações, considere exibir o botão “Aplicar” para disparar as consultas apenas depois que todas as seleções forem feitas.

Observe também que o domínio do filtro é obtido da fonte de dados “primária”, ou seja, a primeira fonte de dados usada da planilha na qual o filtro foi criado. Se um campo relacionado tiver domínios em fontes de dados diferentes, tenha cuidado ao escolher qual delas você usará, visto que o mesmo filtro pode acabar exibindo valores diferentes, conforme mostrado abaixo:

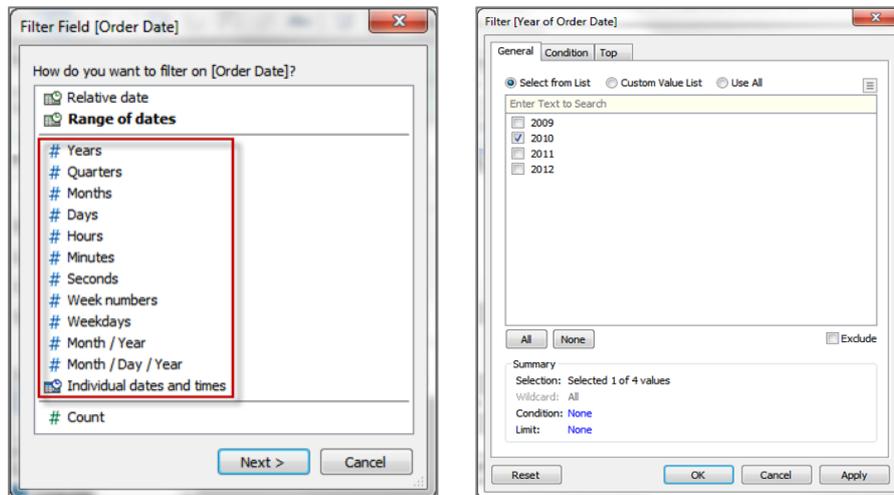


Filtragem de datas: discreta, intervalo e relativa

Os campos de data são um tipo especial de dimensão que o Tableau frequentemente manipula de modo diferente do que os dados categóricos padrão. Isso é especialmente verdadeiro durante a criação de filtros de data. Os filtros de data são extremamente comuns e se enquadram em três categorias: Filtros de datas relativas, que mostram um intervalo de datas relativo a um dia específico; Filtros de intervalo de datas, que mostram um intervalo definido de datas discretas; e Filtros de datas

discretas, que mostram datas individuais selecionadas em uma lista. Como mostrado na seção acima, o método usado pode ter um impacto significativo na eficiência da consulta resultante.

Discreta



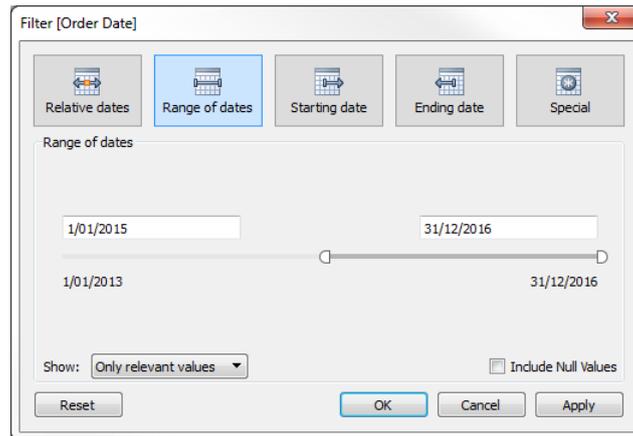
Às vezes, você deseja filtrar para incluir datas individuais específicas ou níveis de datas inteiros. Esse tipo de filtro é denominado filtro de datas discretas, pois você está definindo valores discretos, em vez de um intervalo. Esse tipo de filtro faz com que a expressão de data seja transmitida ao banco de dados como um cálculo dinâmico:

```
SELECT [FactSales].[Order Date], SUM([FactSales].[SalesAmount])
FROM [dbo].[FactSales] [FactSales]
WHERE (DATEPART(year,[FactSales].[Order Date])) = 2010)
GROUP BY [FactSales].[Order Date]
```

Na maioria dos casos, os otimizadores de consulta avaliarão de forma inteligente o cálculo DATEPART. No entanto, há alguns cenários em que o uso de filtros de data discreta pode resultar em uma execução de consulta ruim. Por exemplo, consultar uma tabela particionada com um filtro de datas discretas na chave de partição de data. Como a tabela não está particionada no valor DATEPART, alguns bancos de dados avaliarão o cálculo em todas as partições para encontrar registros que correspondem aos critérios, embora isso não seja necessário. Nesse caso, o desempenho pode ser melhor com o uso de um filtro de “intervalo de datas” ou de data relativa com uma data de referência especificada.

Uma forma de otimizar o desempenho desse tipo de filtro é materializar o cálculo usando uma extração de dados. Primeiro, crie um campo calculado que implemente explicitamente a função DATEPART. Se depois você criar uma extração de dados do Tableau, esse campo calculado será materializado como valores armazenados na extração (porque o resultado da expressão é determinista). Filtrar pelo campo calculado, em vez de filtrar pela expressão dinâmica, será mais rápido porque o valor pode ser procurado, e não calculado no momento da consulta.

Intervalo de datas

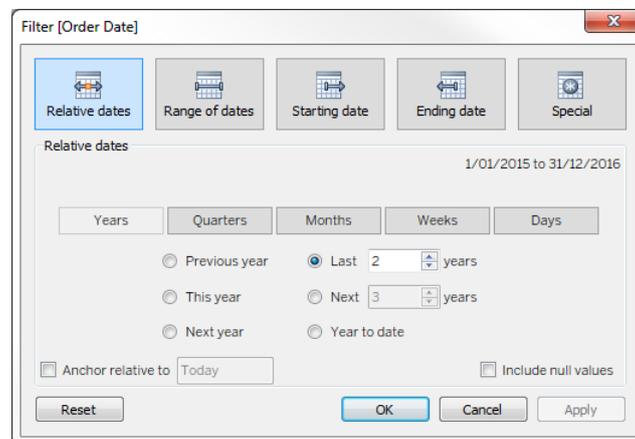


Esse tipo de filtro é usado quando você precisa especificar um intervalo de datas contíguas. Ele resulta na seguinte estrutura de consulta transmitida ao banco de dados:

```
SELECT SUM([factOrders].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[factOrders] [factOrders]
WHERE (([factOrders].[Order Date] >= {d '2015-01-01'}) AND
([factOrders].[Order Date] <= {d '2016-12-31'}))
GROUP BY ()
```

Esse tipo de cláusula WHERE é muito eficiente para otimizadores, permitindo que os planos de execução aproveitem os índices e as partições para obter um efeito completo. Se você estiver percebendo consultas lentas ao adicionar filtros de data discreta, considere substituí-los por filtros de intervalo de datas e veja se faz diferença.

Datas relativas



Um filtro de datas relativas permite a você definir um intervalo de datas que é atualizado com base na data e na hora de abertura da exibição. Por exemplo, talvez você queira visualizar as vendas do Ano até a data, todos os registros dos últimos 30 dias ou bugs fechados na última semana. Os filtros de datas relativas também podem ser relativos a uma data de referência específica, e não a hoje.

```
SELECT SUM([factOrders].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[factOrders] [factOrders]
WHERE (([factOrders].[Order Date] >= {ts '2015-01-01 00:00:00'}) AND
([factOrders].[Order Date] < {ts '2017-01-01 00:00:00'}))
GROUP BY ()
```

Como você pode ver, a cláusula WHERE resultante usa uma sintaxe de intervalo de datas. Portanto, essa também é uma forma eficiente de filtrar dados.

Observe que, devido à natureza mutável dos filtros de dados relacionados à data/hora atual, seja por meio da opção de filtro de dados relativo acima ou pelo uso explícito de NOW() ou TODAY() em uma fórmula de filtro, o cache de consultas não é tão eficiente para as consultas que os utilizam. O Tableau marca essas consultas como “consultas transitórias” no servidor do cache, e seus resultados não são mantidos por tanto tempo quanto os resultados de outras consultas.

Cartões de filtro

Exibir muitos cartões de filtro pode causar lentidão, principalmente se eles forem configurados para usar “Somente valores relevantes” e se você tiver muitas listas discretas. Experimente uma abordagem de análise mais orientada e use filtros de ação em um painel. Se você estiver criando uma exibição com muitos filtros para que ela seja extremamente interativa, avalie se ter vários painéis com diferentes níveis e temas funcionaria melhor (dica: provavelmente sim).

Enumerados x não enumerados

Os cartões de filtro enumerados exigem que o Tableau consulte a fonte de dados para obter valores de campo antes que o cartão de filtro seja renderizado. São eles:

- Lista de valores múltiplos – *todos os membros da dimensão*
- Lista de valores únicos – *todos os membros da dimensão*
- Compactar lista – *todos os membros da dimensão*
- Controle deslizante – *todos os membros da dimensão*
- Filtros de medida – *valores MIN e MAX*
- Filtros de intervalo de datas – *valores MIN e MAX*

Os cartões de filtro não enumerados, por outro lado, não exigem o conhecimento dos valores de campo possíveis. São eles:

- Lista de valores personalizados;
- Correspondência de curinga;
- Filtros de datas relativas;
- Filtros de busca de períodos de data.

Consequentemente, os cartões de filtro não enumerados reduzem o número de consultas relacionadas a filtros que precisam ser executadas pela fonte de dados. Os cartões de filtro não enumerados são renderizados com mais rapidez quando há muitos membros da dimensão para exibir.

Usar cartões de filtro não enumerados pode melhorar o desempenho, embora isso comprometa o contexto visual do usuário final.

Valores relevantes

Os cartões de filtro enumerados podem ser definidos para mostrar os valores de campo possíveis de três formas diferentes:

- Todos os valores no banco de dados – quando você seleciona essa opção, todos os valores no banco de dados são mostrados, independentemente dos outros filtros na exibição. O filtro não precisará consultar novamente o banco de dados quando outros filtros forem alterados.
- Todos os valores no contexto – essa opção está disponível apenas quando há filtros de contexto ativos. O cartão de filtro mostrará todos os valores no contexto, independentemente dos outros filtros na exibição. O filtro não precisará consultar o banco de dados novamente quando os filtros de medida ou dimensão forem alterados, mas isso será necessário se o contexto mudar.

- Somente valores relevantes – quando você seleciona essa opção, outros filtros são considerados e somente valores que passam nesses filtros são mostrados. Por exemplo, um filtro em Estado mostrará apenas os estados da região Leste quando um filtro de Região estiver definido. Consequentemente, o filtro precisará fazer uma nova consulta na fonte de dados quando outros filtros forem alterados.

Como você pode ver, a configuração “Somente valores relevantes” pode ser muito útil para ajudar os usuários a fazer seleções relevantes, mas ela pode aumentar muito o número de consultas enquanto os usuários interagem com o painel. Ela deve ser usada moderadamente.

Alternativas ao cartão de filtro

Existem alternativas ao uso dos cartões de filtro que fornecem resultados de análise semelhantes, mas sem o excesso de consultas adicionais. Você pode criar um parâmetro e filtrar com base nas seleções dos usuários, por exemplo.

- PRÓS:
 - Os parâmetros não exigem uma consulta à fonte de dados antes da renderização
 - Os parâmetros e os campos calculados podem implementar uma lógica mais complexa do que o que pode ser feito com um filtro de campo simples
 - Os parâmetros podem ser usados para filtrar dados em fontes de dados. Nas versões anteriores ao Tableau 10, os filtros operam apenas em uma única fonte de dados. No Tableau 10 e em suas versões posteriores, os filtros podem ser configurados para operar em várias fontes de dados relacionadas
- CONTRAS:
 - Os parâmetros admitem apenas um valor. Você não pode usá-los se quiser que o usuário tenha a opção de selecionar mais de um valor
 - Os parâmetros não são dinâmicos. A lista de valores é definida quando eles são criados e não é atualizada com base nos valores no DBMS

Outra alternativa é usar ações de filtro entre as exibições

- PRÓS:
 - As ações permitem a seleção de mais de um valor, seja por meio da seleção visual ou da combinação de teclas CTRL/SHIFT + clique
 - As ações mostram uma lista dinâmica de valores que é avaliada no momento da execução
 - As ações podem ser usadas para filtrar dados em fontes de dados. Nas versões anteriores ao Tableau 10, os filtros operam apenas em uma única fonte de dados. No Tableau 10 e em suas versões posteriores, os filtros podem ser configurados para operar em várias fontes de dados relacionadas
- CONTRAS:
 - A configuração das ações de filtro é mais complexa do que a dos cartões de filtro
 - As ações não apresentam a mesma interface de usuário dos parâmetros ou dos cartões de filtro e geralmente exigem mais da tela para a exibição
 - A planilha da origem da ação ainda precisa consultar a fonte de dados, apesar de se beneficiar do armazenamento em cache no pipeline de processamento do Tableau

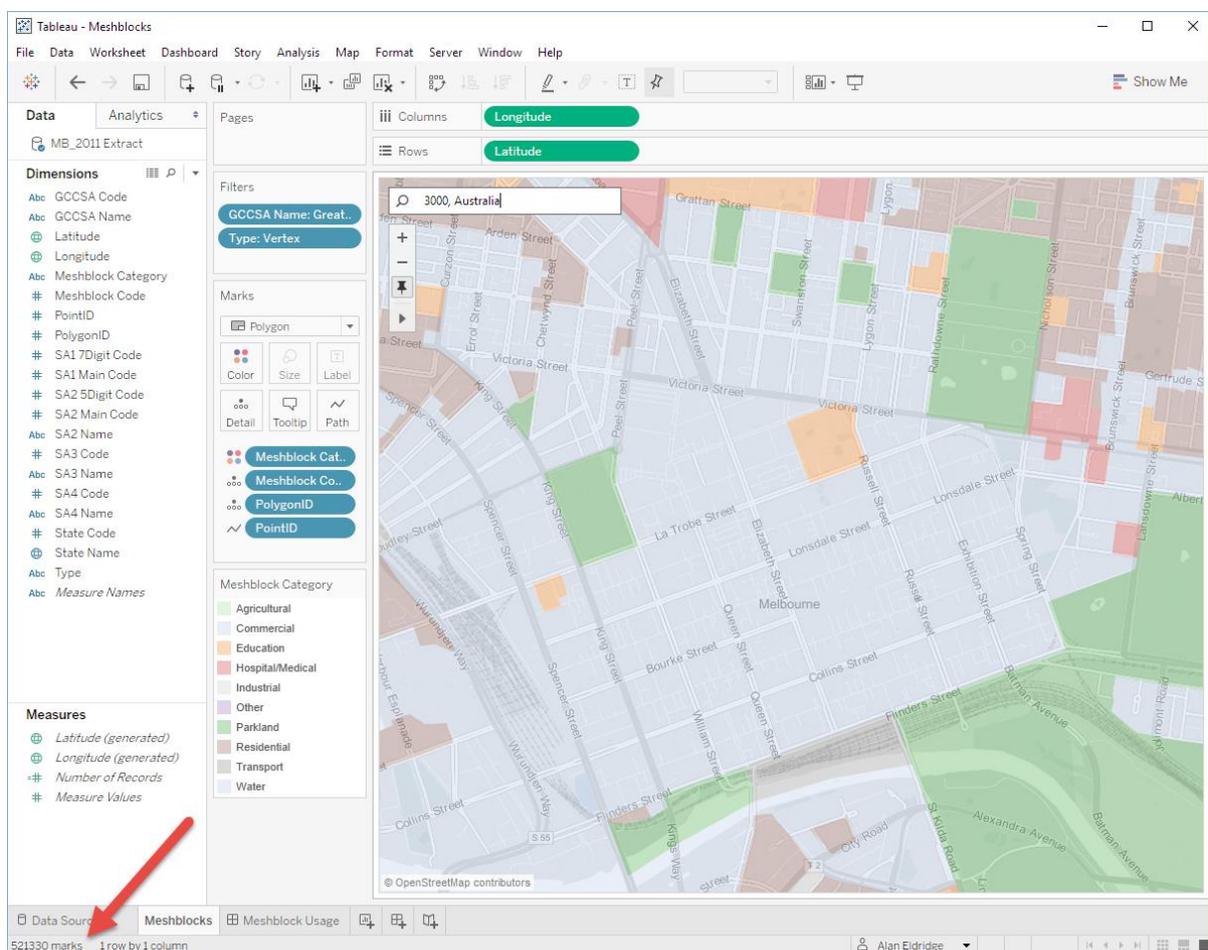
Para obter mais informações sobre outras técnicas de design que não se baseiam tanto em cartões de filtro, consulte a [seção anterior](#) sobre como criar um painel eficiente.

Filtros de usuário

Quando uma pasta de trabalho tiver filtros de usuário, criados na caixa de diálogo “Criar filtro de usuário...” ou por meio de campos calculados que utilizam uma das funções de usuário internas, como ISMEMBEROF(), o cache do modelo não é compartilhado nas sessões de usuário. Isso pode resultar em uma taxa de reutilização do cache muito mais baixa, o que significa mais trabalho para o Tableau Server. Use esses tipos de filtro com moderação.

Ampliar/reduzir x filtragem

Quando ampliamos uma visualização com muitas marcas, não estamos filtrando as marcas que não podemos ver. Estamos apenas alterando o ponto de visualização dos dados. O total de marcas processadas não muda, como podemos ver na imagem abaixo:



Se você precisar de apenas um subconjunto dos dados, use um filtro para ocultar os dados indesejados e deixe o recurso de ampliação automática definir a janela de visualização.

O problema são os meus cálculos?

Em muitos casos, a fonte de dados não fornecerá todos os campos de que você precisa para responder a todas as suas perguntas. Os campos calculados ajudarão você a criar todas as dimensões e medidas necessárias para sua análise.

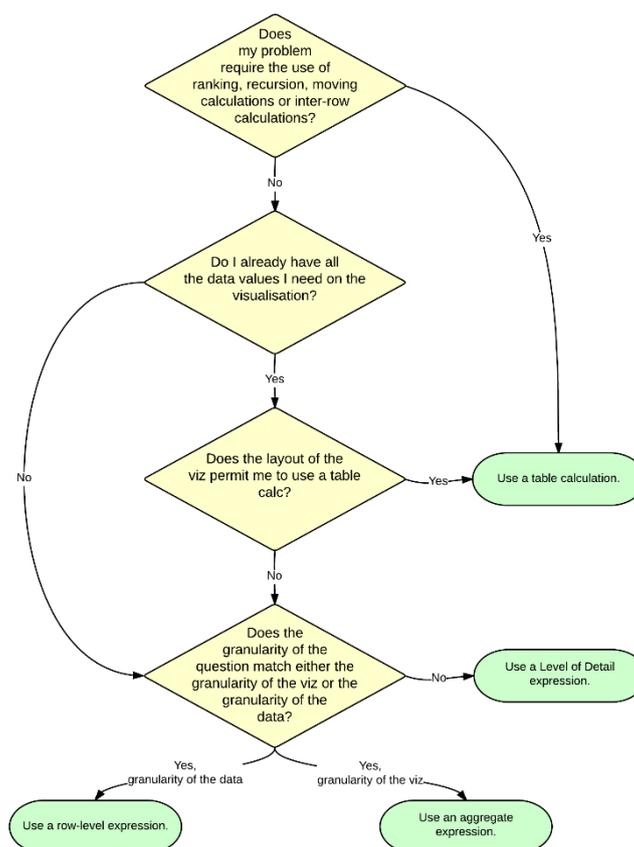
Em um campo calculado, é possível definir uma constante em código fixo (um imposto, por exemplo), fazer operações matemáticas muito simples, como subtração ou multiplicação (por exemplo, receitas menos custos), usar fórmulas matemáticas mais complexas, realizar testes lógicos (IF/THEN, CASE), fazer conversões de tipo e muito mais.

Depois de definido, o campo calculado é disponibilizado em toda a pasta de trabalho, desde que as planilhas estejam usando a mesma fonte de dados. Você pode usar campos calculados na pasta de trabalho da mesma forma que usaria dimensões e medidas na fonte de dados.

Existem quatro tipos diferentes de cálculos no Tableau:

- Cálculos no nível da linha
- Cálculos de agregação
- Cálculos de tabela
- Expressões de nível de detalhe

Use o fluxograma abaixo para saber como selecionar a melhor abordagem:



Uma ótima referência para consultar sobre como realizar cálculos complexos e um fórum onde os usuários compartilham soluções para problemas comuns é a Biblioteca de Referências sobre Cálculos do Tableau:

<http://tabsoft.co/1I0SsWz>

Tipos de cálculo

Cálculos de agregação e no nível da linha

Os cálculos no nível da linha e de agregação são expressos como parte da consulta enviada à fonte de dados e, em seguida, processados pelo banco de dados. Por exemplo, uma visualização que mostra a soma total das vendas de cada ano da data do pedido enviaria a seguinte consulta à fonte de dados:

```
SELECT DATEPART(year, [OrdersFact].[Order Date]) AS [yr:Order Date:ok],
       SUM([OrdersFact].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[OrdersFact] [OrdersFact]
GROUP BY DATEPART(year, [OrdersFact].[Order Date])
```

O cálculo **YEAR** é um cálculo no nível da linha, e o cálculo **SUM(SALES)** é um cálculo de agregação.

Em geral, os cálculos no nível da linha e de agregação são escalonados com facilidade, e existem muitas técnicas de ajuste de banco de dados que podem ser empregadas para melhorar o desempenho.

No Tableau, observe que a consulta no banco de dados pode não ser uma tradução direta dos cálculos básicos usados na visualização. Por exemplo, a consulta abaixo será executada quando a visualização tiver um campo calculado Razão de lucro definido como $SUM([Profit])/SUM([Sales])$:

```
SELECT DATEPART(year, [OrdersFact].[Order Date]) AS [yr:Order Date:ok],
       SUM([OrdersFact].[Profit]) AS
[TEMP (Calculation_0260604221950559) (1796823176) (0)],
       SUM([OrdersFact].[Sales]) AS
[TEMP (Calculation_0260604221950559) (3018240649) (0)]
FROM [dbo].[OrdersFact] [OrdersFact]
GROUP BY DATEPART(year, [OrdersFact].[Order Date])
```

O Tableau coleta os elementos do cálculo e executa a função de divisão na camada do cliente. Isso garante que $SUM([Profit])$ e $SUM([Sales])$ no nível do Ano sejam armazenadas em cache e utilizadas posteriormente em outra área da pasta de trabalho, sem que seja necessário acessar a fonte de dados.

Por fim, a fusão de consultas (combinação de várias consultas lógicas em uma única consulta) pode modificar a consulta executada na fonte de dados. Isso pode resultar em várias medidas de várias planilhas combinadas em uma única consulta, se essas outras planilhas tiverem a mesma granularidade.

Usando a função ATTR()

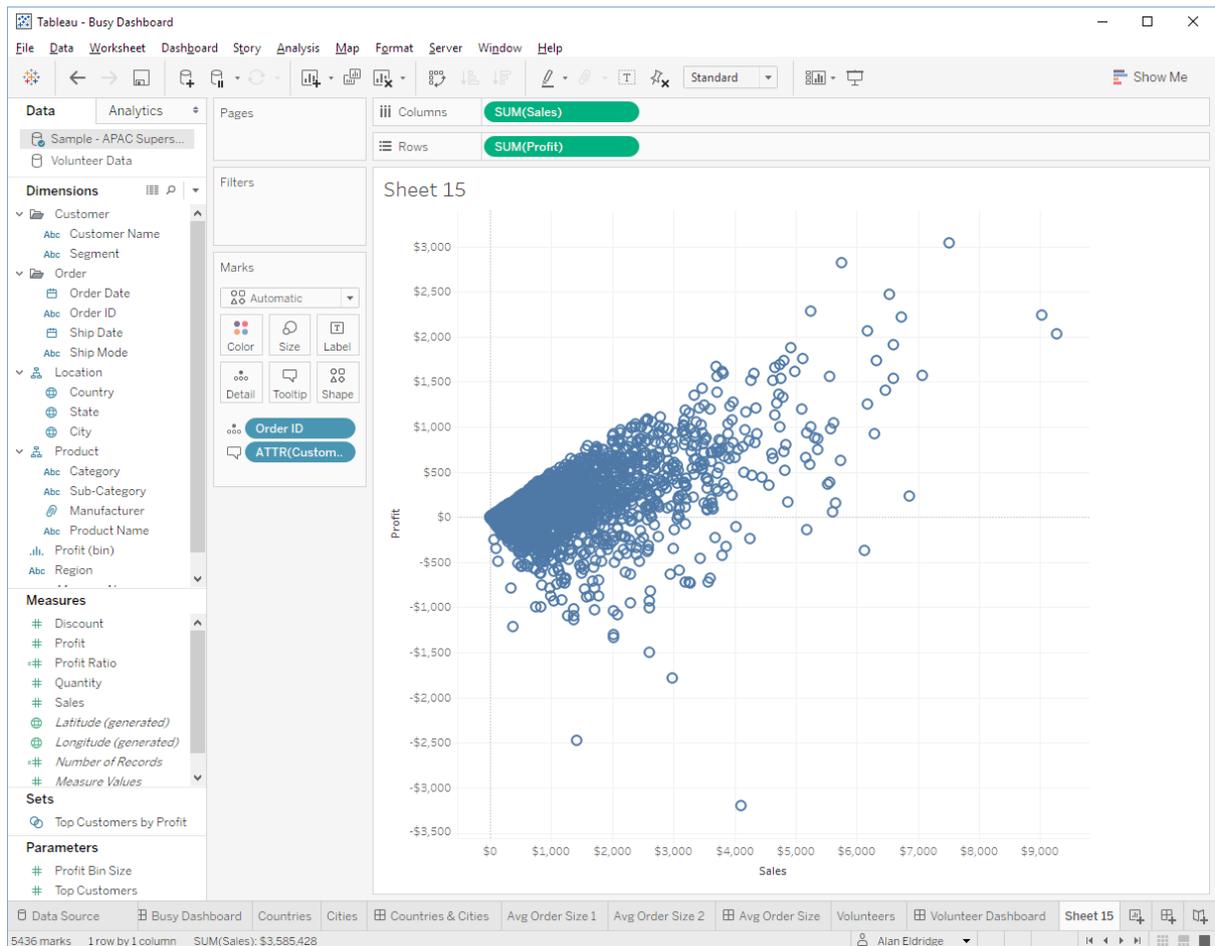
A ATTR() é uma função poderosa, geralmente usada como uma agregação de dimensões categóricas. Em outras palavras, ela retorna um valor se ele for exclusivo. Caso contrário, ela retorna *.

Tecnicamente, a função realiza o seguinte teste lógico:

```
IF MIN([dimension]) = MAX([dimension])
  THEN [dimension]
  ELSE "*"
END
```

Essa operação exige que duas agregações sejam feitas na fonte de dados subjacente, MIN() e MAX(), e que ambos os resultados sejam enviados para o conjunto de resultados. Se você não estiver preocupado com a possibilidade de ter diversos valores de dimensão, uma solução mais eficiente é usar apenas uma agregação – MIN() ou MAX(). Você pode usar qualquer uma das duas, mas deve usar sempre a mesma para maximizar suas chances de obter um resultado do cache.

Para mostrar como a função ATTR() afeta a fonte de dados, criamos esta visualização:



A função ATTR() é a agregação padrão usada quando uma dimensão é adicionada à divisória Dica de ferramenta. Isso gera a seguinte consulta:

```
SELECT [Superstore APAC].[Order ID] AS [Order ID],
       MAX([Superstore APAC].[Customer Name]) AS [TEMP(attr:Customer
Name:nk) (2542222306) (0)],
       MIN([Superstore APAC].[Customer Name]) AS [TEMP(attr:Customer
Name:nk) (3251312272) (0)],
       SUM([Superstore APAC].[Profit]) AS [sum:Profit:ok],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Order ID]
```

Se soubermos que não há vários clientes para cada ID de pedido, podemos alterar a agregação para MIN() e simplificar a consulta:

```
SELECT [Superstore APAC].[Order ID] AS [Order ID],
       MIN([Superstore APAC].[Customer Name]) AS [min:Customer Name:nk],
       SUM([Superstore APAC].[Profit]) AS [sum:Profit:ok],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Order ID]
```

A tabela abaixo mostra a diferença de desempenho das duas opções (os resultados são os segundos necessários para a conclusão da consulta, realizada no arquivo de log):

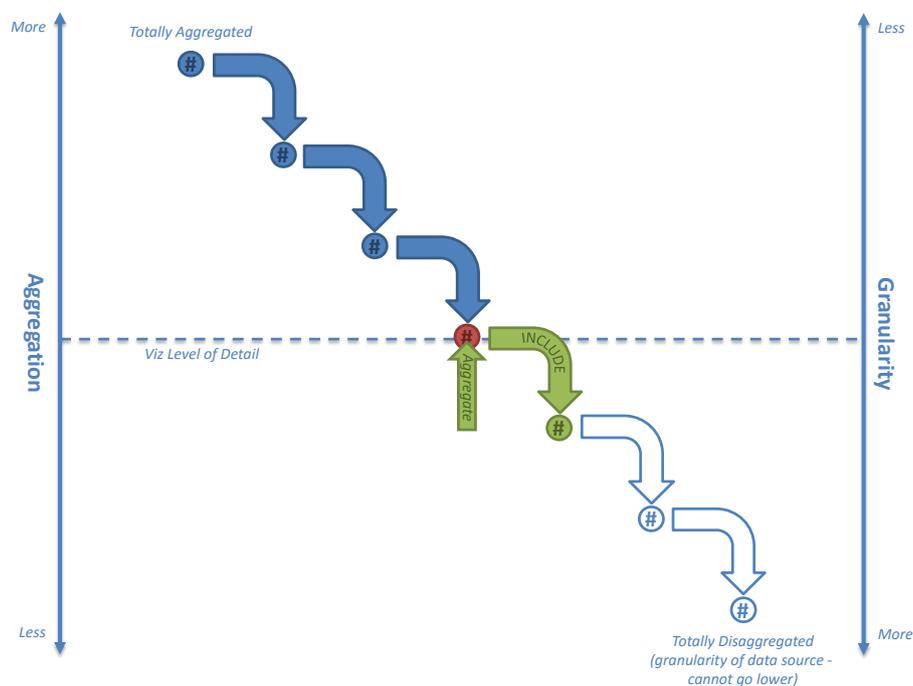
Test #	ATTR	MIN
1	2.824	1.857
2	2.62	2.09
3	2.737	2.878
4	2.804	1.977
5	2.994	1.882
Average	2.7958	2.1368
% improvement		24%

Para obter mais informações sobre a função ATTR(), leia este post no blog da InterWorks:

<http://bit.ly/1YEuZhX>

Cálculos de nível de detalhe

As expressões de nível de detalhe (LOD) permitem definir o nível de detalhe no qual um cálculo deve ser executado, independentemente do LOD da visualização.



As expressões de nível de detalhe (LOD) podem, em alguns casos, permitir que você substitua os cálculos que criou de uma forma mais complexa em versões anteriores do Tableau:

- Usando os cálculos de tabela, talvez você tenha tentado encontrar o primeiro ou o último período de uma partição. Por exemplo, calcular o número de funcionários de uma organização no primeiro dia de cada mês.
- Usando cálculos de tabela, campos calculados e linhas de referência, talvez você tenha tentado colocar campos agregados em um compartimento. Por exemplo, encontrando a média de um número distinto de clientes.
- Usando a combinação de dados, talvez você tenha tentado encontrar uma filtragem de dados relativa referente à data máxima nos dados. Por exemplo, se seus dados forem atualizados semanalmente, calcular os totais do ano até a data, de acordo com a data máxima.

As expressões de LOD são geradas como parte da consulta à fonte de dados subjacente. Elas são expressas como uma seleção aninhada, por isso dependem do desempenho do DBMS:

```
SELECT T1.[State],SUM(T2.[Sales per County, State])
FROM [DB Table] T1 INNER JOIN
    (SELECT [State], [County], AVG([Sales]) AS [Sales per County, State]
    FROM [DB Table] GROUP BY [State],[County]) T2
ON T1.[State] = T2.[State]
GROUP BY T1.[State]
```

Isso significa que, em alguns cenários, pode ser melhor solucionar um problema usando uma abordagem ou a outra. Por exemplo, um cálculo de tabela ou uma combinação de dados pode ter um desempenho melhor do que uma expressão de LOD e vice-versa. Se você achar que uma expressão de LOD está deixando o desempenho lento, tente substituí-la por um cálculo de tabela ou por uma combinação de dados (se possível) para ver se o desempenho melhora. As expressões de LOD também podem ser bastante afetadas pela separação de união. Nesse caso, sugerimos que você volte e releia a seção correspondente se achar que suas consultas estão lentas com o uso de expressões de LOD.

Para entender melhor o mecanismo das expressões de LOD, leia o whitepaper “Entendendo as expressões de nível de detalhe (LOD)”:

<http://www.tableau.com/pt-br/learn/whitepapers/understanding-lod-expressions>

Leia também o post “15 principais expressões de LOD” do blog da Bethany Lyons, que fornece exemplos de vários problemas comuns:

<http://www.tableau.com/pt-br/about/blog/LOD-expressions>

Há muitos outros posts do blog da comunidade sobre esse tópico que podem ser extremamente úteis. A *data + science* fornece uma ótima seleção, aqui:

<http://bit.ly/1MpkFV5>

Cálculos de tabela

Os cálculos de tabela, diferentemente dos cálculos no nível da linha e de agregação, não são executados pelo banco de dados, mas calculados pelo Tableau no conjunto de resultados de consultas. Embora isso signifique mais trabalho para o Tableau, esses cálculos geralmente são feitos com um conjunto de dados muito menor do que a fonte de dados original.

Se o desempenho do cálculo de tabela for um problema (possivelmente porque o conjunto de resultados retornado para o Tableau é muito grande), considere colocar alguns dos aspectos do cálculo na camada da fonte de dados. Uma forma de fazer isso é usar uma expressão de nível de detalhe. Outra forma seria aproveitar uma extração de dados agregada. Imagine um exemplo em que você deseja encontrar a média semanal ou o total de vendas diário de várias lojas. É possível fazer isso com um cálculo de tabela usando:

```
WINDOW_AVG(SUM([Sales]))
```

No entanto, se o número de dias/lojas for muito grande, esse cálculo pode tornar-se lento. Para colocar o cálculo SUM([Sales]) de volta na camada de dados, crie uma extração agregada que inclua a dimensão Date (Data) no nível do dia. O cálculo pode ser feito com MED([Sales]), visto que a extração já materializou os totais diários.

Em alguns casos, sabemos que o valor do fator agregado não muda na partição. Quando isso acontecer, use MIN() ou MAX() como a função de agregação, em vez de AVG() ou ATTR(), pois são

mais fáceis de avaliar. Depois de escolher uma, use-a sempre para aumentar suas chances de obter um resultado do cache!

Cálculos externos

O Tableau tem processos que permitem transmitir uma lógica complexa para a realização de um cálculo em um mecanismo externo, incluindo fazer chamadas para o R (usando uma conexão com o Rserve) ou para o Python (usando uma conexão com o servidor Dato).

No Tableau, essas operações são semelhantes aos cálculos de tabela, logo são calculadas em partições. Isso significa que elas podem ser chamadas várias vezes para uma única visualização, que pode ter seu desempenho sobrecarregado. Não há qualquer otimização ou mesclagem dessas operações, por isso, veja se é possível retornar diversos valores em uma única chamada de função (como em uma string concatenada) em vez de fazer várias chamadas de função. Por fim, o tempo gasto para transferir os dados de/para um mecanismo de computação externo também pode gerar uma sobrecarga.

Análises

O painel Análise fornece acesso rápido a algumas análises avançadas, como:

- Totais
- Linhas de referência
- Linhas de tendência
- Previsões
- Clustering (novo no Tableau 10)

Geralmente, essas análises não exigem a execução de consultas adicionais; é como se elas fossem cálculos de tabela realizados nos dados armazenados no cache de resultados de consultas. E, como acontece com os cálculos de tabela, se você tiver um conjunto de dados muito grande nos resultados de consultas, o processamento do cálculo pode demorar um pouco, mas, no geral, isso não contribuirá muito para que a pasta de trabalho tenha um desempenho ruim.

Outro elemento que pode afetar o desempenho das análises de totais e de linhas de referência é se as agregações de medidas são aditivas ou não aditivas. Para as agregações aditivas, como SUM, MIN e MAX, o cálculo dos totais ou das linhas de referências pode ser feito localmente no Tableau. Para as agregações não aditivas, como COUNTD, TOTAL, MEDIAN e PERCENTILE, precisamos acessar a fonte de dados para calcular os valores de total/linha de referência.

Cálculos x recursos nativos

Às vezes, os usuários criam campos calculados para executar funções que podem ser executadas facilmente com recursos nativos do Tableau. Por exemplo:

- Para agrupar membros de dimensões – considere usar [grupos](#) ou [conjuntos](#).
- Para agrupar valores de medida em faixas de intervalos – considere usar [compartimentos](#).
- Para truncar datas com uma granularidade maior (por exemplo, mês ou semana) – considere usar [campos de data personalizada](#).
- Para criar um conjunto de valores que seja a concatenação de duas dimensões diferentes – considere usar [campos combinados](#).
- Para exibir datas em um formato específico ou converter valores numéricos em indicadores de KPI – considere usar os recursos de formatação integrados.
- Para alterar os valores exibidos para os membros das dimensões – considere usar [alias](#).

Isso nem sempre é possível (por exemplo, talvez seja necessário usar compartimentos com larguras variáveis, o que não é possível com compartimentos básicos), mas considere usar os recursos nativos sempre que possível. Tomar essa medida é geralmente mais eficiente do que fazer um cálculo manual, e como os nossos desenvolvedores continuam melhorando o desempenho do Tableau, você se beneficiará desses esforços.

Impacto dos tipos de dados

Ao criar campos calculados, é importante entender que o tipo de dados usado tem um impacto significativo na velocidade do cálculo. Como regra geral

- Inteiros e booleanos são muito mais rápidos do que strings e datas

Os cálculos com strings e datas são muito lentos e, geralmente, contêm 10 a 100 instruções base que precisam ser executadas para cada cálculo. Em comparação a eles, os cálculos numéricos e booleanos são muito eficientes.

Essas declarações não são verdadeiras apenas para o mecanismo de cálculo do Tableau, mas para a maioria dos bancos de dados. Como os cálculos no nível da linha e de agregação são enviados para o banco de dados, se forem expressos em uma lógica numérica, em vez de em uma lógica de string, eles serão executados com muito mais rapidez.

Técnicas de desempenho

Considere as seguintes técnicas para garantir que seus cálculos sejam o mais eficientes possível:

Use booleanos para cálculos com lógica básica

Se o seu cálculo produzir um resultado binário (sim/não, aprovado/reprovado, acima/abaixo), garanta que o resultado retornado seja booleano, e não uma string. Por exemplo:

```
IF [Date] = TODAY() THEN "Today"  
  ELSE "Not Today"  
END
```

Esse cálculo será lento porque está usando strings. Uma forma mais rápida seria retornar um booleano:

```
[Date] = TODAY()
```

Em seguida, use um alias para renomear os resultados TRUE e FALSE como "Today" (Hoje) e "Not Today" (Não é hoje).

Pesquisas de string

Imagine que você precisa mostrar todos os registros nos quais o nome do produto contém uma string de pesquisa. Você poderia usar um parâmetro para encontrar a string de pesquisa do usuário e, em seguida, criar o seguinte campo calculado:

```
IF FIND([Product Name],[Product Lookup]) > 0  
  THEN [Product Name]  
  ELSE NULL  
END
```

Esse cálculo é lento porque é uma forma ineficiente de testar a restrição. Uma forma melhor de fazer isso é usar a função CONTAINS porque ela será convertida em um SQL perfeito quando transmitida para o banco de dados:

```
CONTAINS([Product Name],[Product Lookup])
```

No entanto, nesse caso, a melhor solução seria não usar um campo calculado, mas usar um cartão de filtro com correspondência de curinga.

Parâmetros para cálculos condicionais

Uma técnica comum no Tableau é fornecer um parâmetro ao usuário, para que ele possa selecionar um valor que determinará como o cálculo será realizado. Por exemplo, imagine que você deseja permitir que o usuário final controle o nível de agregação de dados mostrado em uma exibição selecionando um dos valores possíveis. Várias pessoas poderiam criar um parâmetro de string:

```
Value
Year
Quarter
Month
Week
Day
```

Para depois usá-lo em um cálculo da seguinte forma:

```
CASE [Parameters].[Date Part Picker]
  WHEN "Year" THEN DATEPART('year', [Order Date])
  WHEN "Quarter" THEN DATEPART('quarter', [Order Date])
  ..
END
```

Uma maneira melhor seria usar a função interna DATEPART() e criar o cálculo da seguinte forma:

```
DATEPART(LOWER([Parameters].[Date Part Picker]), [Order Date])
```

Nas versões anteriores do Tableau, a última abordagem era muito mais rápida. No entanto, não há mais diferença entre essas duas soluções, porque reduzimos a lógica condicional da instrução CASE com base no valor do parâmetro e apenas transmitimos a função DATEPART() apropriada para a fonte de dados.

Não se esqueça de que você precisa considerar a manutenção futura da solução e, nesse caso, o último cálculo provavelmente será a melhor opção.

Conversão de data

Frequentemente, os usuários têm dados de data que não estão armazenados nos formatos de data nativos e podem ser uma string ou um carimbo de data e hora. A função DATEPARSE() facilita essas conversões. Basta você digitar uma string de formatação:

```
DATEPARSE("yyyyMMdd", [YYYYMMDD])
```

Observe que DATEPARSE() tem suporte apenas em um subconjunto de fontes de dados:

- Conexões não herdadas de arquivos de texto e do Excel
- MySQL
- Oracle
- PostgreSQL
- Extração de dados do Tableau

Se sua fonte de dados não oferecer suporte à função DATEPARSE(), uma alternativa para converter os dados em um formato de data reconhecido pelo Tableau é transformar o campo em uma string de data, por exemplo "2012-01-01" (prefira as strings ISO, porque elas admitem a internacionalização) e, em seguida, usá-la com a função DATE().

Se os dados originais forem um campo numérico, convertê-los em uma string e, em seguida, em um formato de data é muito ineficiente. É muito melhor manter os dados no formato numérico e usar a função DATEADD() e os valores literais de data para fazer o cálculo.

Um exemplo de cálculo lento pode ser (converter um campo numérico no formato ISO):

```
DATE (LEFT (STR ([YYYYMMDD]), 4)
      + "-" + MID (STR ([YYYYMMDD]), 4, 2)
      + "-" + RIGHT (STR ([YYYYMMDD]), 2))
```

Veja abaixo uma forma muito mais eficiente de fazer esse cálculo:

```
DATEADD ('day', INT ([yyyymmdd]) % 100 - 1,
         DATEADD ('month', INT ([yyyymmdd]) % 10000 / 100 - 1,
                 DATEADD ('year', INT ([yyyymmdd]) / 10000 - 1900,
                         #1900-01-01#)))
```

Uma maneira ainda melhor seria usar a função MAKEDATE(), se sua fonte de dados oferecer suporte a ela:

```
MAKEDATE (2004, 4, 15)
```

Observe que os ganhos em desempenho podem ser significativos com conjuntos de dados grandes. Em uma amostra com mais de um bilhão de registros, o primeiro cálculo demorou mais de quatro horas para ser concluído, enquanto o segundo precisou de aproximadamente um minuto.

Instruções lógicas

Faça um teste para saber qual é o primeiro resultado mais comum

Quando o Tableau avalia um teste lógico, ele para de analisar os resultados possíveis assim que encontra uma correspondência. Isso significa que você deve fazer um teste para saber qual é o primeiro resultado mais comum para que, na maioria dos casos avaliados, a análise pare após o primeiro teste.

Considere o exemplo abaixo. A lógica:

```
IF <unlikely_test>
  THEN <unlikely_outcome>
  ELSEIF <likely_test>
    THEN <likely_outcome>
  END
```

Será mais lenta de avaliar do que esta:

```
IF <likely_test>
  THEN <likely_outcome>
  ELSEIF <unlikely_test>
    THEN <unlikely_outcome>
  END
```

ELSEIF > ELSE IF

Quando você trabalhar com instruções lógicas complexas, lembre-se de que ELSEIF > ELSE IF. Isso acontece porque uma instrução IF aninhada calcula uma instrução CASE aninhada na consulta resultante, em vez de ser calculada como parte da primeira. Então, o campo calculado abaixo:

```
IF [Region] = "East" AND [Segment] = "Consumer"
  THEN "East-Consumer"
  ELSE IF [Region] = "East" and [Segment] <> "Consumer"
    THEN "East-All Others"
  END
END
```

Gera o seguinte código SQL com duas instruções **CASE** aninhadas e quatro testes condicionais:

```

(CASE
  WHEN ([[Global Superstore].[Region] = 'East')
  AND ([[Global Superstore].[Segment] = 'Consumer'))
  THEN 'East-Consumer'
  ELSE (CASE
    WHEN ([[Global Superstore].[Region] = 'East')
    AND ([[Global Superstore].[Segment] <> 'Consumer'))
    THEN 'East-All Others'
    ELSE CAST(NULL AS NVARCHAR)
  )
END)
END)

```

Esse código seria executado mais rápido com ELSEIF no lugar da instrução IF aninhada e um uso mais eficiente dos testes condicionais:

```

IF [Region] = "East" AND [Segment] = "Consumer"
  THEN "East-Consumer"
  ELSEIF [Region] = "East"
    THEN "East-All Others"
END

```

Isso resulta em uma única instrução CASE no código SQL:

```

(CASE
  WHEN ([[Global Superstore].[Region] = 'East')
  AND ([[Global Superstore].[Segment] = 'Consumer'))
  THEN 'East-Consumer'
  WHEN ([Global Superstore].[Region] = 'East')
  THEN 'East-All Others'
  ELSE CAST(NULL AS NVARCHAR)
END)

```

Mas ainda é mais rápido. Mesmo usando uma instrução IF aninhada, este é o uso mais eficiente dos testes condicionais:

```

IF [Region] = "East" THEN
  IF [Segment] = "Consumer"
    THEN "East-Consumer"
    ELSE "East-All Others"
  END
END

```

O código SQL resultante:

```

(CASE
  WHEN ([Global Superstore].[Region] = 'East')
  THEN (CASE
    WHEN ([Global Superstore].[Segment] = 'Consumer')
    THEN 'East-Consumer'
    ELSE 'East-All Others'
  )
  ELSE CAST(NULL AS NVARCHAR)
END)

```

Evite verificações de lógica redundantes

Pelo mesmo motivo, evite verificações de lógica redundantes. O cálculo

```

IF [Sales] < 10 THEN "Bad"
  ELSEIF [Sales] >= 10 AND [Sales] < 30 THEN "OK"
  ELSEIF [Sales] >= 30 THEN "Great"
END

```

Gera o seguinte SQL:

```

(CASE
  WHEN ([Global Superstore].[Sales] < 10)

```

```

        THEN 'Bad'
    WHEN ([Global Superstore].[Sales] >= 10)
    AND ([Global Superstore].[Sales] < 30)
        THEN 'OK'
    WHEN ([Global Superstore].[Sales] >= 30)
        THEN 'Great'
    ELSE CAST(NULL AS NVARCHAR)
END)

```

Ele seria mais eficiente se fosse escrito assim:

```

IF [Sales] < 10 THEN "Bad"
ELSEIF [Sales] >= 30 THEN "Great"
ELSE "OK"
END

```

Gerando o seguinte SQL:

```

(CASE
    WHEN ([Global Superstore].[Sales] < 10)
        THEN 'Bad'
    WHEN ([Global Superstore].[Sales] >= 30)
        THEN 'Great'
    ELSE 'OK'
END)

```

Muitas outras pequenas coisas

Há muitas outras pequenas coisas que você pode fazer para melhorar o desempenho. Veja as dicas abaixo:

- Trabalhar com a lógica de data pode ser complicado. Em vez de escrever testes elaborados com várias funções de data, como MONTH() e YEAR(), considere usar algumas das outras funções internas, como DATETRUNC(), DATEADD() e DATEDIFF(). Elas podem reduzir significativamente a complexidade da consulta gerada no banco de dados.
- Os valores de contagem distintos são um dos tipos de agregação mais lentos em praticamente todas as fontes de dados. Use a agregação COUNTD com moderação.
- Usar parâmetros com um amplo escopo de impacto (em uma instrução SQL, por exemplo) pode afetar o desempenho do cache.
- Filtrar cálculos complexos pode ocasionar a perda dos índices na fonte de dados subjacente.
- Use NOW() apenas se precisar da hora do carimbo de data e hora. Use TODAY() para cálculos de data.
- Lembre-se de que todos os cálculos básicos são transmitidos para a fonte de dados subjacente, até mesmo cálculos literais, como strings de rótulo. Se você precisar criar rótulos (para os cabeçalhos de colunas, por exemplo) e sua fonte de dados for muito grande, crie um arquivo de fonte de dados simples de texto ou do Excel com apenas um registro contendo essas informações, a fim de não sobrecarregar a fonte de dados grande. Isso é especialmente importante se sua fonte de dados utilizar procedimentos armazenados, Consulte [esta seção](#) para obter mais detalhes.

O problema são as minhas consultas?

Um dos recursos mais poderosos do Tableau é sua capacidade de usar os dados na memória (como as conexões de dados extraídos) e os dados locais (como as conexões de dados em tempo real). As conexões em tempo real são um recurso poderoso do Tableau, porque elas nos permitem aproveitar a capacidade de processamento existente na fonte de dados. No entanto, como agora dependemos do desempenho da plataforma dos dados de origem, é essencial gerarmos nossas consultas na fonte de dados da forma mais eficiente e otimizada possível.

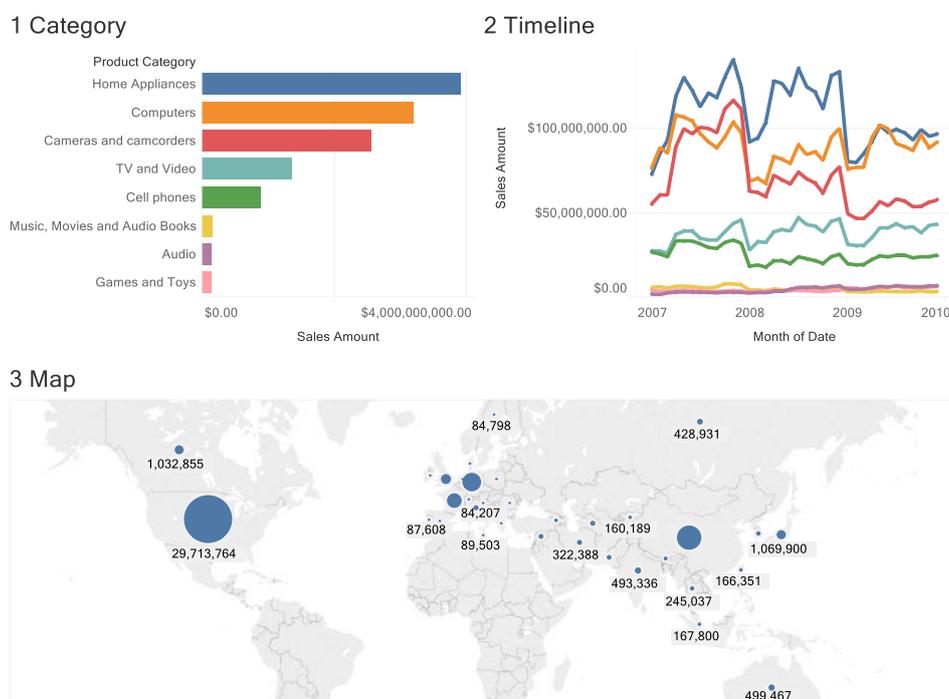
Como já discutimos antes, quanto mais complexa for a pergunta, quanto mais dados você exibir, quanto mais elementos você incluir no painel – tudo isso significa mais trabalho para a fonte de dados. Para aumentar ao máximo a eficiência das suas pastas de trabalho, precisamos minimizar o número de consultas, garantir que as consultas sejam avaliadas com a maior eficiência possível, minimizar a quantidade de dados retornada pelas consultas e reutilizar os dados o máximo possível entre as solicitações.

Otimizações automáticas

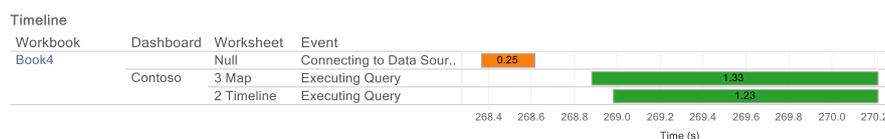
Podemos fazer várias coisas para melhorar o desempenho de nossas consultas, mas o Tableau realizará otimizações automaticamente para garantir que suas pastas de trabalho sejam executadas com eficiência. Você não pode controlar diretamente a maioria delas e, em grande parte dos casos, não precisa se preocupar com isso, mas, ao entendê-las bem, você poderá aproveitá-las para melhorar o desempenho.

Execução paralela – executar várias consultas ao mesmo tempo

O Tableau se beneficia da capacidade que os bancos de dados de origem têm de executar várias consultas ao mesmo tempo. Veja o painel abaixo:



Abrir esta pasta de trabalho no Tableau mostra as seguintes consultas:



Como você pode ver, executar as consultas de forma serial leva 2,66 segundos. Mas, ao executá-las em paralelo, temos apenas a duração da consulta mais longa (1,33 segundos).

O nível do paralelismo das consultas varia em cada sistema de origem, visto que algumas plataformas lidam melhor com consultas simultâneas do que outras. A abordagem padrão é executar um máximo de 16 consultas paralelas para todas as fontes de dados, exceto para arquivos de texto, do Excel e de estatísticas, que estão limitados a uma consulta por vez. É possível definir limites abaixo do padrão para outras fontes de dados. Consulte o link abaixo para obter detalhes:

<http://tabsoft.co/1TjZvxo>

Na maioria dos casos, não é necessário alterar essas configurações, você pode manter suas definições padrão; mas se você tiver uma necessidade específica de controlar o grau de paralelismo, ele pode ser definido como:

- Um limite global no número de consultas paralelas do Tableau Server.
- Limites para um determinado tipo de fonte de dados, como o SQL Server.
- Limites para um determinado tipo de fonte de dados em um servidor específico.
- Limites para um determinado tipo de fonte de dados, em um servidor específico, ao estabelecer uma conexão com um banco de dados específico.

Essas configurações são gerenciadas por um arquivo xml chamado connection-configs.xml, que você cria e salva na pasta app no Windows (C:\Program Files\Tableau\Tableau 10.0) e no Mac (clique com o botão direito do mouse em App, selecione Mostrar conteúdo do pacote e salve o arquivo aqui) para o Tableau Desktop ou no diretório config na pasta vizqlserver (por exemplo:

C:\ProgramData\Tableau\TableauServer\data\tabsvc\config\vizqlserver) para o Tableau Server. Copie este arquivo de configuração para todos os diretórios de configuração do vizqlserver em todos os computadores de trabalho.

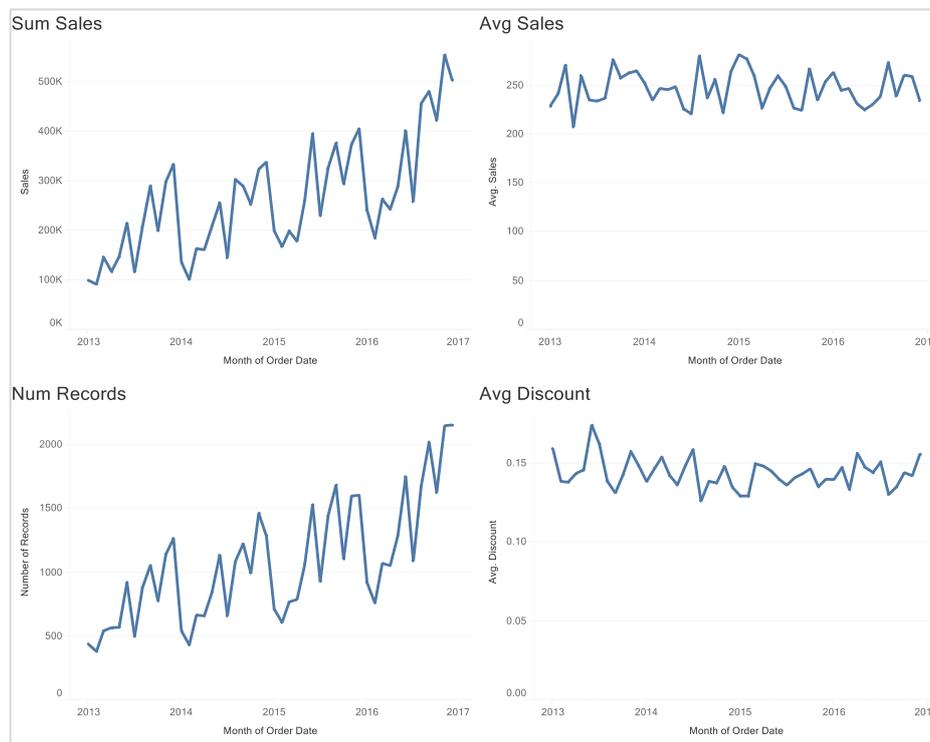
Para obter mais informações sobre como configurar consultas paralelas, inclusive a sintaxe do arquivo connection-configs.xml file, acesse:

<http://kb.tableau.com/articles/knowledgebase/parallel-queries-tableau-server?lang=pt-br>

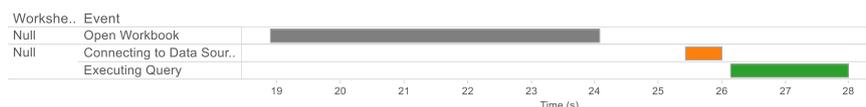
Eliminação de consulta – executando menos consultas

Você também pode ver no exemplo acima que executamos apenas duas consultas, em vez de três. Ao criar um lote de consultas, o Tableau consegue eliminar consultas redundantes. O otimizador de consultas do Tableau classificará as consultas para executar as mais complexas primeiro esperando que as próximas possam ser obtidas no cache de resultados de consultas. No exemplo, como a linha do tempo inclui Categoria do produto e a agregação SUM de Valor das vendas é totalmente aditiva, os dados do gráfico Categoria podem ser obtidos do cache de consultas da planilha Linha do tempo, eliminando a necessidade de encontrar uma correspondência na fonte de dados.

O otimizador de consulta também pesquisará as consultas que estão no mesmo nível de detalhe (aquelas especificadas pelo mesmo conjunto de dimensões) e as mesclará em uma única consulta que retorna todas as medidas solicitadas. Veja o painel abaixo:



Como você pode ver, este painel contém quatro planilhas, e cada uma delas mostra uma medida diferente ao longo do tempo. Todas elas têm o mesmo nível de detalhe, visto que mostram os dados usando um mês contínuo. Em vez de executar quatro solicitações separadamente, o Tableau as combina em uma única consulta:



A consulta é a seguinte:

```
SELECT AVG(cast([Global Superstore].[Discount] as float)) AS
[avg:Discount:ok],
  AVG(cast([Global Superstore].[Sales] as float)) AS [avg:Sales:ok],
  SUM(CAST(1 as BIGINT)) AS [sum:Number of Records:ok],
  SUM([Global Superstore].[Sales]) AS [sum:Sales:ok],
  DATEADD(month, DATEDIFF(month, CAST('0001-01-01 00:00:00' AS datetime2),
[Global Superstore].[Order Date]), CAST('0001-01-01 00:00:00' AS datetime2))
AS [tmn:Order Date:ok]
FROM [dbo].[Global Superstore] [Global Superstore]
GROUP BY DATEADD(month, DATEDIFF(month, CAST('0001-01-01 00:00:00' AS
datetime2), [Global Superstore].[Order Date]), CAST('0001-01-01 00:00:00' AS
datetime2))
```

Como você pode ver, essa otimização (chamada “fusão de consulta”) pode melhorar significativamente o desempenho geral. Sempre que possível, defina o mesmo nível de detalhe para as várias planilhas de um painel.

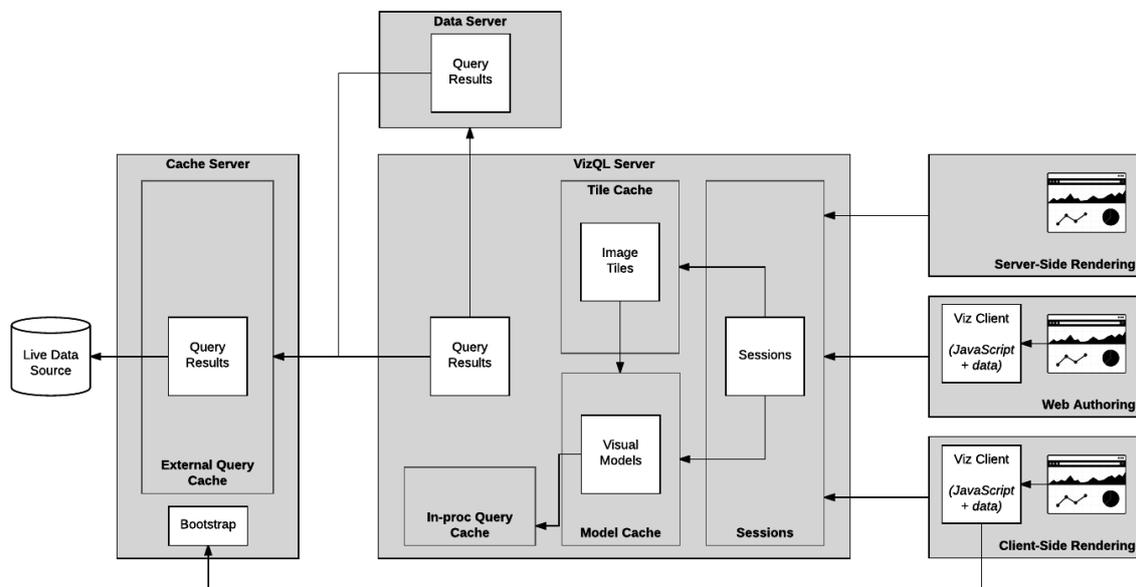
Observe que a fusão de consultas não é feita para fontes de dados TDE.

Usando o cache para não executar nenhuma consulta

O que é melhor do que executar poucas consultas? Não executar nenhuma! O Tableau oferece um recurso de cache extensivo no Tableau Desktop e no Tableau Server que pode reduzir significativamente o número de consultas que precisam ser executadas na fonte de dados subjacente.

Tableau Server

O armazenamento em cache do Tableau Server tem várias camadas:



A primeira camada do armazenamento em cache depende do tipo de renderização usado pela sessão; renderização no cliente ou no servidor. Consulte esta [seção anterior](#) para obter mais informações sobre os dois modelos de renderização e sobre como o Tableau determina qual delas usará.

Se a sessão estiver usando a renderização no cliente, a primeira coisa que o navegador precisa fazer é carregar o cliente do visualizador. Isso envolve o carregamento dos dados e de um conjunto de bibliotecas do JavaScript para que a exibição inicial seja renderizada. Esse processo se chama “bootstrap” e pode demorar alguns segundos. Sabendo que os painéis provavelmente serão visualizados por diversos usuários e para reduzir o tempo de espera das solicitações de exibição seguintes, as respostas do bootstrap são armazenadas em cache. Cada sessão verifica primeiro o cache do bootstrap para ver se já existe uma resposta que possa ser usada. Se ela encontrar uma, o navegador carrega essa resposta a partir do cache, agilizando bastante a renderização da exibição inicial. Depois que o cliente do visualizador for carregado no navegador do cliente, algumas interações poderão ser realizadas integralmente com os dados locais (como usar destaques e dicas de ferramentas), resultando em uma experiência de usuário ágil e suave.

Se o painel estiver sendo exibido com uma renderização no lado do servidor, o servidor renderizará os elementos do painel como uma série de arquivos de imagem (chamados de “blocos”). Esses blocos são transmitidos para o navegador onde são organizados para exibir a visualização. Conforme mencionado acima, esperamos que esses painéis sejam exibidos por diversos usuários, por isso, o servidor armazena essas imagens em cache no disco. Cada solicitação pesquisa o cache de blocos para saber se uma determinada imagem já foi renderizada. Se a imagem for encontrada, o arquivo do cache será carregado. Uma correspondência no cache de blocos agiliza o tempo de resposta e reduz a carga de trabalho no

servidor. Uma forma simples de aumentar a eficiência do cache de blocos é configurar seus painéis para que eles tenham um [tamanho fixo](#).

De uma forma geral, a renderização no cliente oferece interações de usuário mais suaves e responsivas e reduz a carga de trabalho no Tableau Server. Sempre que possível, crie suas pastas de trabalho para renderizá-las no lado do cliente.

A próxima camada do armazenamento em cache é chamada de modelo visual. Um modelo visual descreve como renderizar uma planilha individual para que a visualização de um painel possa consultar diversos modelos visuais – um para cada planilha usada. Ele inclui os resultados de cálculos locais (como cálculos de tabela, linhas de referência, previsões, clustering, etc.) e o layout visual (quantas linhas/colunas serão exibidas para múltiplos pequenos e tabelas de referência cruzada, o número e o intervalo das escalas de eixo/linhas de grade a serem representados no gráfico, o número e o local dos rótulos de marca que serão mostrados, etc.).

Os modelos visuais são criados e armazenados na memória pelo VizQL Server, que faz o seu melhor para compartilhar os resultados nas sessões de usuários sempre que possível. Os principais fatores que definem se um modelo visual pode ser compartilhado são:

- O tamanho da área de exibição da visualização – os modelos só podem ser compartilhados em sessões que tenham visualizações com o mesmo tamanho. Configurar seus painéis para ter um tamanho fixo beneficia o cache de modelos e o cache de blocos, permitindo uma reutilização maior e reduzindo a carga de trabalho no servidor.
- Se as seleções ou os filtros correspondem – se o usuário estiver alterando filtros, parâmetros, detalhando os dados, etc., o modelo será compartilhado apenas nas sessões com exibições no mesmo estado. Evite publicar pastas de trabalho com a opção “Mostrar seleções” marcada, porque isso pode reduzir a probabilidade de obter uma correspondência para sessões diferentes.
- As credenciais usadas para conectar à fonte de dados – se os usuários precisarem fornecer credenciais para se conectar a uma fonte de dados, o modelo só poderá ser compartilhado nas sessões de usuário com as mesmas credenciais. Use esse recurso com cuidado, pois ele pode reduzir a eficiência do cache de modelos.
- Se filtros de usuário estão sendo usados – se a planilha contiver filtros de usuário ou cálculos com funções, como USERNAME() ou ISMEMBEROF(), o modelo não será compartilhado com nenhuma outra sessão de usuário. Use essas funções com cuidado, pois elas podem reduzir significativamente a eficiência do cache de modelos.

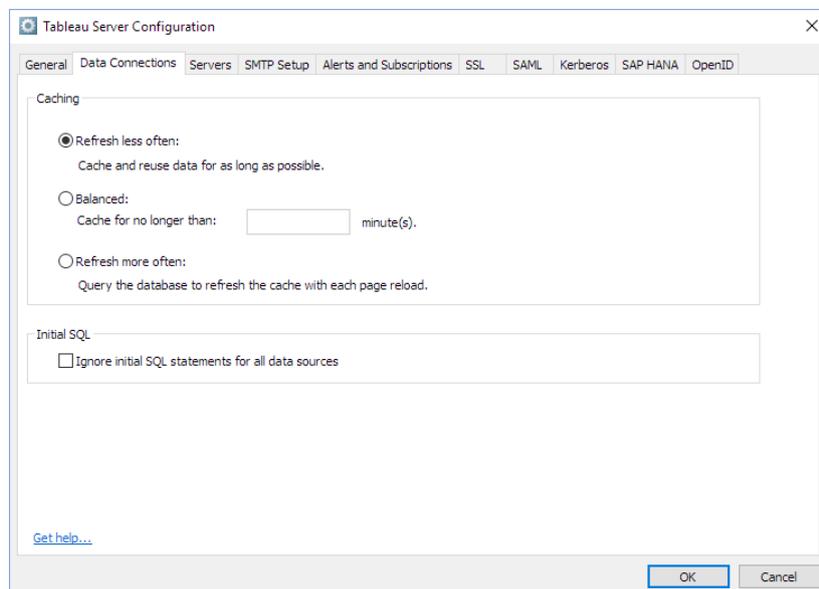
A última camada do armazenamento em cache é o cache de consultas. Ele armazena os resultados de consultas executadas anteriormente e que podem ser usados para responder a consultas futuras. Obter uma correspondência nesse cache confere muita eficiência ao processo, porque não precisamos executar as mesmas consultas repetidamente na fonte de dados, visto que os dados são carregados diretamente do cache. Em alguns casos, ele também nos permite responder a consultas usando os resultados de outra consulta. Falamos sobre os benefícios disso [anteriormente](#) quando abordamos a eliminação das consultas e a fusão de consultas.

Esse cache é dividido em duas partes: uma no processo do VizQL (chamado de “cache in-proc”) e outra compartilhada por vários processos por meio do Servidor cache (chamado de “cache externo”). Se uma consulta for enviada por uma instância do VizQL que já tenha executado a mesma consulta anteriormente, a solicitação é atendida pelo cache in-proc. Observe que esse cache é local para cada processo do VizQL e é mantido na memória.

Além desses caches in-proc, há um cache externo que é compartilhado, não apenas nas instâncias do VizQL, mas em TODOS os processos que acessam a fonte de dados subjacente (como processadores em segundo plano, servidores de dados, etc.). Um serviço chamado Servidor cache gerencia o cache externo em todo o cluster. Observe que nem todas as consultas são gravadas no cache externo. Se uma consulta for executada rapidamente, pode ser mais rápido executá-la de novo do que verificar o cache, por isso há um limite de tempo de consulta mínimo. Além disso, o resultado de uma consulta é muito grande, e pode não ser eficiente gravá-lo no Servidor cache, por isso também há um limite de tamanho máximo.

O cache externo aumenta significativamente a eficiência do armazenamento em cache em implantações com diversas instâncias do VizQL. Ao contrário dos caches in-proc, que são voláteis, o cache externo é persistente (o Servidor cache grava os dados no disco) e permanece entre as plataformas dos serviços.

Para maximizar a eficiência dos caches, você pode ajustar as configurações da instalação do Tableau Server para que eles mantenham seus conteúdos pelo maior tempo possível:



Se você tiver o recurso de servidor disponível, poderá aumentar o tamanho dos caches. Aumentar essas configurações não implica nenhuma penalidade. Portanto, se você tiver RAM suficiente, poderá aumentar bastante os números para evitar que o cache elimine conteúdo. Você pode monitorar a eficiência dos caches nos [arquivos de log](#) ou pelo [TabMon](#).

- **Cache de modelos** – a configuração padrão permite armazenar em cache 60 modelos por instância do VizQL. Você pode ajustar essa configuração com o comando *tabadmin*. Se houver RAM disponível, você pode aumentar esse número para que ele corresponda ao número de exibições publicadas em seu servidor:
 - `tabadmin set vizqlserver.modelcachesize 200`
- **Cache in-proc** – a configuração padrão permite armazenar em cache 512 MB de resultados de consultas por instância do VizQL. Isso pode não parecer muito, mas lembre-se de que estamos armazenando os resultados de consultas originados de consulta agregadas. Você pode ajustar essa configuração com o comando *tabadmin*:
 - `tabadmin set vizqlserver.querycachesize 1024`

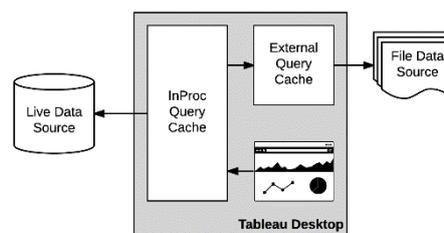
Você também pode aumentar a capacidade e a taxa de transferência do cache de consultas externo adicionando mais instâncias do Servidor cache. A prática recomendada é executar uma instância do Servidor cache para cada instância do VizQL Server.

- **Cache externo** – a configuração padrão é armazenar em cache 512 MB de resultados de consultas para cada instância do Servidor cache.

Por fim, um dos benefícios úteis do compartilhamento do cache é que podemos aquecer o cache para as pastas de trabalho que demoram a abrir, executando uma assinatura da pasta de trabalho. Fazer isso no início do dia, antes que seus usuários cheguem ao trabalho, garantirá que a pasta de trabalho seja executada e que os resultados de consultas sejam carregados no cache. Considerando que os resultados não foram eliminados do cache ou não tenham expirado nesse período, o cache será acessado quando os usuários visualizarem a pasta de trabalho, tornando o tempo de exibição inicial mais rápido.

Tableau Desktop

O armazenamento em cache do Tableau Desktop é muito mais simples do que o do Tableau Server, porque ele é um aplicativo para um único usuário e não precisamos gerenciar várias sessões. O Tableau Desktop tem apenas o cache de consultas:



Como no Tableau Server, há um cache in-proc e um cache externo. O cache in-proc, que é volátil e fica na memória, é usado para estabelecer conexões com todas as fontes de dados. No entanto, o cache externo é usado apenas para fontes de dados criadas com arquivos (como arquivos de extrações de dados, do Excel, do Access, de texto, de estatística, etc.). Como no Tableau Server, o cache externo é persistente. Isso significa que os resultados armazenados no cache são mantidos entre as sessões no Tableau Desktop. Então, se você usar uma fonte de dados de arquivo, reiniciar o Tableau Desktop e usá-la novamente, você ainda poderá se beneficiar do cache.

O cache externo é armazenado em `%LOCALAPPDATA%\Tableau\Caching` no Windows e em `~/Library/Caches/com.tableau/` no Mac. Por padrão, ele tem um limite total de aproximadamente 750 MB e será invalidado se o usuário forçar uma atualização da fonte de dados (pressionando F5, ⌘R).

Por fim, os dados do cache de consultas externo do Tableau Desktop são incluídos quando o arquivo é salvo como uma pasta de trabalho em pacote. Isso permite que o Tableau Desktop e o Tableau Reader renderizem rapidamente a exibição inicial da pasta de trabalho, enquanto ela descompacta um arquivo de fonte de dados grande em segundo plano. Para um usuário final, isso pode melhorar muito a responsividade da abertura da pasta de trabalho. Observe que os dados do cache só serão incluídos se eles forem $\leq 10\%$ do tamanho do arquivo da fonte de dados, e as consultas que usam filtros de data relativa não serão incluídas.

Conexões ociosas

Em versões anteriores ao Tableau 10, quando o usuário abria uma pasta de trabalho com várias fontes de dados, ele primeiro se conectava a todas as fontes de dados que não exigiam credenciais (fontes de dados que não solicitam um nome de usuário e senha ao usuário). Isso significava que poderíamos estar perdendo tempo nos conectando a fontes de dados que nem seriam usadas na planilha, no painel ou na história que o usuário estava visualizando.

No Tableau 10, conectamos a uma fonte de dados apenas quando um usuário precisa dela para visualizar uma planilha, um painel ou uma história selecionada. Essa mudança reduz o tempo de carregamento de uma pasta de trabalho com uma exibição que contém guias (no Tableau Desktop), para que os usuários possam começar a explorar os dados com mais rapidez.

Uniões

Normalmente, quando você trabalha com várias tabelas no Tableau, a abordagem preferencial é definir as uniões na janela de conexão de dados. Ao fazer isso, você não está definindo uma consulta específica, mas apenas definindo a relação entre as tabelas. Quando você arrasta e solta campos em uma visualização, o Tableau usa essas informações para gerar a consulta específica necessária para coletar apenas os dados solicitados.

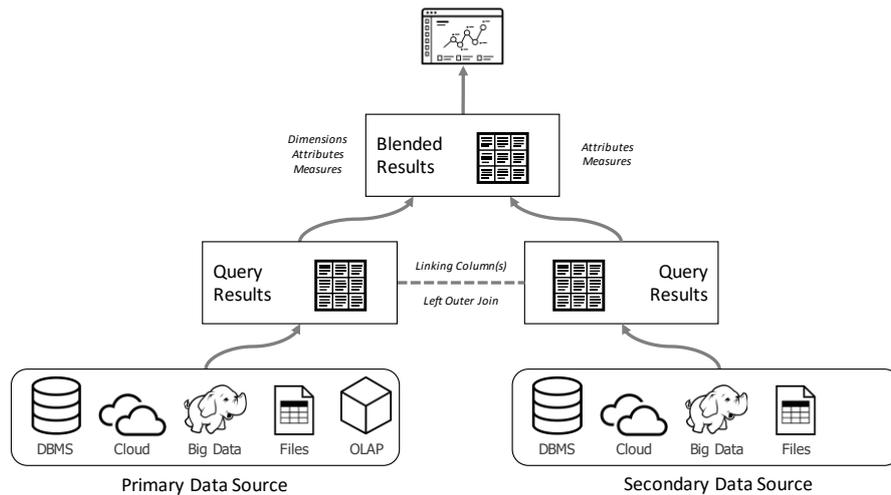
Como regra geral, para garantir um bom desempenho, o número de tabelas unidas deve ser minimizado para incluir apenas as tabelas necessárias para uma planilha/visualização específica. Dependendo da complexidade da conexão de dados e das planilhas, talvez valha a pena separar as conexões de dados para cada planilha e criar padrões de união específicos para essas planilhas.

As uniões funcionam melhor quando são adequadamente restringidas entre as tabelas. As restrições permitem que o Tableau Desktop simplifique livremente as consultas de forma a usar apenas as tabelas necessárias para responder a determinados subconjuntos de uma pergunta (legendas, cartões de filtro).

Combinação

Antes de escolher entre unir ou combinar tabelas de dados no Tableau, considere a origem dos dados, o número de conexões de dados e o número de registros presentes nos dados.

Quando você usa a combinação, uma das principais influências no desempenho da combinação não é o número de registros em cada fonte de dados, mas a cardinalidade dos campos combinados que vinculam os dois conjuntos de dados. A combinação consulta os dados das duas fontes de dados no nível dos campos de vinculação e, em seguida, mescla os resultados das duas consultas na memória local do Tableau.



Se houver muitos valores exclusivos, isso pode exigir uma grande quantidade de memória. É recomendável usar a versão de 64 bits do Tableau Desktop ao combinar dados para evitar consumir toda a memória. No entanto, combinações desse tipo provavelmente continuarão demorando para serem concluídas.

A prática recomendada é evitar a combinação em dimensões com uma grande quantidade de números de valores exclusivos (por exemplo, ID do pedido, ID do cliente, data/hora exatas, etc.).

Grupos e alias principais

Se for necessário combinar duas fontes de dados porque uma contém registros de “fatos” e a outra contém outros atributos dimensionais, talvez seja possível melhorar o desempenho criando um grupo ou alias principal.

Crie os grupos e/ou os aliases na fonte de dados primária para refletir os atributos listados na fonte de dados secundária. Os grupos principais são usados para relações 1:muitos e os aliases principais são usados para relações 1:1. Então, você poderá usar o objeto relevante da fonte de dados primária, eliminando a necessidade de fazer a combinação no momento da exibição.

Para os exemplos abaixo, considere as três tabelas a seguir:

ID	Value
A	89
B	94
C	74
D	88
E	58
F	89
G	95

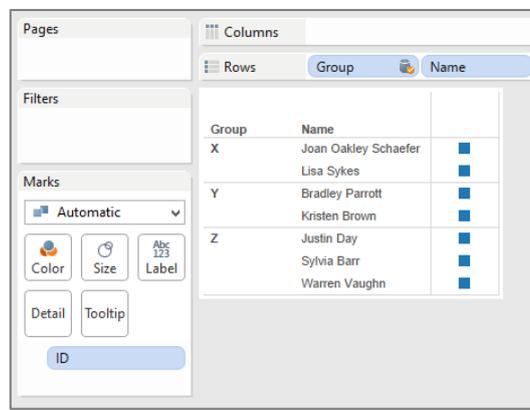
ID	Name
A	Lisa Sykes
B	Joan Oakley Schaefer
C	Bradley Parrott
D	Kristen Brown
E	Sylvia Barr
F	Warren Vaughn
G	Justin Day

ID	Group
A	X
B	X
C	Y
D	Y
E	Z
F	Z
G	Z

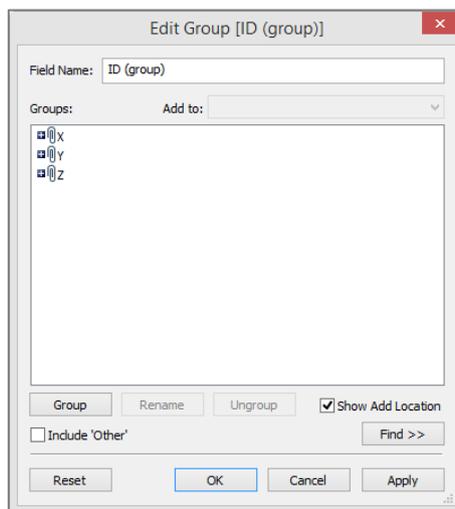
Os grupos principais são úteis quando a fonte de dados secundária contém um atributo com mapeamento 1:muitos para membros da dimensão na fonte de dados primária. Considere que o que queremos produzir a partir dos dados acima é o seguinte:

Group	Name
X	Lisa Sykes
	Joan Oakley Schaefer
Y	Bradley Parrott
	Kristen Brown
Z	Sylvia Barr
	Warren Vaughn
	Justin Day

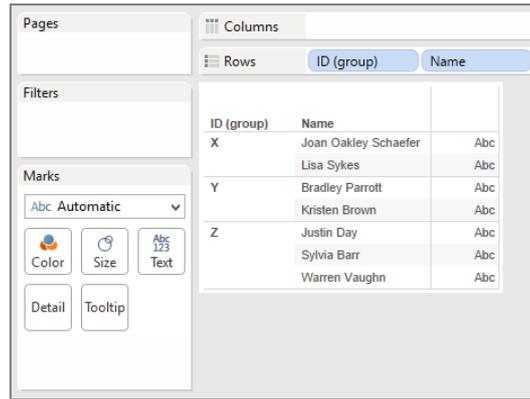
Podemos criar isso com a combinação, mas, como discutimos anteriormente, isso poderia resultar em um desempenho muito ruim, caso haja um número muito grande de IDs:



Clique com o botão direito do mouse em Grupo e selecione “Criar grupo primário”; o Tableau criará um objeto de grupo na fonte de dados primária que mapeia o campo de vinculação (nesse caso, a ID) para a dimensão da fonte de dados secundária selecionada (nesse caso, Grupo).



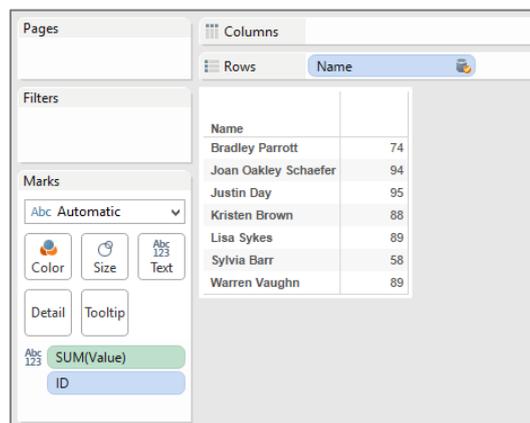
Agora, podemos recriar essa tabela sem precisar de uma combinação:



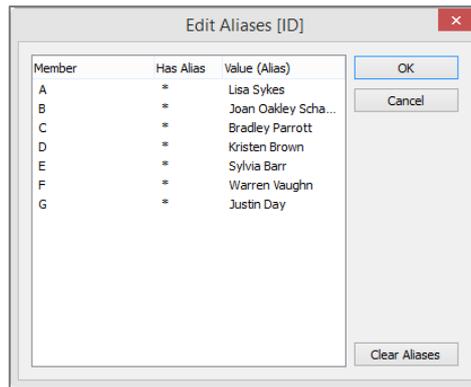
Os alias principais são úteis quando a fonte de dados secundária contém um atributo com mapeamento 1:1 para membros da dimensão na fonte de dados primária. Considere que o que queremos produzir a partir dos dados acima é o seguinte:

Name	Value
Lisa Sykes	89
Joan Oakley Schaefer	94
Bradley Parrott	74
Kristen Brown	88
Sylvia Barr	58
Warren Vaughn	89
Justin Day	95

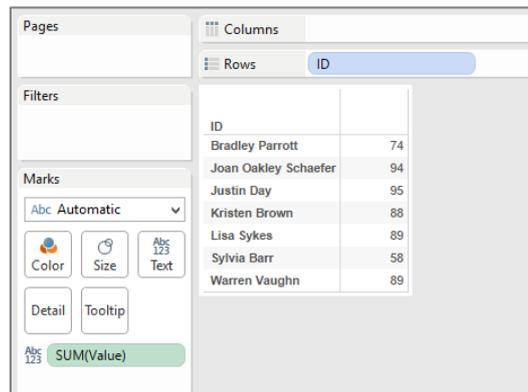
Podemos criar isso com a combinação de duas fontes de dados, mas como discutimos anteriormente, isso poderia resultar em um desempenho muito ruim, caso haja um número muito grande de IDs:



Clicar com o botão direito do mouse no campo Nome e selecionar “Editar aliases primários” permite o mapeamento único do campo Nome ao campo ID como valores de alias:



Agora, podemos criar a visualização necessária sem uma combinação, o que é muito mais rápido:



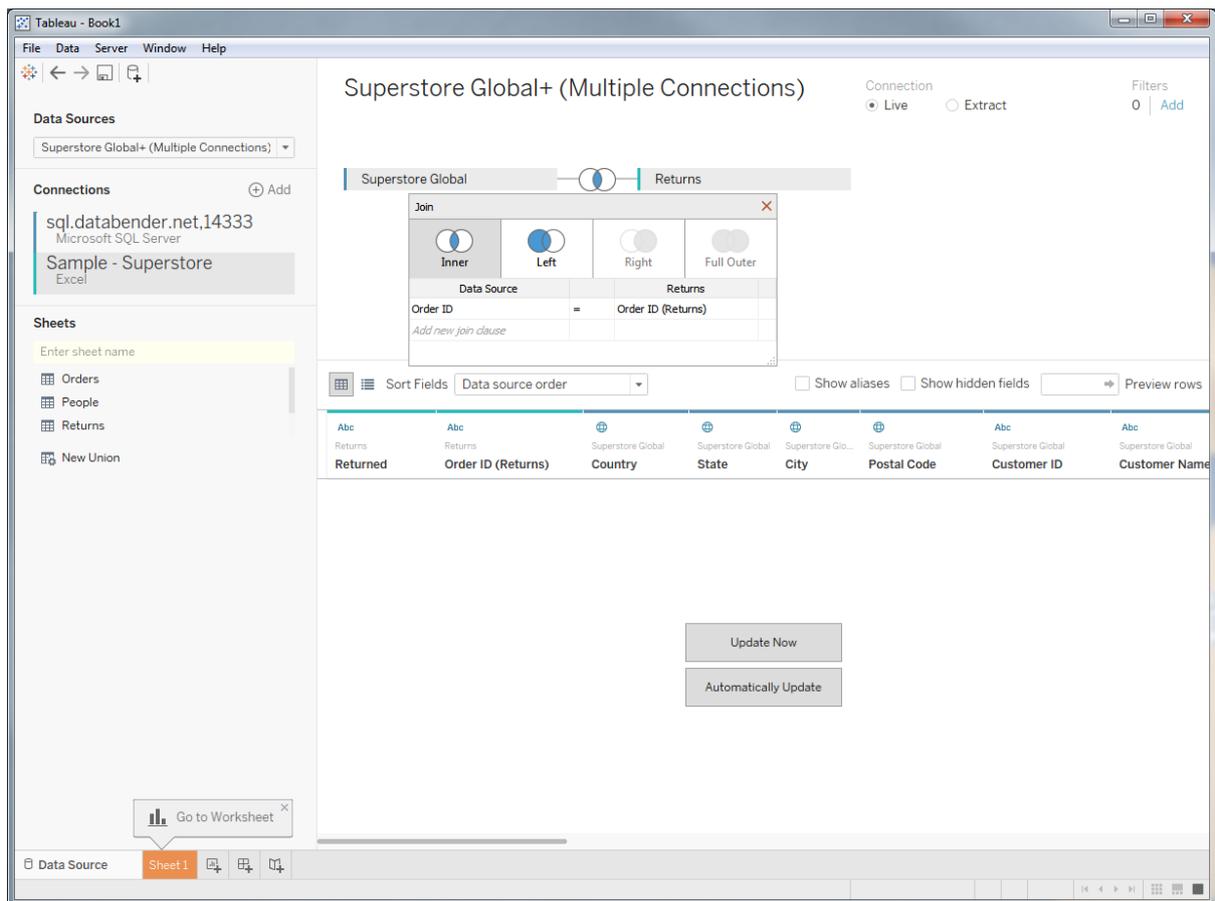
Observe que os grupos e alias primários não são dinâmicos e precisarão ser atualizados se os dados forem alterados. Por isso, eles não são a melhor solução para dados atualizados com frequência, mas se você precisar de um mapeamento rápido, eles podem eliminar a necessidade de combinações trabalhosas.

Para obter mais informações, consulte este artigo da base de dados de conhecimentos:

http://onlinehelp.tableau.com/current/pro/desktop/pt-br/help.htm#multipleconnections_create_primary_group.html

Integração de dados

A integração de dados é um recurso novo que foi introduzido no Tableau 10. Ele permite que as fontes de dados combinem dados de várias conexões de dados potencialmente heterogêneas.



Uma das principais diferenças entre a integração de dados e a combinação de dados é que a primeira é uma união feita no nível da linha e a segunda é feita no conjunto dos resultados agregados de cada fonte de dados. Isso significa que a integração de dados é afetada pelo tamanho das fontes de dados subjacentes. Há quatro pontos importantes a considerar:

- Quanto mais dados precisarmos mover, maior será o tempo de processamento – os dados são extraídos de cada conexão de dados, no nível da linha, por isso o volume de dados é um fator importante.
- Quanto mais longo for o percurso dos dados, maior será o tempo de processamento – o desempenho será afetado se você estiver unindo uma fonte de dados com uma conexão de alta latência.
- Quanto mais lento for o fluxo dos dados, maior será o tempo de processamento – o desempenho será afetado se você estiver unindo uma fonte de dados com uma conexão que tenha uma largura de banda restrita.
- Quanto mais correspondências forem necessárias, maior será o tempo de processamento – de forma semelhante ao primeiro ponto, o desempenho é afetado pelo número de registros que incluímos na união.

No Tableau 10, a integração de dados só pode ser usada com fontes de dados extraídas, ou seja, os resultados de cada conexão de dados devem ser extraídos para um arquivo TDE. Então, essa fonte de dados extraída poderá ser publicada no Tableau Server e utilizada por outras pessoas.

No Tableau 10, não é possível usar a integração de dados com fontes de dados conectadas em tempo real nem com fontes de dados publicadas. Pretendemos tornar isso possível no futuro.

SQL personalizado

Às vezes, novos usuários do Tableau tentam utilizar técnicas antigas com suas pastas de trabalho, como criar fontes de dados usando instruções SQL escritas manualmente. Em muitos casos, isso é contraproducente, visto que o Tableau pode gerar consultas muito mais eficientes quando definimos as relações de união entre as tabelas e deixamos o mecanismo de consulta escrever um SQL específico para a exibição que está sendo criada. Porém, há situações em que especificar uniões na janela de conexão de dados não oferece toda a flexibilidade de que você precisa para definir as relações em seus dados.

Criar uma conexão de dados usando uma instrução SQL escrita manualmente pode ser poderoso, mas, como aprendemos no filme do Homem-Aranha, “grandes poderes trazem grandes responsabilidades”. Em alguns casos, o SQL personalizado pode afetar negativamente o desempenho. Isso acontece porque, ao contrário da definição de uniões, o SQL personalizado nunca é desfeito e é sempre executado atômicamente. Isso significa que não há uma separação de união e que o banco de dados pode acabar tendo que processar a consulta inteira apenas para uma única coluna, por exemplo:

```
SELECT SUM([TableauSQL].[Sales])
FROM (
    SELECT [OrdersFact].[Order ID] AS [Order ID],
           [OrdersFact].[Date ID] AS [Date ID],
           [OrdersFact].[Customer ID] AS [Customer ID],
           [OrdersFact].[Place ID] AS [Place ID],
           [OrdersFact].[Product ID] AS [Product ID],
           [OrdersFact].[Delivery ID] AS [Delivery ID],
           [OrdersFact].[Discount] AS [Discount],
           [OrdersFact].[Cost] AS [Cost],
           [OrdersFact].[Sales] AS [Sales],
           [OrdersFact].[Qty] AS [Qty],
           [OrdersFact].[Profit] AS [Profit]
    FROM [dbo].[OrdersFact] [OrdersFact]
) [TableauSQL]
HAVING (COUNT_BIG(1) > 0)
```

É importante garantir que sua instrução SQL personalizada não contenha cláusulas desnecessárias. Por exemplo, se sua consulta contiver as cláusulas GROUP BY ou ORDER BY, isso provavelmente causará uma sobrecarga, porque o Tableau criará suas próprias cláusulas com base na visualização. Se for possível, elimine essas cláusulas da sua consulta.

Uma boa recomendação é usar o SQL personalizado junto com extrações de dados. Dessa forma, a consulta atômica é executada apenas uma vez (para carregar os dados na extração de dados), e toda a análise subsequente no Tableau será feita por meio de consultas dinâmicas otimizadas na extração de dados. É claro que todas as regras têm exceções, e essa não é diferente: quando você usar um SQL personalizado, crie uma extração de dados, A MENOS QUE seu SQL personalizado contenha parâmetros.

Em alguns casos, usar parâmetros em uma instrução de SQL personalizado pode melhorar o desempenho das conexões em tempo real, visto que a consulta base pode ser mais dinâmica (por exemplo, as cláusulas de filtro que usam parâmetros serão avaliadas apropriadamente). Os parâmetros também podem ser usados para transmitir valores para limitadores de desempenho, como TOP ou SAMPLE, e restringir a quantidade de dados retornada pelo banco de dados. No entanto, se você estiver usando uma extração de dados, ela será depurada e regenerada cada vez que você alterar o parâmetro, e esse processo pode ser lento. Lembre-se de que os parâmetros podem ser usados apenas para transmitir valores literais, logo não podem ser usados para alterar dinamicamente as cláusulas SELECT ou FROM.

Por fim, também é possível unir tabelas a um SQL personalizado:

Custom SQL Query+ (Superstore)

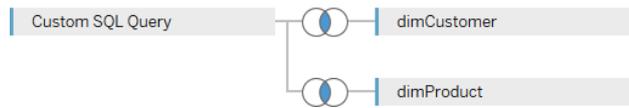
Connection

Live

Extract

Filters

0 | Add



Isso permite que você escreva um SQL personalizado mais específico que consulte um subconjunto do esquema total. As uniões desta com outras tabelas podem (potencialmente) ser separadas como uniões de tabela comuns, criando consultas mais eficientes.

Alternativas ao SQL personalizado

Em vez de usar um SQL personalizado diretamente no Tableau, às vezes é preferível transferir a consulta para a fonte de dados subjacente. Em muitos casos, essa medida melhora o desempenho, porque permite que a fonte de dados analise a consulta com mais eficiência, ou pode significar que uma consulta complexa só precise ser executada uma vez. Ela também é adequada às práticas recomendadas de gerenciamento, porque permite que uma única definição seja compartilhada em várias pastas de trabalho e fontes de dados.

Há diversas maneiras de fazer isso:

Exibições

Praticamente todos os DBMS oferecem suporte ao conceito de exibição – uma tabela virtual que representa o resultado de uma consulta de banco de dados. Com alguns sistemas de banco de dados, basta remover a consulta de uma instrução do SQL personalizado e instanciá-la no banco de dados como uma exibição para que o desempenho seja significativamente melhor. Isso acontece porque o otimizador de consulta é capaz de gerar um plano de execução mais eficiente do que quando a consulta está incluída em uma instrução SELECT externa. Para obter mais detalhes, consulte a seguinte discussão da comunidade:

<http://tabsoft.co/1su5YMe>

Definir uma lógica de consulta personalizada em uma exibição, e não na pasta de trabalho do Tableau, também permite que ela seja reutilizada em várias pastas de trabalho e fontes de dados. Além disso, muitos DBMSs oferecem suporte ao conceito de exibições materializadas ou instantâneas, onde os resultados da consulta da exibição são pré-processados e armazenados em cache, possibilitando respostas muito mais rápidas para o tempo da consulta. Elas podem ser semelhantes às tabelas de resumo (veja abaixo), mas são mantidas automaticamente pelo DBMS.

Procedimentos armazenados

Os procedimentos armazenados são semelhantes às exibições, mas podem conter uma lógica muito mais complexa e, potencialmente, executar várias etapas de preparação de dados. Eles também podem ser parametrizados para retornar um conjunto de dados segmentado com base na entrada do usuário.

Os procedimentos armazenados têm suporte no Sybase ASE, SQL Server e Teradata. A fim de evitar que o procedimento armazenado seja executado várias vezes para uma única visualização, o Tableau executa o procedimento armazenado e grava o conjunto de resultados em uma tabela temporária no banco de dados. Então, as consultas reais feitas na visualização são executadas na tabela temporária. A execução do procedimento armazenado e o preenchimento da tabela temporária ocorrem quando a pasta de trabalho é aberta e sempre que os parâmetros do processo armazenado forem alterados. Esse processo pode demorar um pouco, e essas interações talvez sejam lentas.

Se você estiver extraindo os resultados de um procedimento armazenado parametrizado para uma extração de dados, a extração será depurada e atualizada sempre que você alterar os valores dos parâmetros.

Para obter mais informações sobre como usar os procedimentos armazenados, consulte a ajuda on-line do Tableau:

http://onlinehelp.tableau.com/current/pro/desktop/pt-br/help.htm#connect_basic_stored_procedures.html

Tabelas de resumo

Se você tiver um conjunto de dados muito grande e detalhado que geralmente resume para consultar (por exemplo, você armazena transações individuais, mas costuma usar os dados resumidos por dia, região, cliente, produto, etc.), crie uma tabela de resumo e use-a com o Tableau para agilizar suas consultas.

Nota: você pode usar as extrações de dados do Tableau para obter um resultado semelhante criando uma extração de dados agregada. Consulte a seção sobre extrações para obter mais detalhes.

SQL inicial

Outra alternativa ao SQL personalizado (se sua fonte de dados oferecer suporte) é usar a instrução do SQL personalizado em um bloco do SQL inicial. Você pode usá-la para criar uma tabela temporária que posteriormente será a tabela selecionada em sua consulta. Como o SQL inicial é executado apenas uma vez quando a pasta de trabalho é aberta (em vez de ser executado sempre que a visualização for alterada para o SQL personalizado) isso pode melhorar bastante o desempenho em alguns casos.

Observe que um administrador no Tableau Server pode definir uma restrição no SQL inicial para que ele não seja executado. Verifique se isso pode ser feito em seu ambiente, caso você esteja planejando publicar sua pasta de trabalho para compartilhá-la com outras pessoas.

Para obter mais informações sobre o SQL inicial, consulte a documentação on-line:

http://onlinehelp.tableau.com/current/pro/desktop/pt-br/help.htm#connect_basic_initialsql.html

O problema são os meus dados?

Um dos recursos mais importantes do Tableau é a sua capacidade de se conectar a dados de diversas plataformas. De forma geral, é possível caracterizar essas plataformas assim:

- Fontes de dados de arquivo (como arquivos .csv e do Excel)
- Fontes de dados de bancos de dados relacionais (como Oracle, Teradata e SQL Server, bem como dispositivos de análise especializados, como HP Vertica, IBM Netezza, etc.)
- Fontes de dados OLAP (como Microsoft Analysis Services e Oracle Essbase)
- Fontes de dados de “Big Data” (como o Hadoop)
- Fontes de dados na nuvem (como Salesforce, Google, etc.)

Cada tipo de fonte de dados tem suas vantagens e desvantagens, e é tratado de forma exclusiva.

Observe que o Tableau Desktop pode ser executado nos sistemas operacionais Windows e Mac OS X e que o conjunto de fontes de dados com suporte no Mac não é o mesmo do que aquele com suporte no Windows. A equipe do Tableau fará todo o possível para minimizar as diferenças entre as plataformas, mas atualmente algumas fontes de dados só têm suporte em uma plataforma.

Conselho geral

Use drivers nativos

O Tableau 10 oferece suporte a mais de 40 fontes de dados diferentes. Isso significa que o Tableau implementou técnicas, recursos e otimizações específicos para essas fontes de dados. Atividades de teste e engenharia para essas conexões garantem que elas sejam as mais robustas que o Tableau pode oferecer.

O Tableau também oferece suporte a ODBC para finalidades gerais de acesso a dados além da lista dos conectores nativos. Como um padrão definido publicamente, muitos fornecedores de bancos de dados disponibilizam drivers ODBC para seus produtos, e o Tableau também pode usar esses drivers ODBC para se conectar a dados. Pode haver diferenças na forma como o fornecedor de banco de dados interpreta ou implementa recursos do padrão ODBC. Em alguns casos, o Tableau recomendará ou solicitará que você crie uma extração de dados para continuar trabalhando com um determinado driver. Também haverá alguns drivers ODBC e bancos de dados aos quais o Tableau não poderá se conectar.

Se houver um driver nativo para a fonte de dados que você está consultando, use-o no lugar das conexões ODBC, porque isso geralmente resultará em um desempenho melhor. Observe também que as conexões ODBC estão disponíveis apenas no Windows.

Faça testes com a maior proximidade possível dos dados

Conforme dissemos antes, como um princípio geral, se a execução de consultas em uma fonte de dados for lenta, a experiência no Tableau será lenta. Uma boa maneira de testar o desempenho básico da fonte de dados é (se possível) instalar o Tableau Desktop na máquina onde a fonte de dados está armazenada e executar algumas consultas. Isso eliminará fatores de desempenho, como a latência e a largura de banda da rede, e permitirá que você entenda melhor o desempenho básico da consulta na fonte de dados. Além disso, usar o nome *localhost* para a fonte de dados, em vez do nome DNS, pode ajudar a determinar se fatores do ambiente, como resolução de nome lenta ou servidores proxy estão contribuindo para que o desempenho seja ruim.

Fontes de dados

Arquivos

Essa categoria abrange todos os formatos de arquivos de dados, sendo os arquivos de texto, como CSV, de planilhas do Excel e do MS Access os mais comuns. No entanto, ela também inclui arquivos de dados das plataformas estatísticas SPSS, SAS e R. Os usuários corporativos utilizam com frequência os dados nesse formato, porque essa é uma maneira comum de obter dados de conjuntos de dados “governados” – executando relatórios ou uma extração para consulta.

Em geral, é uma prática recomendada importar fontes de dados de arquivo para o Processador de dados rápido do Tableau. Isso agiliza as consultas e gera um arquivo muito menor para armazenar os valores dos dados. No entanto, se o arquivo for pequeno, ou for necessário estabelecer uma conexão em tempo real com o arquivo para refletir os dados alterados, você pode usar uma conexão em tempo real.

Extrações sombra

Quando você se conecta a um arquivo de texto/Excel ou de estatística não herdado, o Tableau cria de forma transparente um arquivo de extração como parte do processo de conexão. Esse arquivo é chamado de extração sombra e agiliza bastante o trabalho com os dados, em comparação a uma consulta feita diretamente no arquivo.

Você pode perceber que, ao usar um arquivo grande pela primeira vez, ele pode demorar vários segundos para ser carregado no painel de visualização. Isso acontece porque o Tableau está extraindo os dados do arquivo e gravando-os em um arquivo de extração sombra. Por padrão, esses arquivos são criados em

`C:\Users\\AppData\Local\Tableau\Caching\TemporaryExtracts` com um nome com hash que contém o caminho e a data da última modificação do arquivo de dados. Nesse diretório, o Tableau mantém as extrações sombra das cinco fontes de dados de arquivo mais recentes e exclui a mais antiga quando uma nova for criada.

Se você costuma reutilizar um arquivo que tem uma extração sombra, o Tableau simplesmente abre o arquivo da extração, e a visualização dos dados é exibida quase que imediatamente.

Embora os arquivos de extração sombra contenham dados subjacentes e outras informações, como as de uma extração padrão do Tableau, os arquivos de extração sombra são salvos em um formato diferente (a extensão .ttde). Isso significa que eles não podem ser usados como as extrações comuns do Tableau.

Conectores herdados para arquivos do Excel e de texto

Nas versões anteriores do Tableau, as conexões com arquivos do Excel e de texto utilizavam o driver do mecanismo de dados do Microsoft JET. Desde o Tableau 8.2, adotamos um driver nativo que oferece um desempenho melhor e é compatível com arquivos maiores e mais complexos. No entanto, há situações em que talvez você prefira usar os drivers herdados – se quiser usar um SQL personalizado, por exemplo. Nesses casos, os usuários têm a opção de redefinir a configuração para usar o driver JET herdado. O driver JET continuará sendo usado para ler arquivos do MS Access.

Uma lista detalhada das diferenças entre esses dois drivers pode ser encontrada aqui:

http://onlinehelp.tableau.com/current/pro/desktop/pt-br/help.htm#upgrading_connection.html

Observe que os drivers JET não estão disponíveis para Mac OS e, por isso, o Tableau Desktop para Mac não oferece suporte à leitura de arquivos do MS Access, nem a opção da conexão herdada para arquivos de texto e do Excel.

Bancos de dados relacionais

As fontes de dados relacionais são as formas mais comuns de fonte de dados para os usuários do Tableau, e o Tableau fornece drivers nativos para diversas plataformas. Elas podem ser baseadas em linhas ou colunas, pessoais ou empresariais e acessadas por meio de drivers nativos ou ODBC genérico. Essa categoria tecnicamente também inclui as fontes de dados do MapReduce, porque elas são acessadas pelas camadas de acesso do SQL, como Hive ou Impala, mas discutiremos isso com mais detalhes na seção “Big Data” abaixo.

Existem muitos fatores internos que afetam a velocidade de uma consulta em sistemas de bancos de dados relacionais (RDBMS). Para alterá-los ou ajustá-los, geralmente você precisa do DBA, mas isso pode gerar melhorias significativas no desempenho.

Linhas x colunas

Os sistemas de RDBMS são oferecidos em dois modelos: baseado em linhas e baseado em colunas. Os layouts de armazenamento baseados em linhas são ideais para cargas de trabalho do tipo OLTP, que são mais pesadas com transações interativas. Já os layouts de armazenamento baseados em colunas são ideais para cargas de trabalho de análise, como data warehouses, que geralmente envolvem consultas muito complexas em conjuntos de dados enormes.

Atualmente, diversas soluções de análise de alto desempenho usam o modelo de RDBMS baseado em coluna, e suas consultas serão executadas com mais rapidez se você usar uma solução desse tipo. Exemplos de bancos de dados baseados em colunas aos quais o Tableau oferece suporte são: Actian Vector, Amazon Redshift, HP Vertica, IBM Netezza, MonetDB, Pivotal Greenplum, SAP HANA e SAP Sybase IQ.

SQL como interface

Existem vários sistemas que não utilizam tecnologias tradicionais de RDBMS, mas que são considerados fontes relacionais, porque oferecem uma interface baseada no SQL. Para o Tableau, esses sistemas abrangem diversas plataformas NoSQL (como MarkLogic, DataStax, MongoDB, etc.) e plataformas de aceleração de consulta, como Kognitio, AtScale e JethroData.

Índices

A indexação correta do banco de dados é essencial para o bom desempenho de uma consulta:

- Verifique se há índices em todas as colunas que fazem parte das uniões de tabela.
- Verifique se há índices nas colunas utilizadas em filtros.
- Saiba que usar filtros de dados discretos em alguns bancos de dados pode fazer com as consultas não usem os índices nas colunas de data e data e hora. Falaremos sobre isso posteriormente na seção sobre filtros, mas usar um filtro de intervalo de datas garante que o índice de data seja usado. Por exemplo, em vez de usar `YEAR([DateDim])=2010` expresse o filtro como `[DateDim] >= #2010-01-01# and [DateDim] <= #2010-12-31#` .
- Garanta que o recurso de estatística esteja habilitado em seus dados para permitir que o otimizador de consulta crie planos de consulta de alta qualidade.

Muitos ambientes de DBMS têm ferramentas de gerenciamento que analisarão uma consulta e recomendarão índices que podem ajudar.

NULOS

Ter valores NULOS nas colunas de dimensão pode reduzir a eficácia das consultas. Vamos supor que temos uma visualização na qual desejamos mostrar o total de vendas por país para os 10 principais produtos. Primeiro, o Tableau gera uma subconsulta para encontrar os 10 principais produtos e, em seguida, adiciona essa informação à consulta por país para obter o resultado final.

Se a coluna Produto estiver definida como PERMITIR NULO, será necessário executar uma consulta para verificar se NULO é um dos 10 principais produtos retornados pela subconsulta. Se esse for o caso, precisaremos fazer uma união para preservar os valores nulos, que é uma operação muito mais desgastante do que uma união regular. Se a coluna Produto estiver definida como NÃO NULO, sabemos que o resultado da subconsulta não contém valores nulos e que podemos fazer uma união normal, eliminando a consulta “verificar valores NULO”.

Por isso, sempre que possível, defina as colunas de suas dimensões como NÃO NULO.

Integridade referencial

Quando você une várias tabelas em uma fonte de dados, o Tableau tem um ótimo recurso (e geralmente invisível ao usuário) chamado “separação de união”. Como as uniões demandam tempo e recursos para serem processadas no servidor do banco de dados, não queremos enumerar cada união que declaramos em nossa fonte de dados o tempo todo. A separação de união permite consultar apenas as tabelas relevantes, em vez de todas as tabelas definidas na união.

Considere o seguinte cenário, onde unimos várias tabelas em um esquema estrela pequeno:

The screenshot shows the Tableau interface for a star schema named 'OrdersFact+ (NewWave)'. The schema diagram shows 'OrdersFact' as the fact table, connected to five dimension tables: 'CustomerDim', 'DeliveryDim', 'LocationDim', 'ProductDim', and 'TimeDim'. Below the diagram, a table displays data from the 'OrdersFact' table, sorted by 'Data Source Order'. The table has 7 columns: Customer ID, Name, Delivery ID, Shipping Mode, Days, Place ID, and Region. The data is as follows:

Customer ID (Cust...)	Name	Delivery ID (Delive...)	Shipping Mode	Days	Place ID (Location...)	Region
75.00	Altd	8900.00	First Class	3.00000	1051.00	Europe
1195.00	Systed Systems, Inc	8300.00	First Class	2.00000	661.00	Asia
1004.00	Seas	2400.00	Standard Class	5.00000	885.00	Asia
1439.00	Vellutems Corporation	0.00	Standard Class	4.00000	595.00	Asia
883.00	Mics Consulting	0.00	Standard Class	4.00000	786.00	Asia
984.00	Quan	2400.00	Standard Class	5.00000	173.00	Africa
1029.00	Sems Group	4200.00	Standard Class	6.00000	31.00	Africa
953.00	Nettpaq USA	2400.00	Standard Class	5.00000	130.00	Africa
683.00	Inetems Research	6000.00	Second Class	2.00000	282.00	America
1302.00	TER	5400.00	Standard Class	7.00000	541.00	Asia
1054.00	Silian Corporation	0.00	Standard Class	4.00000	151.00	Africa

Com a separação de união, clicar duas vezes na medida Vendas gera a seguinte consulta:

```

SELECT SUM([OrdersFact].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[OrdersFact] [OrdersFact]
GROUP BY ()

```

Sem isso, uma consulta muito menos eficiente é gerada:

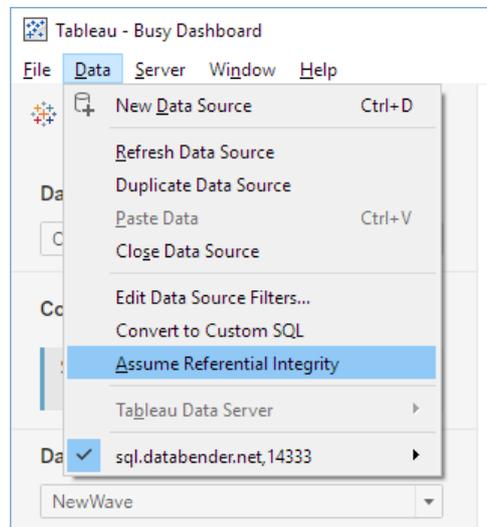
```

SELECT SUM([OrdersFact].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[OrdersFact] [OrdersFact]
INNER JOIN [dbo].[CustomerDim] [CustomerDim]
ON ([OrdersFact].[Customer ID] = [CustomerDim].[Customer ID])
INNER JOIN [dbo].[DeliveryDim] [DeliveryDim]
ON ([OrdersFact].[Delivery ID] = [DeliveryDim].[Delivery ID])
INNER JOIN [dbo].[LocationDim] [LocationDim]
ON ([OrdersFact].[Place ID] = [LocationDim].[Place ID])
INNER JOIN [dbo].[ProductDim] [ProductDim]
ON ([OrdersFact].[Product ID] = [ProductDim].[Product ID])
INNER JOIN [dbo].[TimeDim] [TimeDim]
ON ([OrdersFact].[Date ID] = [TimeDim].[Date ID])
GROUP BY ()

```

Todas as tabelas de dimensões devem ser unidas para garantir que as somas de medidas corretas sejam calculadas desde o início. Por exemplo, se a nossa tabela de fatos contivesse dados de 2008 a 2012, mas a tabela de dimensões de tempo tivesse valores apenas de 2010 a 2012, o resultado de SUM([Sales]) possivelmente mudaria se a tabela com as dimensões de tempo fosse incluída.

Antigamente, era necessário ter uma integridade referencial “pesada” onde a regra fosse aplicada pelo DBMS. No entanto, muitos clientes têm fontes de dados nas quais a integridade referencial é usada na camada do aplicativo ou por meio de um processo de ETL (chamada de integridade referencial “suave”). Os usuários podem informar o Tableau de que uma integridade referencial suave foi aplicada e que a separação de união pode ser usada com segurança por meio da opção “Presumir integridade referencial”:



Embora o Tableau possa usar a integridade referencial pesada ou suave, geralmente é muito melhor usar a integridade referencial pesada, porque o banco de dados também pode fazer a separação de união. Para obter mais informações, consulte a seguinte série de artigos escritos por Russell Christopher em seu blog Tableau Love:

- <http://bit.ly/1HACmPV>
- <http://bit.ly/1HACqPn>

Particionamento

Particionar um banco de dados ajuda a melhorar o desempenho com a divisão de uma tabela grande em tabelas individuais menores (chamadas de partições ou fragmentos). Isso significa que as consultas podem ser executadas com mais rapidez, porque há menos dados para verificar e/ou há mais unidades para oferecer E/S. O particionamento é uma estratégia recomendada para volumes de dados muito grandes e é transparente para o Tableau.

O particionamento funcionará bem com o Tableau se ele for feito com uma dimensão (por exemplo, um período, uma região, uma categoria, etc.) que possa ser filtrada, para que cada consulta precise ler apenas os registros em uma única partição.

É importante saber disso em relação a alguns bancos de dados, filtros de “intervalos de datas” (não filtros discretos), para garantir que os índices da partição sejam usados corretamente. Caso contrário, uma verificação na tabela inteira pode resultar em um desempenho muito ruim.

Tabelas temporárias

Muitas operações no Tableau podem resultar no uso de tabelas temporárias. Por exemplo, a criação de conjuntos e grupos ad hoc e a execução de combinação de dados. É recomendável conceder uma permissão para os usuários criarem e excluam tabelas temporárias, bem como garantir que o ambiente tenha espaço suficiente no spool para as consultas que estão sendo executadas.

OLAP

O Tableau oferece suporte a várias fontes de dados OLAP:

- Microsoft Analysis Services
- Microsoft PowerPivot (PowerPivot para Excel e PowerPivot para SharePoint)
- Oracle Essbase
- SAP Business Warehouse
- Teradata OLAP

As conexões com OLAP e bancos de dados relacionais têm um funcionamento distinto, devido às diferenças de linguagem subjacentes entre MDX/DAX e SQL. É importante lembrar que ambos terão a mesma interface de usuário no Tableau, as mesmas visualizações e a mesma linguagem de expressão para as medidas calculadas. As diferenças estão, em sua maioria, relacionadas a metadados (como e onde são definidos), filtros, como os totais e as agregações funcionam e como a fonte de dados pode ser usada na combinação de dados.

Para obter mais detalhes sobre as diferenças de usar OLAP e fontes de dados relacionais no Tableau, consulte este artigo da base de dados de conhecimento:

<http://kb.tableau.com/articles/knowledgebase/functional-differences-olap-relational?lang=pt-br>

Extrações de dados do SAP BW

O SAP BW é único entre nossas fontes de dados OLAP porque você pode extrair dados de cubos do SAP BW para o Processador de dados do Tableau (observação: é necessário ter um código de chave especial que pode ser obtido da Tableau). O Tableau recupera os nós no nível da folha (sem detalhamento no nível dos dados) e os transforma em uma fonte de dados relacionais. Como a transformação de multidimensional para relacional não preserva todas as estruturas de cubo, com as extrações de cubo, não é possível alternar livremente entre extrações e conexões em tempo real sem impactar o estado das visualizações. Você terá que escolher uma das duas antes de criar a visualização, mas não precisa

decidir tudo previamente. Você pode alternar entre as opções de alias (principal, nome longo, etc.) após a extração.

Encontre mais informações sobre como criar extrações do SAP BW aqui:

<http://tabsoft.co/1SuYf9d>

Big Data

O Big Data é um termo muito abrangente no mundo da análise de dados. Contudo, neste documento, ele se refere especificamente às plataformas baseadas no Hadoop. O Tableau 10 oferece suporte a quatro distribuições do Hadoop compatíveis com conexões do Hive e/ou do Impala:

- Amazon EMR
 - HiveServer
 - HiveServer2
 - Impala

- Cloudera Hadoop
 - HiveServer
 - HiveServer2
 - Impala

- Hortonworks Hadoop Hive
 - HiveServer
 - HiveServer2
 - Hortonworks Hadoop Hive

- MapR Hadoop Hive
 - HiveServer
 - HiveServer2

- Spark SQL
 - SharkServer *
 - SharkServer2 *
 - SparkThriftServer

** Observe que as conexões do SharkServer e do SharkServer2 são fornecidas para o seu uso, mas o Tableau não oferece suporte a elas.*

O Hive atua como uma camada de tradução SQL-Hadoop, traduzindo a consulta para o MapReduce, que, em seguida, é executada nos dados do HDFS. O Impala executa a instrução SQL diretamente nos dados do HDFS (sem precisar do MapReduce). O Tableau também oferece suporte ao Spark SQL, um mecanismo de processamento de software livre para Big Data, que tem um desempenho até 100 vezes mais rápido do que o do MapReduce, porque sua execução é feita na memória, não no disco.

Normalmente, o Impala é mais rápido do que o Hive, mas o Spark vem demonstrando uma agilidade ainda maior.

Mesmo com esses componentes adicionais, muitas vezes o Hadoop não é suficientemente responsivo para consultas de análise, como as criadas pelo Tableau. As extrações de dados do Tableau frequentemente são usadas para melhorar os tempos de resposta de consulta. Forneceremos mais informações sobre extrações e falaremos sobre como elas podem ser usadas com “Big Data” posteriormente.

Você pode encontrar mais detalhes sobre como melhorar o desempenho com as fontes de dados do Hadoop aqui:

<http://kb.tableau.com/articles/knowledgebase/hadoop-hive-performance?lang=pt-br>

Nuvem

O Tableau atualmente oferece suporte às seguintes fontes de dados na nuvem:

- Salesforce.com
- Google Analytics
- oData (incluindo o Windows Azure Marketplace DataMarket)

Esse primeiro grupo de fontes leu um conjunto de registros de dados de um serviço da Web e os carregou em um arquivo de extração de dados do Tableau. A conexão em tempo real não é uma opção para essas fontes de dados, apesar de o arquivo da extração poder atualizar os dados que contém. Com o Tableau Server, esse processo de atualização pode ser automatizado e agendado.

O Tableau também oferece suporte às seguintes plataformas de dados na nuvem:

- Amazon Aurora
- Amazon Redshift
- Amazon RDS
- Google BigQuery
- Google Cloud SQL
- Planilhas Google
- SQL Data Warehouse do Microsoft Azure
- Snowflake

Em comparação ao grupo de fontes de dados anterior, os membros deste grupo funcionam como bancos de dados relacionais e permitem conexões em tempo real e extrações. Não forneceremos mais informações sobre eles nesta seção (consulte a seção sobre fontes de dados relacionais acima), mas diremos apenas que você deve usar conexões em tempo real com eles para evitar a transferência de grandes volumes de dados para a nuvem.

Por fim, o Tableau também oferece um conector de dados genérico para importar dados de serviços baseados na Web que publicam dados nos formatos JSON, XML ou HTML – o Conector de dados da Web.

Salesforce

Quando você se conectar ao Salesforce, considere as seguintes limitações do conector:

Não é possível usar uma “conexão em tempo real”

São vários os motivos que nos fizeram optar somente pelo uso de extrações, no lugar de uma conexão em tempo real, incluindo:

- Desempenho – as consultas de análise em tempo real no Salesforce (geralmente) são lentas.
- Cotas de API - acessar o Salesforce em tempo real com muita frequência pode resultar na suspensão da conta, caso a cota diária seja atingida. Com a extração, conseguimos utilizar as APIs de forma eficiente e minimizar o número de chamadas de API necessárias, evitando que o limite de acesso seja atingido. Para manter um ótimo desempenho e garantir que a API Force.com esteja disponível para todos os clientes do site, o Salesforce.com equilibra as cargas das transações impondo dois tipos de limites: Limites de solicitações simultâneas

de API (<http://sforce.co/1f19cQa>) e Limites do total de solicitações de API (<http://sforce.co/1f19kiH>).

A primeira extração pode ser muito lenta

A primeira extração de dados do Salesforce pode demorar bastante, dependendo do tamanho da tabela, da carga no force.com, etc. Isso pode acontecer porque os objetos são baixados integralmente.

As opções de união são limitadas

Quando escolher a opção de várias tabelas, lembre-se de que você só pode unir objetos por suas chaves PK/FK (somente união esquerda e interna).

Não é possível pré-filtrar os dados

Não há nenhum recurso para pré-filtrar os dados usando o conector do Salesforce. Se isso for muito importante para você, use um driver ODBC de terceiros para o Salesforce (como o do Simba ou do DataDirect) que ofereça suporte a conexões em tempo real. Então, você poderá criar uma extração a partir dessa conexão.

O DBAmp também oferece uma solução que permite você despeje os dados do Salesforce em um banco de dados SQL Server. Depois de fazer isso, você poderá conectá-lo ao Tableau usando o conector do SQL Server.

Colunas com fórmula não podem ser migradas

Se houver campos calculados em seus dados, você precisará recriá-los no Tableau depois de extrair os dados.

Há um limite de 10 mil caracteres para as consultas

A API do Force.com restringe as consultas a um total de 10.000 caracteres. Se estiver se conectando a uma ou mais tabelas muito grandes (com várias colunas e possivelmente nomes de coluna longos), você pode atingir esse limite ao tentar criar uma extração. Nesses casos, selecione menos colunas para reduzir o tamanho da consulta. Em alguns casos, o Salesforce.com consegue aumentar esse limite de consulta para sua empresa. Fale com seu administrador do Salesforce para obter mais informações.

Google Analytics

O Google Analytics (GA) gera uma amostra dos seus dados quando um relatório inclui muitas dimensões ou um grande volume de dados. Se os dados de uma propriedade da Web em um determinado intervalo de datas excederem (em uma conta regular do GA) 50.000 visitas, o GA agregará os resultados e retornará um conjunto de amostra desses dados. Quando o GA retornar um conjunto de amostras desses dados para o Tableau, o Tableau exibirá a seguinte mensagem no canto inferior direito de uma exibição:

“Dados de amostra do Google Analytics retornados. A amostragem ocorre quando a conexão inclui um grande número de dimensões ou uma grande quantidade de dados. Consulte a documentação do Google Analytics para saber mais sobre como a amostragem afeta seus resultados de relatório.”

É importante saber quando uma amostra dos seus dados está sendo gerada, porque agregar determinados conjuntos de amostras de dados pode ocasionar inferências muito distorcidas e imprecisas. Por exemplo, suponha que você agregou um conjunto de amostras de dados que descreve uma categoria incomum dos dados. As inferências no conjunto de amostras podem ser distorcidas devido ao número insuficiente de amostras dessa categoria. Para criar exibições do GA que permitem realizar inferências precisas sobre seus dados, é necessário ter uma amostra grande o bastante na categoria em que você deseja fazer as inferências. O mínimo de amostras recomendado é 30.

Para obter informações sobre como ajustar as amostras do GA e sobre como gerar amostras do GA, consulte a documentação do GA:

<http://bit.ly/1BkFoTG>

Se não quiser usar amostras, você tem duas opções:

- Executar vários relatórios do GA no nível da sessão ou do acesso para dividir os dados em blocos sem amostras. Em seguida, baixe os dados para um arquivo do Excel e use o mecanismo de extração do Tableau para “adicionar dados da fonte de dados...” e reorganizar os dados em um único conjunto de dados.
- Atualizar o GA para uma conta Premium, aumentando, assim, o número de registros que podem ser incluídos em um relatório. Isso facilita bastante a divisão dos dados em blocos para análise. Pensando no futuro, o Google anunciou que permitirá aos clientes GA Premium exportar suas sessões e acessar o nível dos dados do Google BigQuery para oferecer mais análises. Essa seria uma abordagem muito mais simples, porque o Tableau pode se conectar diretamente ao BigQuery.

Por fim, observe que a API usada pelo Tableau para consultar o GA limita a consulta a um máximo de sete dimensões e 10 medidas.

Conector de dados da Web

Um conector dados da Web do Tableau permite que você se conecte aos dados que ainda não têm um conector. Com um conector de dados da Web, é possível criar e usar uma conexão com praticamente todos os dados acessíveis via HTTP. Isso inclui: serviços Web internos, dados JSON, dados XML, APIs REST e muitos outros recursos. Como você controla o método de coleta dos dados, é possível até mesmo combinar dados de várias fontes.

Você cria um conector de dados da Web escrevendo uma página da Web que contenha JavaScript e HTML. Depois de escrever um conector de dados da Web, você poderá compartilhá-lo com outros usuários do Tableau publicando-o no Tableau Server.

Para ajudar você a criar conectores de dados da Web, elaboramos um kit de desenvolvimento de software (SDK) que inclui modelos, um código de exemplo e um simulador para você testar seus conectores de dados da Web. Essa documentação também inclui um tutorial que ensina a criar um conector de dados da Web do zero. O SDK de conector de dados da Web é um projeto de software livre. Para obter mais informações, consulte a página do Conector de dados da Web do Tableau no GitHub.

Encontre mais informações sobre conectores de dados da Web aqui:

<http://tabsoft.co/1NnGsBU>

Preparação de dados

Os dados que precisamos usar em nossas pastas de trabalho nem sempre são perfeitos. Eles podem estar desorganizados, não ter a forma correta e ainda estar espalhados em diversos arquivos ou bancos de dados. Historicamente, o Tableau funciona melhor quando é conectado a dados organizados e normalizados, o que indica que, em alguns casos, você precisou prepará-los usando outras ferramentas antes de carregá-los no Tableau.

Isso não é mais a regra, porque o Tableau agora oferece vários recursos para ajudar você a carregar dados desorganizados. Esses recursos incluem:

- Dinamizar

- União
- Interpretador de dados
- Mesclar colunas

Incluir a preparação de dados no processo de análise a) ajuda você a permanecer no fluxo de análise e b) permite que os usuários que não têm acesso a outras ferramentas continuem analisando seus dados no Tableau. No entanto, essas operações geralmente têm uma computação intensa (principalmente com grandes volumes de dados) e podem afetar o desempenho dos usuários do relatório. Sempre que possível, é recomendável usar uma extração de dados para materializar os dados depois dessas operações.

Você também pode considerar o uso desses recursos como um complemento para os processos de ETL existentes e avaliar se o pré-processamento de upstream de dados do Tableau será uma abordagem melhor e mais eficiente se os dados forem usados em vários relatórios ou compartilhados com outras ferramentas de BI.

Extrações de dados

Já discutimos as técnicas para melhorar o desempenho de conexões de dados nas quais os dados permanecem em seus formatos originais. Essas conexões são chamadas de conexões em tempo real e, nesses casos, o desempenho e a funcionalidade dependem da plataforma da fonte de dados. Para melhorar o desempenho com conexões em tempo real, geralmente é necessário fazer alterações na fonte de dados, e, para muitos clientes, isso não é possível.

Uma alternativa para todos os usuários é usar o mecanismo de dados rápido do Tableau e extrair os dados do sistema de dados de origem para uma extração de dados do Tableau. Para a maioria dos usuários, essa é a maneira mais rápida e fácil de melhorar significativamente o desempenho de uma pasta de trabalho com qualquer fonte de dados.

Uma extração:

- é um cache de dados permanente gravado no disco e que pode ser reproduzido;
- é um armazenamento de dados em colunas (um formato no qual os dados foram otimizados para consultas de análise);
- está completamente desconectada do banco de dados durante a consulta. Ela é uma substituição da conexão de dados em tempo real;
- pode ser atualizada, por meio da regeneração completa da extração ou da adição incremental de linhas de dados a uma extração existente;
- adapta-se à arquitetura (diferentemente da maioria das tecnologias executadas na memória, ela não é restringida pela quantidade de RAM física disponível);
- é portátil, pois as extrações são armazenadas como arquivos para que possam ser copiadas em uma unidade de disco rígido local e usadas quando o usuário não estiver conectado à rede da empresa. Elas também podem ser usadas para incorporar dados em pastas de trabalho em pacote, distribuídas para uso com o Tableau Reader;
- geralmente é mais rápida do que a conexão em tempo real com os dados subjacentes.

Tom Brown, do The Information Lab, escreveu um excelente artigo que explica vários casos de uso em que as extrações oferecem benefícios (leia também os comentários dos exemplos adicionais de outros usuários):

<http://bit.ly/1F2iDnT>

Há uma observação importante sobre as extrações de dados: elas não substituem um data warehouse, mas o complementam. Embora seja possível usá-las para coletar e agregar dados ao longo do tempo (por exemplo, adicionar dados incrementalmente de acordo com um ciclo periódico), elas devem ser usadas como uma solução tática, e não como uma solução de uso prolongado. As atualizações incrementais não oferecem suporte à atualização ou exclusão de registros que já foram processados. Alterar esses registros exige uma recarga completa da extração.

Por fim, as extrações não podem ser criadas em fontes de dados OLAP, como o SQL Server Analysis Services ou o Oracle Essbase. A exceção a essa regra é que você pode criar extrações do SAP BW (consulte a seção correspondente acima).

Quando usar extrações? Quando usar conexões em tempo real?

Como tudo, há hora e lugar para as extrações de dados. Veja abaixo alguns cenários para os quais as extrações podem ser vantajosas:

- Execução de consulta lenta – se o sistema de dados de origem estiver processando com lentidão as consultas geradas pelo Tableau Desktop, criar uma extração pode ser uma maneira simples de melhorar o desempenho. O formato dos dados da extração foi desenvolvido para oferecer uma resposta rápida a consultas de análise. Portanto, nesse caso, você pode pensar na extração como um cache de aceleração de consulta. Para alguns tipos de conexão, essa é uma prática recomendada (por exemplo, para arquivos de texto grandes e conexões de SQL personalizado), e algumas fontes funcionarão apenas nesse modelo (consulte a seção sobre fontes de dados na nuvem).
- Análise off-line – se você precisa trabalhar com dados e a fonte de dados original não estiver disponível (por exemplo, se você não estiver conectado à rede durante uma viagem ou trabalhando em casa). As extrações de dados persistem como arquivos que podem ser copiados facilmente para um dispositivo portátil, como um laptop. Então, basta alternar entre uma conexão em tempo real e uma extração, quando você se conectar e se desconectar da rede.
- Pastas de trabalho em pacote para Tableau Reader/Online/Public – se você planeja compartilhar suas pastas de trabalho com outros usuários que irão abri-las no Tableau Reader ou se pretende publicá-las no Tableau Online ou no Tableau Public, será necessário incorporar os dados em um arquivo de pasta de trabalho em pacote. Mesmo se a pasta de trabalho usar fontes de dados que também podem ser incorporadas, como fontes de dados de arquivo, as extrações de dados sempre fornecem um nível alto de compactação de dados para que a pasta de trabalho em pacote resultante seja significativamente menor.
- Funcionalidade adicional – para algumas fontes de dados, como as fontes de dados de arquivo utilizadas com o driver JET herdado), algumas funções no Tableau Desktop não têm suporte (por exemplo, agregações de percentil/mediana/distinção de contagem/classificação, operações de entrada/saída definidas, etc.). Extrair os dados é uma forma simples de habilitar essas funções.
- Segurança dos dados – se você deseja compartilhar um subconjunto dos dados do sistema de origem, crie uma extração e disponibilize-a para outros usuários. Você pode limitar os campos/colunas que podem ser incluídos e compartilhar dados agregados onde deseja que os usuários vejam valores de resumo, e não os dados individuais no registro.

As extrações são muito úteis, mas não são uma solução mágica para todos os problemas.

Há situações em que o uso de extrações não é apropriado:

- Dados em tempo real – como as extrações refletem os dados de um ponto no tempo, não é apropriado usá-las se você precisa de dados em tempo real em suas análises. É possível

atualizar extrações automaticamente usando o Tableau Server, e muitos clientes fazem isso várias vezes ao dia, mas para ter acesso a dados em tempo real, você precisa de uma conexão em tempo real.

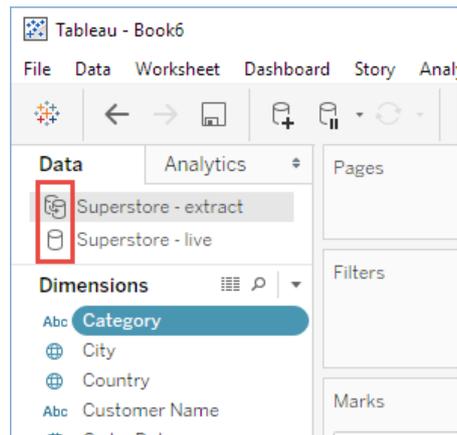
- Dados em massa – se o volume de dados com que você precisa trabalhar é muito grande (a definição de “em massa” pode variar de usuário para usuário, mas geralmente se refere a bilhões ou trilhões de registros), extrair esses dados pode não ser prático. O arquivo de extração resultante pode ser grande demais ou o processo de extração pode demorar muitas horas para ser concluído. Contudo, há exceções. Se você tiver um conjunto de dados muito grande, mas trabalhará em uma amostra filtrada ou em um subconjunto agregado desses dados, usar extrações pode ser uma ótima ideia. De forma geral, o mecanismo de extração do Tableau foi criado para trabalhar bem com centenas de milhões de registros, mas seu desempenho pode ser afetado pela forma e pela cardinalidade dos dados.
- Funções de SQL não processado de passagem – se sua pasta de trabalho usar funções de passagem, elas não funcionarão com uma extração de dados.
- Segurança robusta no nível do usuário – se você precisa de segurança robusta no nível do usuário, ela deve ser implementada na fonte de dados. Caso haja filtros no nível do usuário aplicados na pasta de trabalho, para removê-los, basta um usuário conceder acesso a todos os dados da extração. Essa orientação não deve ser seguida se a extração tiver sido publicada no Tableau Server com os filtros da fonte de dados definidos, e outros usuários estiverem acessando essa extração por meio do Servidor de dados. Observação: certifique-se de que as permissões de download tenham sido revogadas para os usuários, garantindo que eles não possam ignorar os filtros aplicados.

Criando extrações no Tableau Desktop

Na maioria dos casos, a primeira criação de uma extração é feita no Tableau Desktop e é muito simples. Depois de se conectar aos dados, acesse o menu Dados e clique em “Extrair dados”. Em seguida, aceite as configurações padrão na caixa de diálogo (falaremos mais sobre isso depois). O Tableau perguntará onde você deseja salvar a extração. Escolha um local para salvar o arquivo, embora o Tableau provavelmente direcione você para o diretório “Meu repositório do Tableau | Fontes de dados”, que também é um bom lugar, e aguarde a extração ser criada.

O tempo de espera pode variar de acordo com a tecnologia de banco de dados que estiver sendo usada, da velocidade da rede, dos volumes de dados, etc. Ele também depende da velocidade e da capacidade da estação de trabalho, visto que criar uma extração é uma atividade que consome bastante memória e exige muito do processador.

Você saberá que a operação foi concluída quando o ícone da fonte de dados mudar (exibindo outro ícone de banco de dados atrás dele, que representa uma cópia, exatamente o que uma extração é).



Quando você cria uma extração dessa forma (usando o Tableau Desktop), o processamento é feito em sua estação de trabalho, por isso, é necessário garantir que ela tenha recursos suficientes para concluir essa tarefa. A criação de uma extração usa todos os tipos de recursos (CPU, RAM, armazenamento em disco, entrada/saída de rede), e processar grandes volumes de dados em um computador com recursos limitados pode resultar em erros, caso um desses recursos seja insuficiente. É recomendável que extrações grandes sejam criadas em uma estação de trabalho adequada, com uma CPU rápida, vários núcleos, muita memória RAM, E/S rápida, etc.

O processo de criação de extração exige um espaço temporário no disco para gravar os arquivos de trabalho (até duas vezes o tamanho do arquivo de extração final). Esse espaço é alocado no diretório especificado pelo ambiente TEMP variável (normalmente `C:\WINDOWS\TEMP` ou `C:\Users\USERNAME\AppData\Local\Temp`). Se essa unidade não tiver espaço suficiente, aponte a variável de ambiente para um local maior.

Se for impossível (ou impraticável) realizar um processo de extração inicial em uma estação de trabalho, você pode adotar a solução a seguir para criar uma extração vazia que será publicada no Tableau Server. Crie um campo calculado que contenha `DateTrunc("minute", now())`. Em seguida, adicione-o aos filtros da extração e exclua o único valor que ele mostra. Seja rápido, porque você tem um minuto antes que esse filtro não seja mais válido. Se precisar de mais tempo, aumente o intervalo de publicação (altere-o para 5 minutos, 10 minutos ou até uma hora, se necessário). Isso criará uma extração vazia em seu computador. Depois que você publicar no Tableau Server e acionar o agendamento de atualizações, ele preencherá toda a extração porque o carimbo de data e hora que excluimos não é mais o mesmo.

Criando extrações com a API de extração de dados

O Tableau também oferece uma API (interface de programação de aplicativos) que permite aos desenvolvedores criar diretamente um arquivo de extração de dados do Tableau (TDE). Os desenvolvedores podem usar essa API para gerar extrações de softwares locais e de softwares como um serviço. Com essa API, é possível inserir dados sistematicamente no Tableau quando não houver um conector nativo para a fonte de dados que você está usando.

Essa API está disponível para desenvolvedores no Python e no C/C++/Java nos sistemas operacionais Windows e Linux. Encontre mais informações sobre a API aqui:

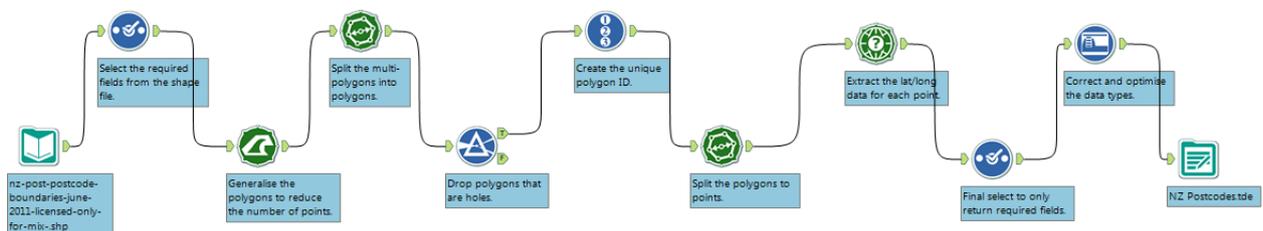
http://onlinehelp.tableau.com/current/pro/desktop/pt-br/extracting_TDE_API.html

Criando extrações com ferramentas de terceiros

Muitos desenvolvedores de ferramentas de terceiros usaram a API de extração de dados para adicionar um TDE nativo resultante aos seus aplicativos. Esses aplicativos incluem plataformas

de análise, como o Adobe Marketing Cloud, bem como ferramentas de ETL, como o Alteryx e o Informatica.

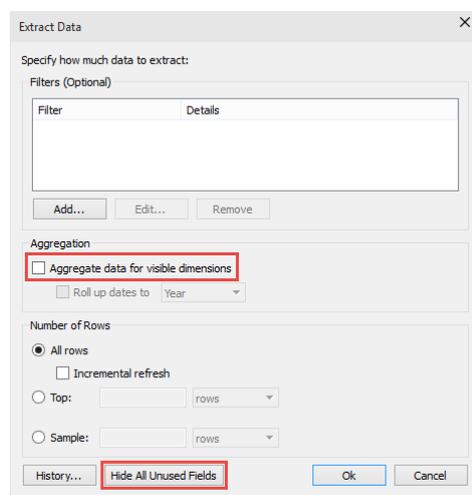
Se os seus requisitos de preparação de dados forem complexos, você pode usar ferramentas como o Alteryx e o Informatica para processar com eficiência os estágios de ETL e, em seguida, criar diretamente um arquivo TDE com os dados preparados, para ser usado no Tableau Desktop.



Extrações agregadas

O uso de uma extração agregada sempre melhora o desempenho. Mesmo se você estiver usando o Teradata ou o Vertica com grandes quantidades de dados, extrair os dados pode ser vantajoso, contanto que eles sejam agregados e filtrados apropriadamente. Por exemplo, você pode filtrar os dados se quiser analisar apenas os dados mais recentes.

Você pode definir a extração antecipadamente, escolhendo os campos desejados e marcando a caixa de seleção “Agregado dados para dimensões visíveis” na caixa de diálogo Extrair dados do Tableau Desktop. Depois de fazer sua análise e criar seu painel, quando estiver pronto para publicar, você também pode voltar à caixa de diálogo Dados da extração e clicar no botão Ocultar todos os campos não utilizados. Assim, ao extrair os dados, você terá o mínimo necessário para criar a exibição.



Criar extrações agregadas é uma técnica muito poderosa quando você tem uma grande quantidade de dados base, mas precisa criar exibições resumidas que executam consultas no conjunto de dados inteiro. Por exemplo, você pode ter bilhões de registros de dados de transações detalhadas que representam 10 anos de vendas e quer começar mostrando a tendência geral das vendas nesse período. A consulta nesta exibição inicial possivelmente seria lenta, porque ela precisaria percorrer bilhões de linhas. Criando uma extração agregada no nível do ano, conseguimos reduzir o esforço necessário à consulta no momento da exibição, porque teremos apenas 10 números na extração. É claro que esse é um exemplo bastante simples, visto que provavelmente haveria mais dimensões além do tempo, mas a capacidade de reduzir significativamente o número de registros que precisam ser consultados é muito útil.

Podemos criar pastas de trabalho extremamente sofisticadas com diversos níveis de detalhe criando várias extrações agregadas, cada uma delas ajustada para oferecer suporte a um nível de detalhe específico, ou combinando extrações agregadas com conexões em tempo real. Por exemplo, você pode ter um conjunto de exibições de resumo inicial que utiliza uma extração extremamente agregada, mas ao detalhar os dados, você usa os filtros de ação das exibições de resumo em outra planilha com uma conexão em tempo real. Isso significa que as exibições de resumo serão mais rápidas porque não precisarão percorrer todo o conjunto de dados base, além de não precisarmos extrair todos os dados base para oferecer suporte às ações de detalhamento. A conexão em tempo real também será rápida, porque estamos acessando um pequeno conjunto de recursos no nível de detalhamento.

Dessa forma, você pode combinar, corresponder e agregar em diferentes níveis para resolver praticamente qualquer problema de desempenho e obter os resultados com a rapidez necessária. Como o Tableau é eficiente com a memória, melhorar o desempenho dessa forma é relativamente fácil, e você pode ter várias extrações em execução ao mesmo tempo.

Otimizando as extrações

O Tableau Server otimiza as colunas físicas do banco de dados e as colunas adicionais criadas no Tableau. Essas colunas incluem os resultados dos cálculos deterministas, como manipulações e concatenações de strings, onde o resultado nunca será alterado, bem como os grupos e os conjuntos. Os resultados de cálculos não deterministas, como aqueles que envolvem um parâmetro ou agregações (por exemplo, soma ou média), calculados em tempo de execução, não podem ser armazenados.

O usuário deve atualizar a extração após adicionar apenas duas linhas de dados, e observe que o tamanho da extração saltou de 100 MB para 120 MB. Esse salto em tamanho deve-se à otimização, obtida com a criação de colunas adicionais com valores de campo calculado, porque é mais barato armazenar os dados em disco do que recalculá-los sempre que forem necessários.

É importante prestar atenção no seguinte: se você estiver fazendo cópias duplicadas de uma conexão para uma extração de dados, garanta que todos os campos calculados existam na conexão selecionada para a opção “Otimizar” ou “Atualizar”, pois o Tableau não criará campos que ele acredita que não estão sendo usados. Uma boa prática é definir todos os campos calculados na fonte de dados primária, copiá-los à medida que forem necessários para outras conexões e atualizar ou otimizar a extração apenas na fonte de dados primária.

Aviso: um membro confiável da nossa equipe de desenvolvimento afirmou que, embora seja possível criar várias conexões para um único TDE, ele não foi desenvolvido para isso. Essa prática pode causar muitos problemas, se as conexões perderem a sincronia com a estrutura do TDE. O conselho dele: não faça isso.

Atualizando extrações

No Tableau Desktop, para atualizar uma extração, você usa uma opção de menu (Dados > [sua fonte de dados] > Extração > Atualizar), que atualiza os dados e adiciona novas linhas. No Tableau Server, durante ou após o processo de publicação, você pode anexar um agendamento definido por um administrador para atualizar a extração automaticamente. O menor incremento de agendamento permitido é de 15 minutos. O agendamento pode ser definido para o mesmo horário diariamente, semanalmente, etc. Você pode definir uma “janela móvel” para atualizar os dados continuamente para os mais recentes.

Observação: se você quiser atualizar os dados com uma frequência maior do que a cada 15 minutos, use uma conexão de dados em tempo real ou configure um banco de dados de relatório sincronizado.

É possível definir dois agendamentos de atualização para uma única extração:

- Uma atualização incremental apenas adiciona linhas, não altera as linhas existentes.
- Uma atualização completa descarta a extração atual e gera uma completamente nova a partir da fonte de dados.

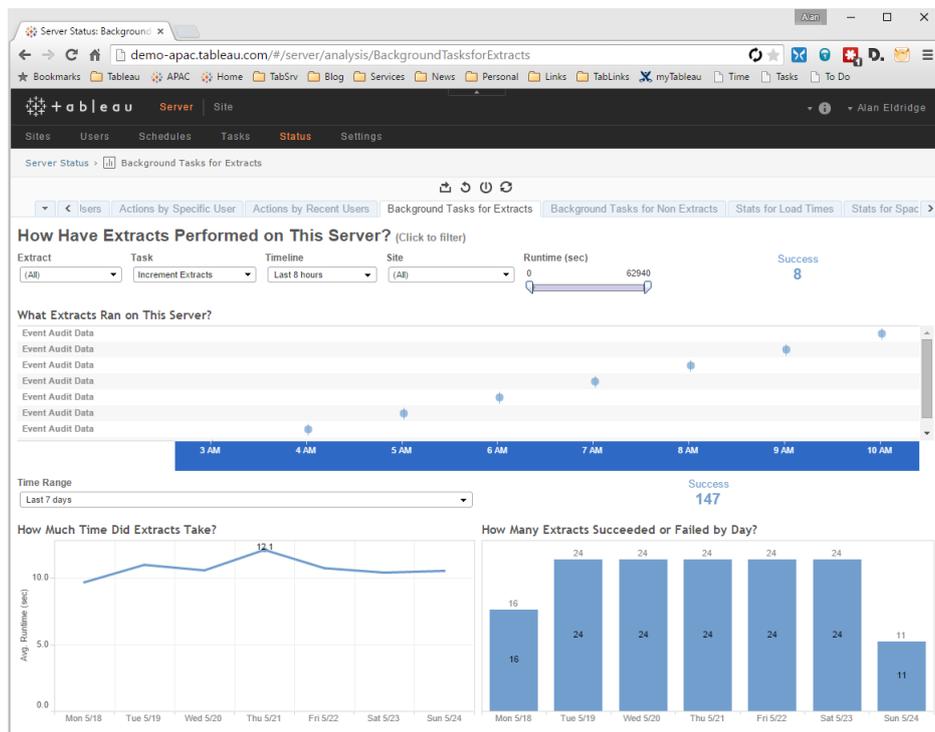
O que acontece se a atualização demorar mais do que o incremento?

Se o tempo de atualização de uma extração demorar mais do que o incremento, as atualizações intermediárias serão ignoradas. Por exemplo, o agendamento é definido para atualizar os dados a cada hora, mas a quantidade de dados é tão grande que a atualização leva uma hora e meia. Antes:

- A primeira atualização começa à 1:00 e termina às 2:30.
- A próxima atualização está agendada para começar às 2:00, mas é ignorada porque a primeira ainda está em execução.
- A próxima atualização começa às 3:00 e termina às 4:30.

Manutenção da extração

As telas de manutenção mostram as tarefas em segundo plano que estão sendo executadas e as que foram executadas nas últimas 12 horas. Um código de cores é usado para mostrar o status dessas tarefas. As telas de manutenção estão disponíveis para administradores e, com as permissões apropriadas, para alguns outros usuários, que podem ter permissões para iniciar uma atualização ad hoc em uma extração. Além disso, por exemplo, se for necessário carregar um banco de dados, você pode definir um disparador para iniciar uma extração após o carregamento do banco de dados.



Também é possível atualizar uma pasta de trabalho de forma incremental ou completa usando a ferramenta de linha de comando com o Tableau Server ou a linha de comando do Tableau.exe com o Tableau Desktop. Se você tiver requisitos complexos de agendamento, poderá invocá-los de uma ferramenta de agendamento externa, como o Agendador de Tarefas do Windows. Essa abordagem será necessária se você precisar de um ciclo de atualização mais curto do que o ciclo mínimo de 15 minutos permitido na interface do Tableau Server.

Por fim, para os usuários do Tableau Online, você pode usar o cliente de sincronização do Tableau Online e definir agendas no Tableau Online para manter atualizados os dados armazenados localmente. Encontre mais informações sobre esse utilitário aqui:

http://onlinehelp.tableau.com/current/online/pt-br/help.htm#qs_refresh_local_data.htm

Governança de dados

Outra forma de se conectar a fontes de dados é usar o Servidor de dados do Tableau Server. O Servidor de dados oferece suporte a conexões em tempo real e a extrações de dados, além de fornecer diversas vantagens em relação às conexões de dados independentes:

- Como os metadados são armazenados centralmente no Tableau Server, eles podem ser compartilhados em várias pastas de trabalho e com diversos autores/analistas. As pastas de trabalho mantêm um ponteiro para a definição de metadados centralizada, e, sempre que são abertas, verificam se houve alguma mudança. Se uma mudança for detectada, o usuário é solicitado a atualizar a cópia incorporada na pasta de trabalho. Isso significa que as alterações na lógica de negócios precisam ser feitas em um só lugar e que podem ser propagadas em todas as pastas de trabalho dependentes.
- Se a fonte de dados for uma extração de dados, ela poderá ser usada em várias pastas de trabalho. Sem o Servidor de dados, cada pasta de trabalho conteria sua própria cópia local da extração. Isso reduz o número de cópias redundantes, o que, por sua vez, reduz o espaço de armazenamento necessário no servidor, bem como os processos de atualização duplicados.
- Se a fonte de dados for uma conexão em tempo real, os drivers da fonte de dados não precisarão ser instalados no computador de cada analista, apenas no Tableau Server. O Servidor de dados atua como um proxy para consultas do Tableau Desktop.

O problema é o ambiente?

Em alguns casos, uma pasta de trabalho pode ter um bom desempenho em um teste com um único usuário, mas, depois de implantada no Tableau Server para ser utilizada por vários usuários, seu desempenho fica ruim. A seção a seguir identifica as áreas em que as diferenças entre os cenários de um único usuário e de vários usuários são impactantes.

Atualização

A equipe de desenvolvimento da Tableau trabalha constantemente para melhorar o desempenho e a usabilidade dos softwares da empresa. Atualizar para a versão mais recente do Tableau Server pode resultar em melhorias significativas no desempenho e na estabilidade do software, eliminando a necessidade de fazer alterações na pasta de trabalho.

Viste a página com as notas da versão do Tableau e atualize para a versão mais recente possível:

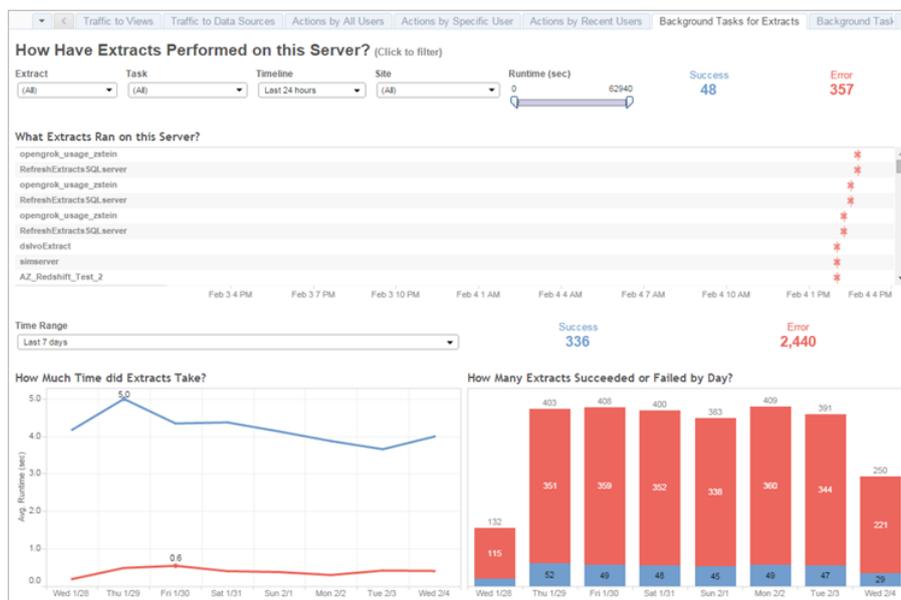
<http://www.tableau.com/pt-br/support/releases>

Teste o Tableau Desktop no Tableau Server

Às vezes, uma pasta de trabalho pode ter um bom desempenho no Tableau Desktop instalado em sua estação de trabalho, mas um desempenho ruim quando visualizada no Tableau Server. Abrir essa pasta de trabalho com uma cópia do Tableau Desktop instalada na máquina do Tableau Server pode ajudar você a descobrir se o problema é da pasta de trabalho ou de alguma configuração do servidor. Com essa abordagem, você pode identificar se é um problema de incompatibilidade de driver ou se há algum problema com a rede, como um roteamento ou configurações de DNS ou proxy ruins.

Separe atualizações e cargas de trabalho interativas

Se o desempenho do servidor estiver ruim, use a exibição administrativa Tarefas em segundo plano para ver seus agendamentos atuais de tarefa de atualização.



Se for possível agendar as atualizações para horários fora de pico, faça isso. Se sua configuração de hardware permitir, você também pode mover os processos do Processador em segundo plano para um nó dedicado do computador de trabalho.

Monitore e ajuste as configurações do Tableau Server

O Tableau Server é fornecido com várias exibições para administradores, criadas para ajudar no monitoramento da atividade do software. As exibições estão localizadas na tabela Análise da página Manutenção do Tableau Server:

Views	Analysis
Traffic to Views	View count, viewers, and viewer behavior for published views.
Traffic to Data Sources	Data source usage, users, and user behavior for published data sources.
Actions by All Users	Actions for all users.
Actions by Specific User	Actions for a specific user, including items used.
Actions by Recent Users	Recent actions by users, including last action time and idle time.
Background Tasks for Extracts	Completed and pending extract task details.
Background Tasks for Non Extracts	Completed and pending background task details (non-extract).
Stats for Load Times	View load times and performance history.
Stats for Space Usage	Space used by published workbooks and data sources, including extracts and live connections.

Para obter mais informações sobre essas exibições, clique no link abaixo:

<http://onlinehelp.tableau.com/current/server/pt-br/adminview.htm>

Também é possível criar exibições administrativas conectando o banco de dados PostgreSQL que faz parte do repositório do Tableau. As instruções podem ser encontradas aqui:

http://onlinehelp.tableau.com/current/server/pt-br/adminview_postgres.htm

Verifique o tempo limite da sessão do VizQL

O tempo limite padrão da sessão do VizQL é de 30 minutos. Mesmo que uma sessão do VizQL esteja ociosa, ela ainda estará consumindo memória e ciclos de CPU. Se você puder fazer isso com um limite inferior, use o `tabadmin` para alterar a configuração `vizqlserver.session.expiry.timeout`:

http://onlinehelp.tableau.com/current/server/pt-br/reconfig_tabadmin.htm#ida6864815-db67-4a51-b8b6-c93617582091

Avalie a configuração do processo

O Tableau Server é dividido em vários componentes chamados serviços. Embora a configuração padrão tenha sido desenvolvida para trabalhar em uma ampla gama de cenários, você também pode reconfigurá-la para atingir diferentes metas de desempenho. Especificamente, você pode controlar em quais computadores os processos são executados e quantos são executados. Obtenha mais informações sobre como otimizar o desempenho do Tableau Server para implantações em um, dois e três computadores aqui:

http://onlinehelp.tableau.com/current/server/pt-br/perf_extracts_view.htm#idd21e8541-07c4-420f-913e-92dcdf5f0c34

Infraestrutura

64 bits

As versões do Tableau Server anteriores ao Tableau 10 podem ser executadas nos sistemas operacionais de 32 bits da Microsoft. No entanto, recomendamos veementemente que nossos clientes instalem seus

ambientes de produção usando a versão de 64 bits. As instalações de 32 bits são recomendadas apenas para atividades de desenvolvimento ou testes.

A partir do Tableau 10, o Tableau Server estará disponível apenas na versão de 64 bits.

Adicione mais CPU/RAM

Independentemente de estar executando o Tableau Server em um computador ou em vários, a regra geral é quanto mais núcleos de CPU e mais RAM, melhor desempenho você terá. Certifique-se de que você atendeu aos requisitos de hardware e software recomendados do Tableau Server (<http://onlinehelp.tableau.com/current/server/pt-br/requ.htm#ida4d2bd02-00a8-49fc-9570-bd41140d7b74>) e consulte Quando adicionar computadores de trabalho e reconfigurar http://onlinehelp.tableau.com/current/server/pt-br/distrib_when.htm#idfdd60003-298e-47e6-8133-3d2490e21e07) para avaliar se você precisa adicionar mais computadores.

O TabMon (mencionado anteriormente neste documento) é uma excelente ferramenta para coletar dados de uso que podem ajudar em seu processo de planejamento de capacidade.

Não desconsidere os volumes de entrada e saída

Algumas ações do Tableau têm muitas entradas/saídas (por exemplo, carregar/criar/atualizar extrações de dados) e se beneficiarão do uso de SSDs, em vez de discos rotacionais. Em seu blog Tableau Love, Russell Christopher publicou dois excelentes textos que falam sobre o impacto do número de núcleos, da velocidade da CPU e de IOPS no desempenho geral. Apesar de seu experimento ter sido feito no AWS, ele pode ser aplicado a todos os ambientes:

<http://bit.ly/1f17oa3>

<http://bit.ly/1f17n5O>

Físico x virtual

Muitos clientes estão implantando o Tableau Server em uma infraestrutura virtualizada. A virtualização sempre sobrecarrega o desempenho, por isso, ele não será tão rápido quanto o de uma implantação em um ambiente físico; embora a tecnologia de hipervisor moderna já tenha reduzido bastante essa sobrecarga.

Quando a instalação for feita em máquinas virtuais, é importante garantir que o Tableau Server tenha recursos dedicados de CPU e RAM. Se ele precisar competir por recursos com outras máquinas virtuais no host físico, seu desempenho pode ser significativamente afetado.

Uma boa fonte de informações sobre como configurar implantações em ambientes virtuais é o whitepaper “Implantando aplicativos extremamente sensíveis à latência no vSphere 5.5” (em inglês) da VMWare. Na página 15 desse whitepaper, há uma lista com ótimas práticas recomendadas para aplicativos sensíveis a latência:

<http://vmw.re/1L4Fyr1>

Navegador

O Tableau utiliza muito o JavaScript para que a velocidade do interpretador de JavaScript no navegador tenha um impacto na velocidade de renderização. Em um campo que evolui com tanta rapidez, a competição entre os navegadores modernos é acirrada, mas vale a pena verificar se a versão e o desempenho do seu navegador podem estar afetando a sua experiência.

Conclusão

Como alguém sábio uma vez disse (e me aconselhou quando decidi criar este documento): “explique o que você vai dizer, diga o que precisa dizer e repita o que você disse”. Sábio conselho.

Então, mais uma vez, estes são os principais pontos para você guardar depois de ler este documento:

- Não existe uma solução mágica. O desempenho pode ser lento por vários motivos. Colete dados para identificar onde o tempo está sendo gasto. Depois, concentre-se nas melhorias da área com processamento mais intenso, passe para a próxima área e assim por diante. Pare quando o desempenho estiver satisfatório ou a proporção entre esforço e recompensa não seja mais viável.
- A lentidão de muitos painéis se deve a um design ruim. Prefira a simplicidade. Não tente mostrar muita informação de uma só vez e use designs de análises orientadas sempre que possível.
- Não trabalhe com dados desnecessários. Use filtros, oculte campos que não estão sendo usados e utilize a agregação.
- Não tente brigar com o Processador de dados. Ele é inteligente e está tentando ajudar você. Por isso, confie que ele criará consultas eficientes.
- Quanto mais organizados seus dados estiverem e melhor corresponderem à estrutura de suas perguntas, mais rápido suas pastas de trabalho serão executadas e mais feliz será a sua vida.
- As extrações são uma forma rápida e fácil de agilizar a execução de pastas de trabalho. Se não precisar ter acesso em tempo real aos dados e não estiver trabalhando com bilhões de linhas, você deveria experimentá-las.
- Strings e datas são lentas, números e booleanos são rápidos.
- Embora este documento seja baseado nas melhores orientações de muitos especialistas, suas recomendações são apenas isso: recomendações. Você precisa testar quais delas proporcionam uma melhora real no desempenho em seu caso específico.
- Se você estiver trabalhando com pequenos volumes de dados, muitas das orientações neste documento podem ser desconsideradas, já que você pode corrigir manualmente o erro. De qualquer forma, não custa nada seguir essas recomendações para todas as suas pastas de trabalho, porque nunca se sabe quando o volume de dados crescerá.
- A prática leva à perfeição.

Agora, vá em frente e crie muitas pastas de trabalho eficientes!