



# IBM Informix Warehouse Accelerator

## Performance é tudo

Keshava Murthy  
Senior Technical Staff Member  
IBM Informix Development

-----  
Translated to Portuguese by  
Carolina L Leme, Heloisa Maria Brites  
IBM Worldwide Technical Support

# IBM





**Imagine** seus analistas de negócios, os executivos de nível C, ficando os seus relatórios em questão de segundos, em vez de horas. Imagine o aumento da velocidade de análise e insights. Imagine melhorar a velocidade de execução de negócios com os pontos de dados disponíveis dentro de segundos. Imagine melhorar o seu desempenho de consulta armazém sem constantemente monitorar e ajustar o sistema. Imagine fazer isso sem cubos de construção, tabelas de resumo, índices, estatísticas ou estratégia de particionamento criando.

Após extensos testes de Informix Warehouse Accelerator, Ashutosh Khunte, Senior Database Architect, da Skechers, EUA, diz: **Antes de usar o Informix Warehouse Accelerator, inventário consultas complexas e análise de vendas no armazém da empresa com mais de um bilhão de linhas levou em qualquer lugar de alguns minutos para 45 minutos para ser executado. Quando rodamos aquelas mesmas perguntas utilizando Informix Warehouse Accelerator, que terminou em 2 a 4 segundos! Isso significa que eles correram 60-1400 vezes mais rápida, com um factor de aceleração média de mais de 450 - tudo sem qualquer edifício ou índice de cubo, de ajuste de consulta ou de alterações de aplicativo!**

Lester Knutsen da Advanced Data Tools, Informix Data champion, levou os dados de seus clientes dos Estados Unidos Departamento de Agricultura e passou a carga de trabalho. Ele diz: **"Ele oferece um desempenho muito impressionante, com consultas em execução 30 vezes mais rápido do que anteriormente. A tecnologia colunar poupa muito tempo de processamento, que reduziu nosso tempo de carga de trabalho de 9,5 horas para 15 minutos, tudo sem qualquer ajuste do banco de dados ou a necessidade de gerenciar o armazenamento físico.**

Thomas Gemesi, ATG IT Consulting GmbH diz, **Informix Warehouse Accelerator (IWA), traz a capacidade de armazenamento de dados e OLTP (Online Transaction Processing) em uma única plataforma. As consultas podem ser executados em questão de segundos, sem precisar fazer nenhum investimento ferramenta adicional.**

**A mudança é constante.** Necessidade de velocidade é constante. Você tem que entender os padrões, prever tendências e ajustar o fluxo de negócios. A análise dos dados ajudam a entender as tendências e mudanças nos negócios. Fazendo isso ninguém mais rápido do que outra coisa, fazendo isso de forma consistente, fazendo isso com menor TCO lhe dará uma vantagem sobre a concorrência.

**Informix Warehouse Accelerator** oferece uma performance extrema, removendo a maior parte do ajuste necessário para armazenagem de dados tradicionais. Ele foi concebido para processar dados de grandes dimensões em poucos segundos. Ele foi projetado para fornecer o negócio com os dados certos no momento certo, sem aumentar a manutenção ou mudança de infra-estrutura do aplicativo.

**Inovação** é a chave. A diferença de desempenho entre o processador para acelerar o acesso à memória e processador para acelerar o acesso ao disco está aumentando com o tempo. Sistemas de banco de dados tradicionais assumem configurações de memória baixa e otimizar as operações para minimizar / S de disco e uso de buffer para manter os dados recentemente acessados na memória. Caindo os preços da memória e sistemas acessíveis e com capacidade de memória terabyte exigem um novo design. A IBM tem sistemas de apoio à 3 TB de RAM e essa capacidade deve aumentar. As

façanhas do acelerador esta tendência através da compressão e cache de todos os dados na memória e IO de disco eliminando. Processadores estão adicionando mais núcleos e maior cache on-chip com todas as liberações. As façanhas do acelerador vai a tendência ao paralelismo, minimizando a sincronização e otimização dos algoritmos para explorar no cache do chip. Isso dimensiona o desempenho da consulta a novas alturas e conduz a um design simples e elegante.

## Poder da Simplicidade

A perfeição é atingida, não quando não há mais nada a acrescentar,  
mas quando não há mais nada para levar.

-- Antoine de Saint-Exupery

O acelerador tem três princípios fundamentais:

- Aceleração sem ajuste manual para cada carga de trabalho
- Apoio existentes ferramentas de negócios e aplicações.
- Trabalhar com infra-estrutura existente de armazém

Se você já tem Informix em seu ambiente, as velas de acelerador em seu ambiente existente. Para novas implantações, você tem opções de plataforma flexível e aberta de integração com servidor de banco de dados Informix.



Basta dar os dados para o acelerador, ele está pronto para mostrar o seu desempenho máximo.

**Nenhum índice de criar, nenhum conselheiro índice necessário, sem a reorganização do índice exigido.**

O motor do acelerador de processamento de consulta logicamente scans bilhões de linhas em milissegundos ou segundos. O fundo colunar representação de dados, processamento de consultas em comprimido, algoritmos inovadores para explorar recursos do processador moderno anula a necessidade de índices.

**Não há estatísticas para coletar.** Otimizadores tradicionais contam com coleta de estatísticas regulares para criar melhores planos de consulta. Com o acelerador, junte-se as encomendas são feitas automaticamente, a estrela de juntar os planos são usados de forma consistente. A falta de índices

simplifica as opções e otimização de tempo de execução (explicado mais tarde no papel) anula a necessidade de coleta de estatísticas e assessores coleta de estatísticas.

**Não particionamento (fragmentação) esquemas para criar.** Os dados são automaticamente particionado verticalmente e horizontalmente. As consultas também beneficiam de poda partição vertical e horizontal (aka eliminação fragmento), devido à célula de armazenamento baseado em colunas de profundidade. Isso elimina o planejamento para os regimes de particionamento de tabela.

**Nenhum ajuste para cada consulta ou carga de trabalho.** Durante a instalação do acelerador, é necessário fornecer a memória ea configuração básica de armazenamento. planos consistentes, eliminação de IO de disco, rápido exames e junta-se a eliminar a regulagem de tempo de execução.

**Não gerenciamento de armazenamento.** Todos os dados são armazenar na memória com apenas uma cópia da imagem em memória no disco. Não ha necessidade de planejar e criar espaços de armazenamento para tabelas, índices.

**Nenhum hardware caro.** O acelerador é projetado para rodar em hardware e evoluir com a sua infraestrutura. O acelerador é executado em computadores Linux / Intel. Ele se integra com banco de dados Informix em Linux / Intel, AIX / Power, HP-UX/Itanium, Solaris / Sparc.

**Não alterações banco de dados.** As façanhas do acelerador o esquema lógico existente em seu data warehouse.

**Nenhuma alteração no aplicativo.** As fichas de acelerador para o servidor de banco de dados Informix como um recurso. O servidor de banco de dados Informix sabe a data marts, que são armazenados no acelerador e automaticamente encaminha as consultas pertinentes ao acelerador. Você aplicação ou ferramentas requerem nenhuma mudança.

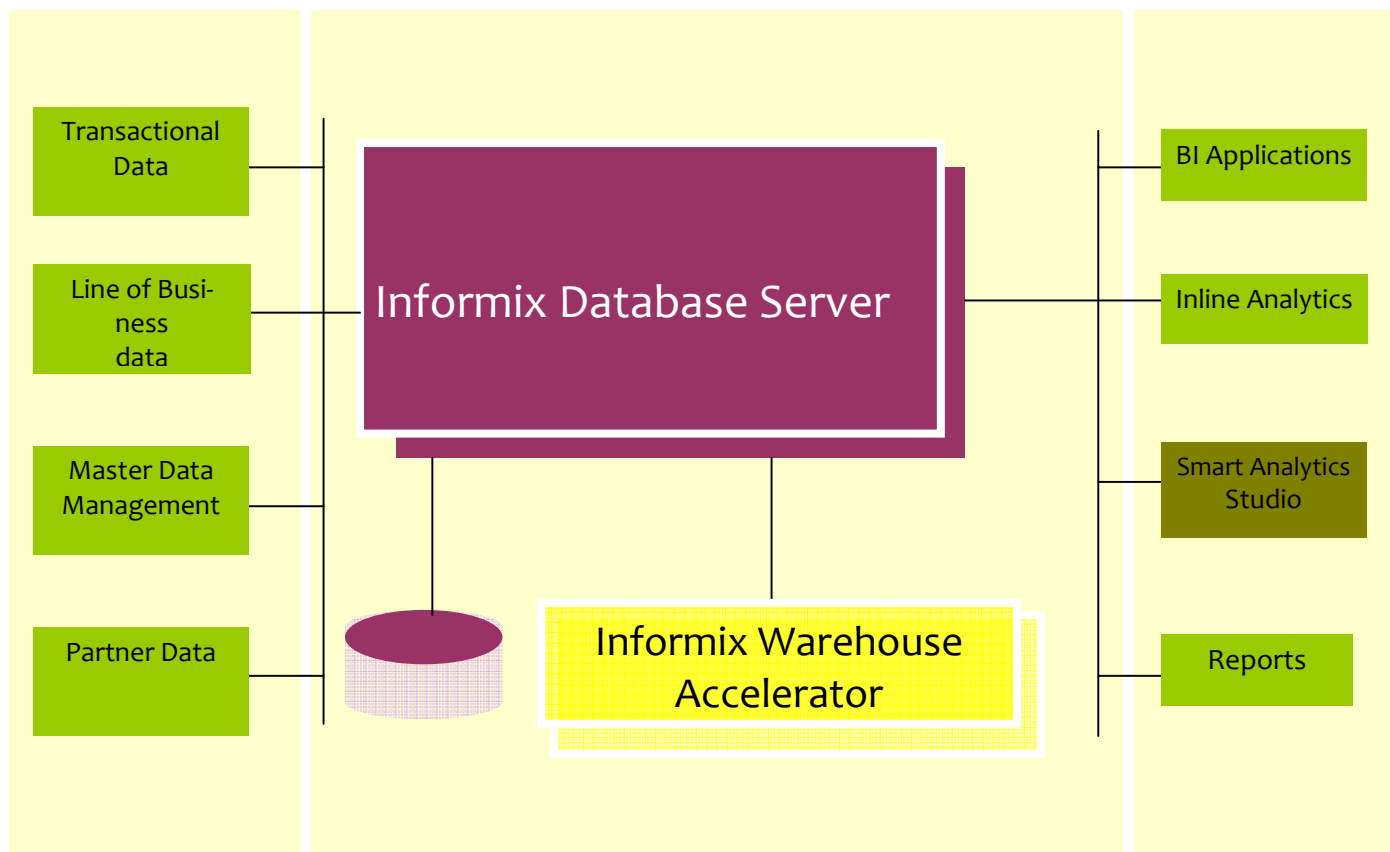
**Sem tabelas sumárias ou visões materializadas para criar.** A tabela acelerador exames e junta-se pelo menos uma ordem de grandeza mais rápido do que os bancos de dados tradicionais. Isso elimina a criação e manutenção de tabelas de resumo e da utilização de consultores relacionados.

**Não para tamanho da página de configuração ou tamanho do bloco necessário.** A tecnologia colunar profunda determina automaticamente e otimiza o tamanho da célula de memória.

**Não alocação de espaço temporário ou de controlo.** O conjunto de resultados intermediários também é compactado e ocupa menos espaço.

**Não dicas de consulta e otimizador de consultas acelerado.** O otimizador usa de junção em estrela planos de forma consistente. O processador de consultas acelerador ajusta a juntar-se encomendas em função da estatística de tempo de execução.

# Informix Warehouse arquitetura



Informix database server é um servidor escalável e altamente disponível utilizado por dezenas de milhares de clientes da unidade de missão crítica aplicações transacionais e análise. Informix database server é um servidor de banco de dados completo com as ferramentas de ETL warehouse, construído em apoio à gestão de tempo cíclica de dados, as operações online, compactação profunda e otimização de consultas e técnicas de processamento de dados complexos projetados para armazenagem de carga de trabalho. Clientes Informix podem usar as ferramentas de BI como Cognos, Microstrategy, etc, para analisar e melhorar seus negócios.

Tradicionalmente, as consultas de data warehouse são chamadas complexas por uma razão. Essas consultas milhões de acesso e bilhões de linhas, os resultados intermédios são muito grandes, as consultas têm melhor desempenho quando estão em paralelo. A técnica de otimização de junção em estrela é projetado para lidar com isso no sistema tradicional. Tudo isso leva um DBA experiente para entender a carga de trabalho e ajustar os parâmetros do sistema e os índices que atendam a carga de trabalho.

Informix Warehouse Accelerator oferece uma performance de consulta warehouse, elimina as tarefas de ajuste de banco de dados e faz tudo isso sem precisar de mudanças para a sua aplicação existente e ambiente de ferramentas. O restante deste artigo explica a tecnologia de aceleração, o seu uso e sua integração com Informix.

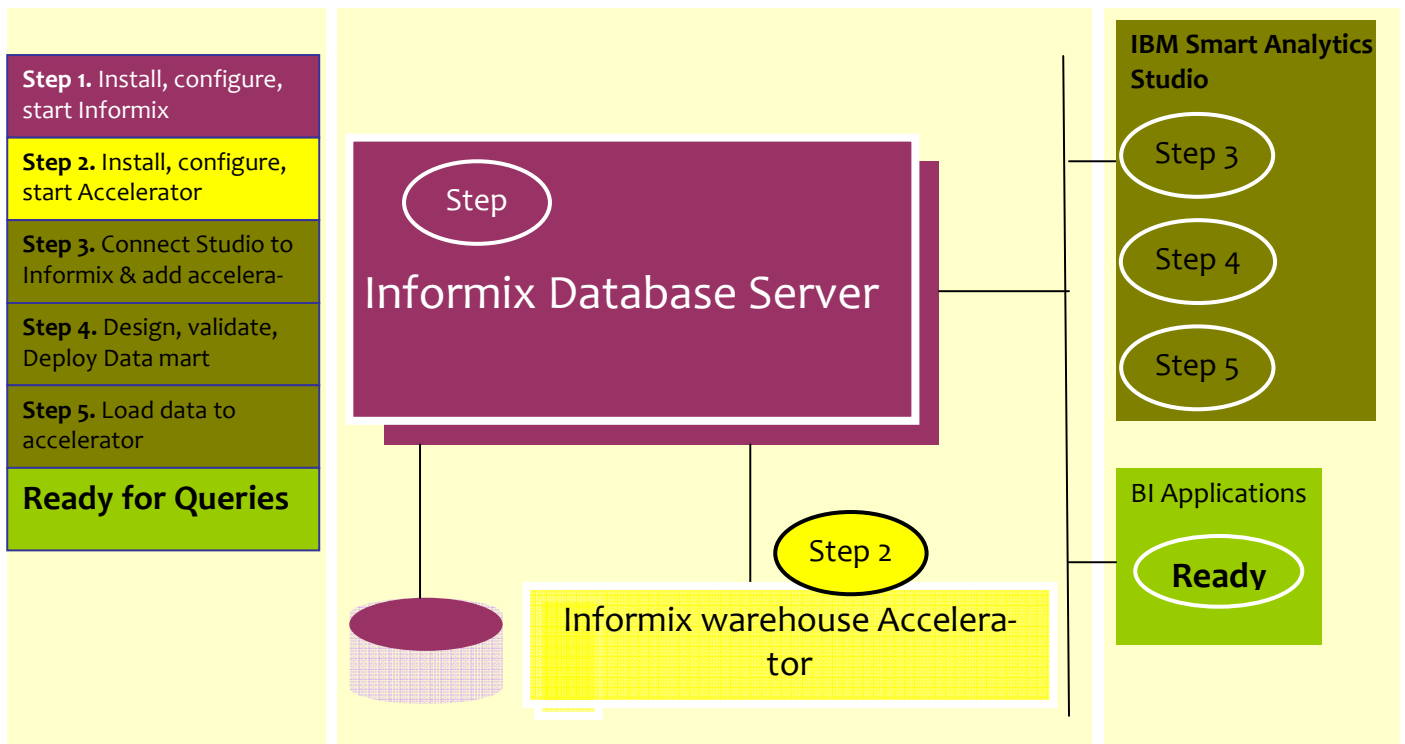
# Implantando Informix Warehouse Accelerator

Informix Warehouse Accelerator sempre é executado em um sistema de alta performance operacional Linux em um servidor Intel. O servidor de banco de dados Informix e o acelerador pode ser executado no mesmo servidor ou em servidores diferentes. Você pode executar o servidor de banco de dados Informix em Linux / Intel, AIX / Power, HPUX / HP Itanium, Solaris / Sparc e executar o acelerador em um servidor Linux em Intel.

O acelerador é parte de Informix Ultimate Warehouse Edition (IUWE), que inclui servidor de banco de dados Informix (Ultimate Edition) e Informix Warehouse Accelerator. O Acelerador de mídia inclui o binário acelerador, e o IBM Smart Analytics Optimizer Studio, uma ferramenta para configurar e gerenciar o acelerador com o servidor de banco de dados Informix. Abaixo está a seqüência de operações que vão de zero a implantação rapidamente. O produto inclui um Guia Rápido e um Guia de Administração com documentação detalhada sobre como instalar e configurar o Informix e o acelerador. Você verá que o acelerador requer apenas alguns passos simples de configuração: localização do sistema de arquivos para backup de dados, quantidade de memória e recursos da CPU. Informações sobre a configuração é discutido mais adiante no documento e no Guia de Administração.

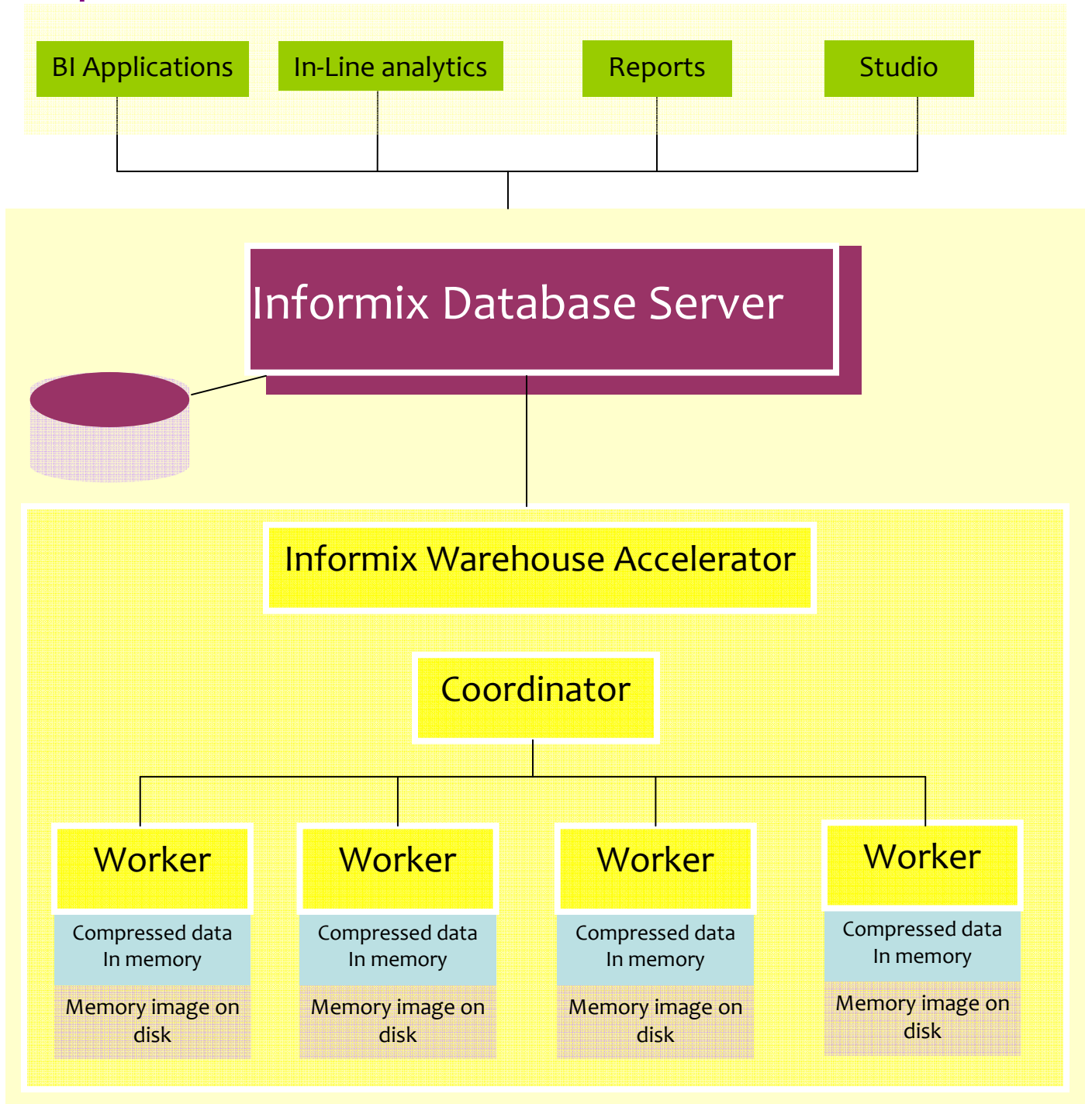
IBM Smart Analytics Optimizer Studio é uma ferramenta baseada em Eclipse usado para gerenciar os aceleradores, e definir e implantar o data marts de Informix para o acelerador. O Studio vem em duas versões: uma versão para Linux e outra para Windows. Você pode usar a versão para Linux no mesmo computador como o servidor de banco de dados Informix ou em computador diferente. Para usar a versão Windows, transferir o binário de extração automática para um computador com Windows e instalá-lo.

## Etapas da implantação do servidor de banco de dados Informix eo acelerador.





# Componentes do Acelerador



**O acelerador** está ligado ao servidor de banco de dados Informix através de uma rede TCP / IP. Se o acelerador e o servidor de banco de dados estão no mesmo computador, eles se comunicam usando uma conexão TCP / IP de loopback. O acelerador consiste em processos de coordenador e trabalhador. Informix se comunica com o acelerador usando com o coordenador de processos. Coordenador e trabalhador de processos compartilham a responsabilidade de processamento de consulta. A configuração que você especificar determina o número destes processos, e a quantidade de memória e recursos da CPU utilizada.

Adicione as informações do acelerador através do Studio após Informix e acelerador estiver executando [Veja o manual para maiores detalhes]. Em caso de sucesso desta etapa, Informix adiciona informações de conexão do acelerador em seu arquivo sqlhosts. A entrada do arquivo sqlhosts para o acelerador será parecido com este:

```
sales_acc      group  -- c=1,a=4b3f3f457d5f552b613b4c587551362d2776496f226e714d75217e22614742677b424224
sales_acc_1    dwsoc tcp    127.0.0.1      21022  g=sales_acc
```

O nome do acelerador é sales\_acc. Informix cria novo grupo com esse nome. O valor hexadecimal é o código de autenticação usado para garantir que apenas comunica o banco de dados Informix com o acelerador, sales\_acc. O nome do coordenador é sales\_acc\_1. O dw protocolo de banco de dados é usado para comunicação via TCP / IP. O protocolo de banco de dados é muito próximo ao protocolo DRDA, otimizado para Informix e comunicação acelerador. 127.0.0.1 (TCP / IP loopback endereço) indica que o acelerador está sendo executado no mesmo computador como Informix. Com um grande número de trabalhadores, terá vários coordenadores para a manipulação de failover. Nestas configurações, haverá uma entrada para cada coordenador.

**Coordenador** tem três funções importantes.

1. Este é o ponto principal de comunicação para Informix servidor de banco de dados. servidor de banco de dados se conecta com o coordenador de enviar mais dados e é também o principal contato para enviar a consultas e obter resultado.
2. Durante a fase de carregamento de dados (Etapa 8), que distribui os dados entre os vários trabalhadores. O coordenador, em seguida, recolhe a compressão dicionário inteiro, mescla-lo e redistribui o dicionário, todo mundo está usando a mesma referência dicionário.
3. Durante a fase de processamento da consulta, o coordenador recebe a consulta do servidor de banco de dados, envia a consulta para cada trabalhador, obtém o conjunto de resultados intermediários, mescla os grupos e, em seguida, descomprime os dados e classifica os dados, se necessário, antes de enviar os resultados finais ao o servidor de banco de dados Informix.

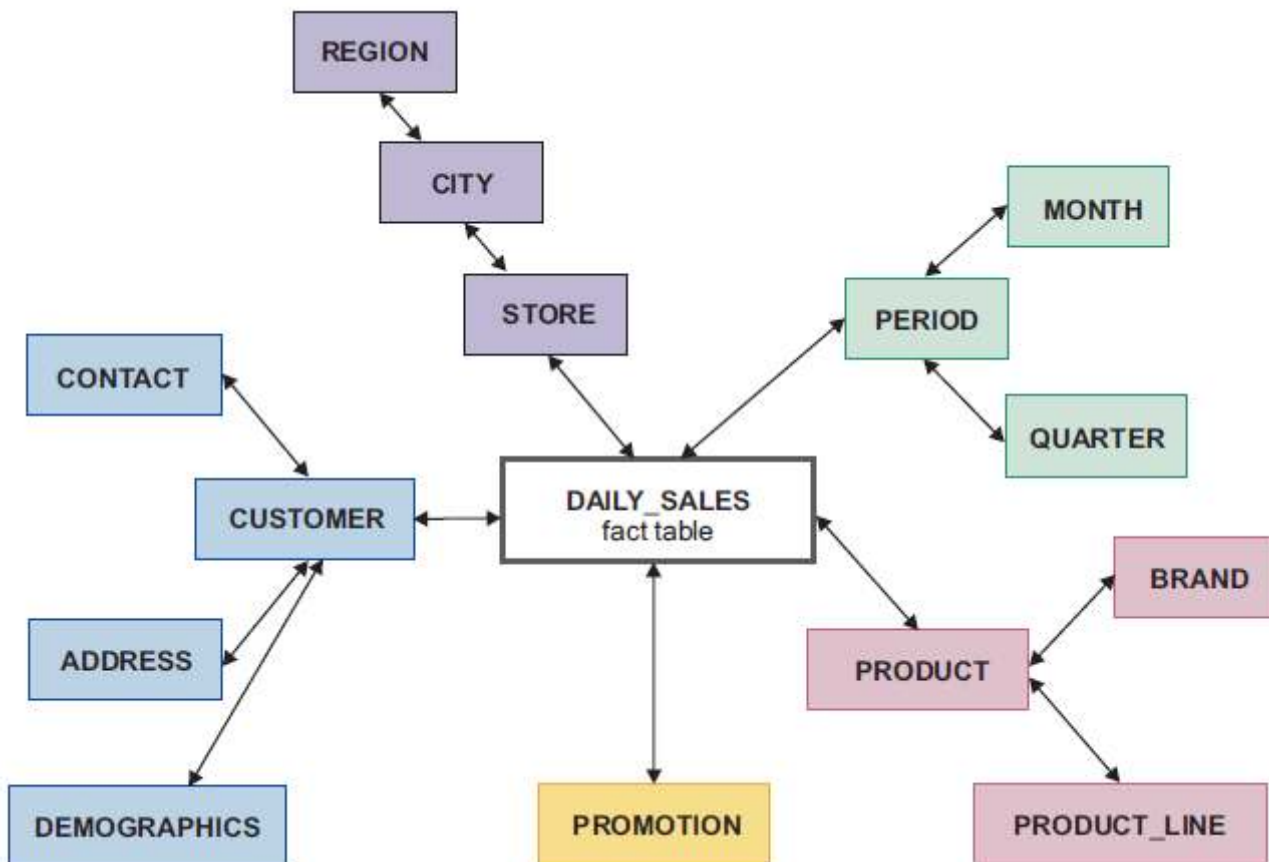
Os **processos de trabalho** têm duas funções importantes.

1. Na fase de carregamento de dados, cada trabalhador analisa os dados de entrada usando o particionamento de frequência, automaticamente as partições de dados verticalmente e horizontalmente em pilhas e comprime os dados usando técnicas de Deep Colunar. Uma vez que os dados são compactados, os dados são gravados em um disco de recuperação. Nós iremos rever as técnicas Deep Colunar depois no papel.

2. A segunda função dos processos de trabalho é o processamento de consultas em dados compactados. Cada trabalhador mantém cópia compactada das tabelas de dimensão e de parte da tabela de fatos. Cada trabalhador retorna resultados intermédios, para o coordenador. O processamento de consultas é feito 100% in-memory.

Os coordenadores e os trabalhadores trabalham juntos para paralelizar a execução de cada consulta e retornar os resultados rapidamente.

Wikipedia define Data Mart como um subconjunto do data warehouse, geralmente voltadas para uma linha específica de negócios ou da equipe. Nós usaremos esta definição neste trabalho. Sua enterprise data warehouse em Informix poderia conter tudo, de vendas, estoque, atendimento ao cliente e dados do mercado. Fora isso, você acelerar o fornecimento de data marts valor muito elevado. Por exemplo, o gerente de vendas pode desejar para analisar dados de vendas e estoque para compreender as tendências e criar incentivos adequados vendas. Assim, você só precisa acelerar a data marts necessários para isso com o fato de as tabelas de vendas e inventário. No contexto do accelerator, a cada data mart contém um ou mais esquema floco de neve. Cada esquema floco de neve tem uma tabela de fatos e tabelas de dimensões relacionadas. No seguinte esquema floco de neve, DAILY\_SALES é a tabela de fatos e está relacionado com as dimensões que descrevem os fatos.



A seguinte seção explica os passos desde a elaboração do data mart usando o acelerador. Estes passos referem-se ao diagrama na seção *Implantando o Informix Warehouse Accelerator*.

Use o IBM Smart Analytics Optimizer Studio para **desenhar** e implantar data marts. Um data mart define fato e dimensão das tabelas e seus relacionamentos. As tabelas de dimensão normalmente contêm fileiras um número significativamente menor que as tabelas de fato, por exemplo, informações sobre o produto, informações de clientes. Em alguns casos, as tabelas de dimensão podem ser muito grandes, por exemplo, todos os residentes da Califórnia. Depois de ter identificado as tabelas para criar um data mart, você tem que definir a relação entre o fato e todas as tabelas de dimensão.

A etapa de **validação** do data mart garante que todos os relacionamentos entre as tabelas estão definidos. Certifique-se de corrigir eventuais erros nesta etapa antes de implantar o data mart.

Na etapa de **implantação**, o IBM Smart analytics Optimizer studio envia a definição de data mart no formato XML para o acelerador que envia de volta à definição do SQL. Esta definição é guardada como uma VISÃO dentro de catálogos do sistema Informix com uma flag especial e informações relacionadas. Esta visão é conhecida como Accelerated Query Table (AQT). Este AQT é usado mais tarde para coincidir com as consultas e redirecionar consultas correspondentes ao acelerador.

Aqui está um exemplo com AQT DAILY\_SALES como a tabela fato e suas dimensões: produto, loja, cliente, promoção. Este é criado automaticamente e é dada aqui para suas informações. O usuário ou DBA não usa essa visão. Da aplicação SQL ou ferramentas de trabalhos, como de costume, usando tabelas definidas no esquema.

```
create view "dwa"."aqt2dbca0d9-509d-434b-9cc9-4a12c6de6b3d"
("COL16", "COL17", "COL18", "COL19", "COL20", "COL21", "COL22", "COL23", "COL24", "COL25", "COL2
6", "COL27", "COL28", "COL29", "COL30", "COL31", "COL32", "COL33", "COL34", "COL35", "COL36", "CO
L37", "COL38", "COL39", "COL40", "COL41", "COL42", "COL43", "COL44", "COL45", "COL46", "COL47", "
COL07", "COL08", "COL09", "COL10", "COL11", "COL12", "COL13", "COL14", "COL15", "COL48", "COL49"
, "COL50", "COL51", "COL52", "COL53", "COL54", "COL55", "COL56", "COL57", "COL58", "COL59", "COL6
0", "COL61", "COL01", "COL02", "COL03", "COL04", "COL05", "COL06", "COL62", "COL63", "COL64", "CO
L65", "COL66", "COL67", "COL68", "COL69", "COL70", "COL71", "COL72", "COL73", "COL74", "COL75", "
COL76", "COL77", "COL78", "COL79", "COL80", "COL81") as
select x0.perkey ,x0.storekey ,x0.custkey ,x0.prodkey ,x0.promokey
, x0.quantity_sold ,x0.extended_price ,x0.extended_cost ,x0.shelf_location
, x0.shelf_number ,x0.start_shelf_date ,x0.shelf_height ,x0.shelf_width
, x0.shelf_depth ,x0.shelf_cost ,x0.shelf_cost_pct_of_sale
, x0.bin_number ,x0.product_per_bin ,x0.start_bin_date ,x0.bin_height
, x0.bin_width ,x0.bin_depth ,x0.bin_cost ,x0.bin_cost_pct_of_sale
, x0.trans_number ,x0.handling_charge ,x0.upc ,x0.shipping
, x0.tax ,x0.percent_discount ,x0.total_display_cost ,x0.total_discount
, x1.perkey ,x1.calendar_date ,x1.day_of_week ,x1.week ,x1.period
, x1."year" ,x1.holiday_flag ,x1.week_ending_date ,x1."month"
, x2.prodkey ,x2.upc_number ,x2.package_type ,x2.flavor ,
x2.form ,x2.category ,x2.sub_category ,x2.case_pack ,x2.package_size
, x2.item_desc ,x2.p_price ,x2.category_desc ,x2.p_cost ,x2.sub_category_desc
, x3.storekey ,x3.store_number ,x3.city ,x3.state ,x3.district
, x3.region ,x4.custkey ,x4."name" ,x4."address" ,x4.c_city
```

```

,x4.c_state ,x4.zip ,x4.phone ,x4.age_level ,x4.age_level_desc
,x4.income_level ,x4.income_level_desc ,x4.marital_status
,x4.gender ,x4.discount ,x5.promokey ,x5.promotype ,x5.promodesc
,x5.promovalue ,x5.promovalue2 ,x5.promo_cost
from
((((("informix".daily_sales x0 left join "informix".period x1 on (x0.perkey
= x1.perkey ) )left join "informix".product x2 on (x0.prodkey
= x2.prodkey ) )left join "informix"."store" x3 on (x0.storekey
= x3.storekey ) )left join "informix".customer x4 on (x0.custkey
= x4.custkey ) )left join "informix".promotion x5 on (x0.promokey
= x5.promokey ) );

```

Na etapa do **Loading**, um snapshot de informação do banco de dados é enviado para o acelerador. Nesta fase, o acelerador distribui a informação para cada um de seus workers. O worker analisa a informação para ocorrência de valores frequentes, e o relacionamento entre as colunas e então particiona a informação vertical e horizontalmente. Uma vez que a partição está decidida, a informação é comprimida utilizando um processo colunar profundo descrito neste artigo. Note que nesta fase, a informação é comprimida e mantida na memória com uma cópia no disco para persistência. Indexes não são criados, nem tabelas sumário ou cubos são criados. A informação pode ser atualizada periodicamente ( por exemplo, toda noite) do banco de dados do servidor Informix.

Assim que o loading está completo, o data mart do acelerador está pronto para ser usado. Apenas dê as informações para o acelerador e já comece a experienciar o pico da performance. O acelerador Informix warehouse inclui o command line para projetar, validar, implantar e carregar. Isto é muito útil para escrever scripts para automatizar este processo.

As **Consultas** das informações envolvem tipicamente a junção da tabela fact com uma ou mais dimensões e então a busca por padrões específicos dentro das informações. No exemplo abaixo, a tabela fact web\_sales é unida a 4 outras dimensões. Se você criou e implantou um data mart que inclui estas tabelas, Informix vai fazer a correspondência da consulta a uma view específica do data mart (Accelerated Query Table) e então enviar a consulta para o acelerador. Isto funciona como uma consulta distribuída do banco de dados do servidor Informix para outro servidor. O resultado retorna na mesma conexão e é enviado de volta para a aplicação do cliente. Para a aplicação do cliente, isto é transparente, exceto se o cliente receber o resultado muito mais rápido.

```

select first 100 i_item_id,
      avg(ws_quantity) avg_quantity,
      avg(ws_list_price) avg_list_price,
      avg(ws_coupon_amt) avg_coupton_amt,
      sum(ws_sales_price) sum_sales_price
from web_sales, customer_demographics, date_dim, item, promotion
where ws_sold_date_sk = d_date_sk and
      ws_item_sk = i_item_sk and
      ws_bill_cdemo_sk = cd_demo_sk and
      ws_promo_sk = p_promo_sk and
      cd_gender = 'F' and
      cd_marital_status = 'M' and
      cd_education_status = 'College' and

```

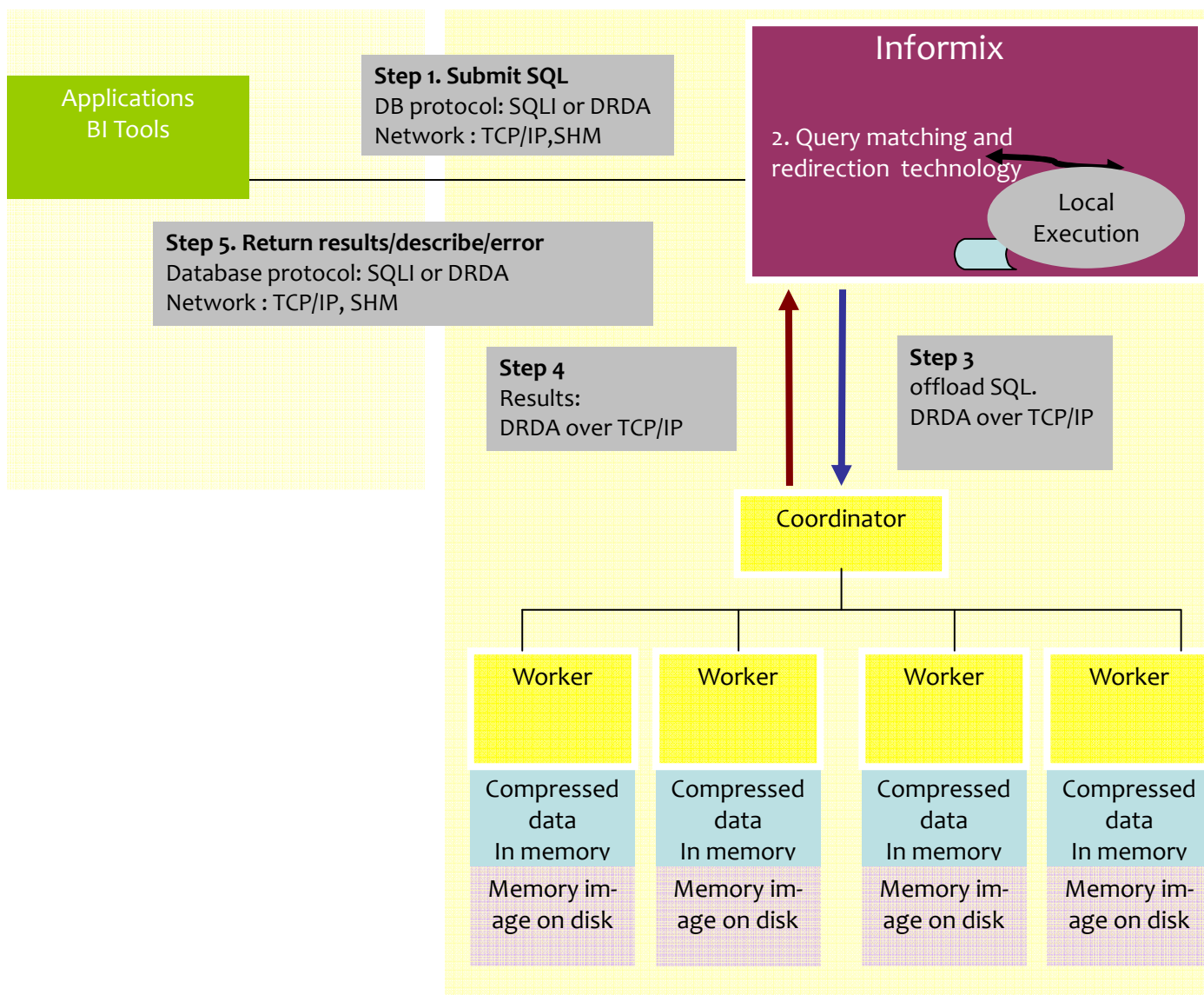
```

(p_channel_email = 'N' or p_channel_event = 'N') and
d_year = 2001
group by i_item_id;
order by sum(ws_sales_price) desc;

```

O acelerador requer que as consultas se juntem às tabelas fato com zero ou mais dimensões e se juntem a cada tabela usando as chaves de junção especificadas quando definindo o relacionamento do data mart. Também suporta inner joins e left joins com a tabela fato no lado dominante. Veja o Guia do Administrador para detalhes sobre qualificação de consulta. O acelerador roda cada consulta em uma base de servidor first come first sem interrupção. Todas as informações e resultados intermediários estão na memória. Cada worker tem uma cópia da dimensão para unir-se. Cada join thread tenta colocar em cache a tabela hash no cache L2 para otimizar o acesso à memória. O worker processa scan e join independentemente com pouca troca de informações e sincronização com outros workers. É típico para cada consulta se completar em poucos segundos comparados a minutos e horas num sistema tradicional. Portanto, o acelerador enfileira as requisições de consulta e roda uma consulta após a outra.

## Fluxo da Consulta com Acelerador



# Notas sobre Configuração

Durante a instalação do Acelerador Informix Warehouse, vai ser pedido para você configurar o número de nodes (coordenadores, workers), memória para os workers, e configuração de memória para o nó coordenador. O número de coordenadores e workers vai ser determinado automaticamente. Por exemplo, a definição de 5 nodes vai automaticamente criar 1 coordinator e 4 nós workers. Detalhes estão descritos no Guia do Administrador. Considere uma configuração com 4 workers e um coordinator. Vamos dizer que você implemente um data mart com as tabelas sales fact e customer, store, time dimentions. As tabelas de dimensão customer, store e time são comprimidas e mantidas em memória privada de cada worker. Assim, haverá 4 cópias das tabelas de dimensão; as linhas da tabela fato sales serão igualmente divididas dentre os 4 workers. Cada worker manterá as tabelas de dimensão customer, store, time e então 25% das linhas da tabela fato – sales neste caso – na memória principal.

A taxa de transmissão de dados aumenta tipicamente conforme você aumenta o número de workers, assumindo que haja capacidade de CPU suficiente. Maior número de workers ajuda na velocidade de processo da consulta, embora menos dramaticamente. O efeito do número de workers na consulta depende da consulta também.

Dito isto, quanto de memória você deve alocar para cada worker? Quanto de memória é necessário para o sistema?

Geralmente, nós temos visto uma taxa de compressão 3:1 dos dados de Informix não comprimidos para os dados comprimidos do Acelerador Informix Warehouse na memória. Se a soma da tabela fato sales e as tabelas de dimensão customer, store, time é por volta de 100Gb and a maioria disto é consumida pela tabela sales, você vai precisar de cerca de 33Gb de memória para armazenar os dados. Você pode descobrir o tamanho das tabelas facilmente usando o Open Admin Tool (OAT) ou consultando os catálogos diretamente.

Cada worker necessita de tempo de execução de memória suficiente para armazenar resultados intermediários no tempo de execução. Workers alocam e liberam a memória necessária para processamento de consultas dinamicamente. Planejar aqui é similar ao planejamento de espaço temporário no Informix. Quanto de resultados intermediários, classificação você espera do workload? O quão relacionados estão seus dados e quantos grupos haveriam em cada categoria? Embora seja difícil dizer precisamente, nós achamos que ter outros  $1/5^{\text{th}}$  a  $1/3^{\text{rd}}$  do tamanho de dados é normalmente suficiente.

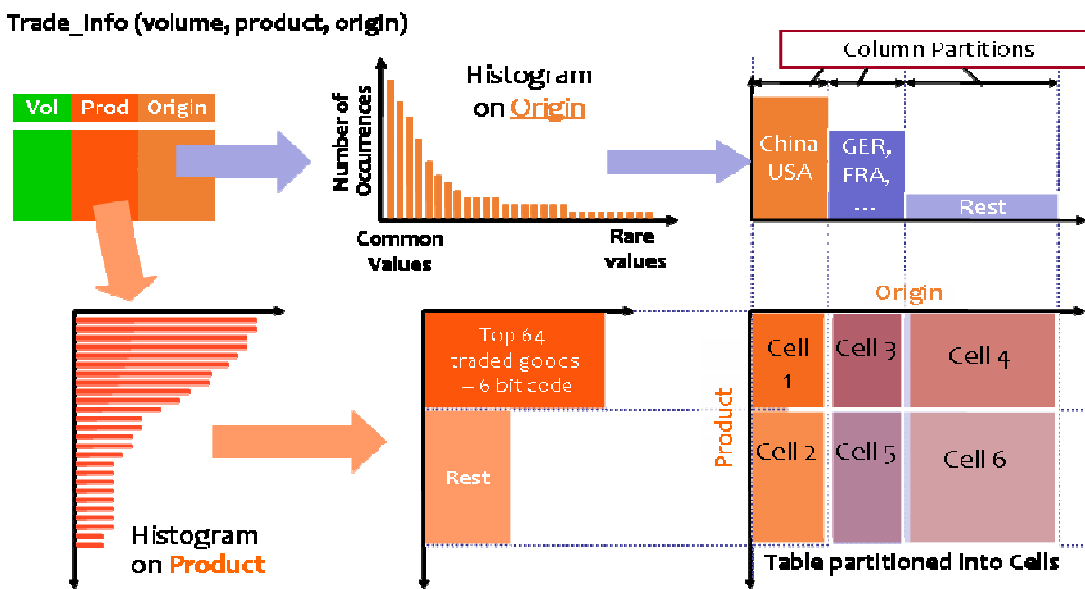
O estado final do processamento de consultas acelerado é feito no coordenador. O coordenador precisa de memória para fusão, descompressão, e classificação do resultado. De novo, a memória requerida aqui depende do tamanho do resultado que você espera. Lembre-se, quando você emite a consulta com a cláusula FIRST, “por exemplo: SELECT FIRST 1000... ORDER BY sum(sales.amount)”, o coordenador vai ter que ordenar todos os dados para conseguir os primeiros 1000. Claro, uma vez que o coordenador cria os primeiros 1000 grupos, ele pode realocar ou rejeitar grupos novos.

# Pico da Performance

A performance extrema com o acelerador é o resultado cumulativo de tecnologias desenvolvidas pela pesquisa e desenvolvimento da IBM. Este artigo fornece uma visão global das tecnologias e técnicas usadas no acelerador para aumentar o desempenho e eliminar ajustes e tarefas de manutenção. Artigos listados na sessão de referência tem mais detalhes sobre teorias e técnicas subjacentes. Estes artigos são publicados em revistas conhecidas e revisadas ou em conferências. A IBM arquivou requerimentos de patentes sobre estas técnicas.

Profunda tecnologia colunar vai além do armazenamento colunar tradicional. Extrema compressão e processamento de consultas sobre dados comprimidos elimina IO no disco durante o processamento de consultas e permite grande armazenamento na memória. Exploração de tecnologias multinúcleo e SIMD te traz velocidade incrível sem indexes ou tabelas de sumário. Vamos dar uma olhada em cada técnica agora.

## Frequência de Particionamento



Cada tabela no data mart é analisada em busca de valores que ocorrem frequentemente nas colunas e grupos de colunas relacionados para determinar as colunas ideais para combinar e formar um **tuplet**. Um **tuplet** é uma fração de uma tupla. Uma tupla é uma linha completa. O benefício da codificação Huffman é que os valores ocorridos mais frequentemente são codificados com menor número de bits. Na figura acima, os 64 primeiros valores resultantes são combinados na origem com os valores mais frequentes ocorridos (EUA, China) para formar uma menor célula1 usando a codificação Huffman. Esta



técnica aumenta a eficiência da compressão e pode ser usada para avaliar ambas características de igualdade e intervalo. Como o processamento de consulta é feito em dados comprimidos, menos bits querem dizer maior velocidade.

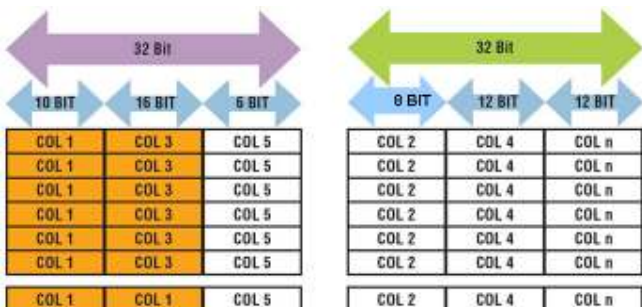
## Armazenamento Colunar no acelerador

Banco de dados de **linhas** tradicionais armazenam linhas (também conhecidas como tuplas) em uma página seguida por outra linha. O design otimiza a eficiência de IO, supõe que a consulta está interessada na maioria dos valores das colunas. Se os dados são comprimidos em armazenamentos, a linha é descomprimida em cada busca. Isto é eficiente para workloads transacionais acessando poucas linhas por consulta.

Em consultas analíticas, você acessa tipicamente milhões ou bilhões de linhas, mas cada consulta analisa o relacionamento entre as partes das colunas na tabela fact. Por exemplo, você pode estar interessado no total de vendas dos itens por localização em 2010. Esta consulta apenas precisa acessar três das colunas na tabela fact: item, quantidade de vendas e localização, e uní-los com dimensão. Neste caso, acessar e descomprimir cada linha não é eficiente.

Banco de dados **Colunares** armazenam cada valor das colunas juntos. Cada vez que você insere ou carrega os dados, cada linha é descomprimida para separar os valores das colunas e cada vez que você acessa a linha, os valores das colunas são buscados separadamente e então comprimidos para formar a linha. Porquê os os valores das colunas são armazenados juntos, você pode conseguir melhor compressão.

No exemplo acima, nós vimos que consultas analíticas estão tipicamente interessadas em uma parte das linhas. Para esta consulta, em banco de dados de armazenamento colunar, você pode apenas buscar as páginas que armazenam os itens, vendas e localização. Para as consultas acessando grandes partes das linhas e fazendo scans sequenciais, armazenamentos colunar melhora a eficiência.



O acelerador Informix Warehouse armazena dados em **grupos de colunas**, ou partições verticais da tabela, chamados “bancos”. A fração de uma linha (ou tupla) que encaixa em cada banco é chamada “**tuplet**”. A atribuição das colunas para os bancos é específico da célula, porque o comprimento das colunas variam de célula para célula.

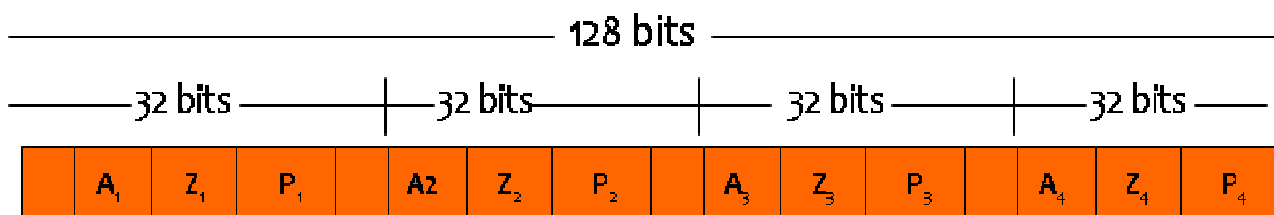
A atribuição utiliza um algoritmo binário de junção que tem por base se a coluna se encaixa em um banco, cuja largura é alguma fração de palavra, ao invés do uso em um workload. Também, o scan apenas necessita acessar o banco que contém colunas referenciadas em uma dada consulta, com exceção dos scans nos bancos com nenhuma referência na consulta. Esta projeção é parecida com a forma em que um simples armazenamento de coluna minimiza o I/O do disco. O acelerador armazena todos os dados na memória, para que não haja IO no disco. Embora o acelerador Informix Warehouse não tenha IO no disco, esta técnica minimiza a quantia de memória para scanear e economiza ciclos consideráveis de CPU.

## Única instrução e Paralelismo de Dados Múltiplos

Considere uma consulta:

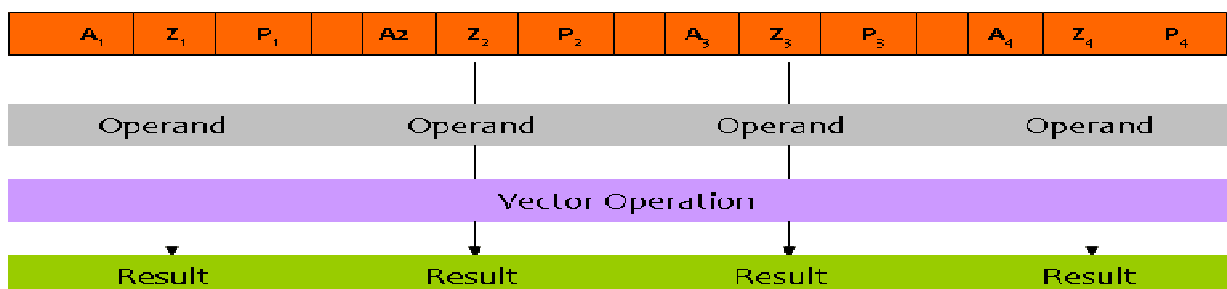
```
SELECT SUM(s.amount) FROM sales AS s WHERE s.prid = 100 GROUP BY s.zip;
```

Se a quantia da coluna (A), prid (P), zip(Z) são do mesmo banco, múltiplos desses valores podem ser carregados em um registrador de CPU de 128 ao mesmo tempo. Neste caso, nós temos 12 valores de colunas ao mesmo de cada vez.



Instruções SIMD nos processadores Intel Xeon operam em registradores de 128 bits. A técnica de descompressão no acelerador requer tipicamente poucos bits para cada coluna e portanto, pode carregar muitos campos em cada registrador de 128 bits. Nós podemos carregar valores múltiplos e aplicar predicados em todas as colunas simultaneamente. Durante o processamento da consulta, esta operação sobrecarregada está acontecendo em todos os núcleos alicados resultando em paralelismo extremo para cada consulta.

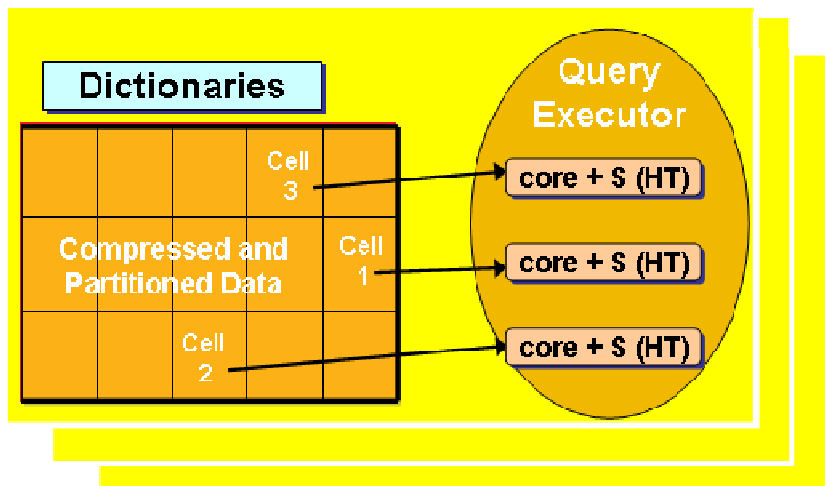
Uma vez que você tem isso, você pode operar em todos os 12 valores simultaneamente com uma única instrução de CPU resultando em ganho significativo de performance.



# Processamento de Consulta

Até agora, nós vimos como os dados são codificados e armazenados, e como o processamento é feito em nível micro. Vamos explorar o processamento de consulta em um nível macro.

**Scanning Eficiente** proporciona a fundação para o processamento de consultas. Nós sabemos que os dados são organizados em **tuplets**, bancos e células. Para scanear, célula é a unidade do processamento. O acelerador gera automaticamente threads apropriados para utilizarem recursos configurados de CPU e atribui células para cada núcleo. Este scan opera em dados não comprimidos, usando instruções SIMD, explorando as vantagens da codificação Huffman. Avaliação de atributos, GROUP BY é feito em dados comprimidos, mas agregação é feita em dados não comprimidos. A técnica de codificação também funciona como uma função hash muito densa permitindo o cache da tabela hash no cache do processador L2 em uma rápida visualização.



Isto permite um GROUP BY muito rápido nos dados comprimidos. Joins entre duas tabelas hash densamente codificadas pode resultar em valores espalhados. O acelerador vai detectar estas situações e mudar dinamicamente para um exame linear. A combinação destas técnicas permite o processamento destas consultas em dados comprimidos.

Nós sabemos que o data mart é uma estrela de schema de flocos de neve contendo uma grande tabela usando os predicados join de fatos a dimensões e então de dimensões a outras dimensões. Para cada consulta, cada worker cria um modelo de floco de neve para as tabelas. Então ele começa pela borda mais externa da ramificação e vai até a tabela fact. Cada ramificação de floco de neve é processada e seu resultado se comporta como uma entrada dimensional para o próximo nível. Primeiro os predicados locais são aplicados às tabelas dimensão para criar uma lista de chaves qualificadas. Estas chaves da tabela dimensão são unidas com a tabela fact (ou a tabela dimensão se comportando como uma fact na ramificação de floco de neve) para formar o próximo nível de agregações e relações. O processo é aplicado recursivamente até que a junção completa é processada em cada worker. Como cada worker possui cópia da tabela dimensão, há pouca troca de dados entre os workers durante o processamento da consulta em cada um prossegue com o máximo desempenho possível. O worker envia então, o conjunto de resultado intermediário para o coordenador.

Como há apenas uma representação dos dados – dados de tabela colunar comprimidos – o acelerador segue o mesmo caminho de código toda vez. Toda a eficiência das células, eliminação de bloqueio para a consulta vem da compressão de decodificação. Basicamente, o acelerador não tem indexes e tabelas sumário para se preocupar. Ele segue praticamente segue o mesmo processo toda vez. Por causa disso, o desempenho do acelerador também é consistente.

O nó coordenador pega o resultado de cada worker, mescla o resultado intermediário em grupos apropriados, descomprime os dados e então executa instruções ORDER BY e HAVING antes de enviar os dados para o servidor de dados Informix sob o protocolo DRDA. Então o servidor de dados Informix roteia os dados para o programa da aplicação.

## Conclusão

A abordagem inovativa do acelerador Informix Warehouse aborda processamentos complexos de consultas e melhora a produtividade para o seu negócio fornecendo respostas mais rápidas sem aumentar seu trabalho ou dividir seu orçamento. Por que é fortemente integrado com o servidor de banco de dados Informix, você pode explorar o ambiente para dividir o carregamento entre Informix e o acelerador se necessário.

Tempo de resposta mais rápido significa respostas mais rápidas, introspecções mais rápidas e um negócio que pode adaptar-se mais rápido. Você pode planejar para acelerar a parte de alto valor de seu armazém e desenvolver dinamicamente a infraestrutura para servir as necessidades de negócio.

## Próximo Passo

Para aprender mais sobre o acelerador Informix Warehouse e o Informix Ultimate Warehouse Edition, por favor contacte seu representante de marketing ou seu IBM Business Partner, ou visite os seguintes websites:

<http://www.ibm.com/informix>

<http://www.ibm.com/informix/warehouse>

## Reconhecimentos

Este produto foi desenvolvido por colaboração da IBM Almaden Research, o laboratório IBM Böblingen e o time IBM Informix. Obrigado ao time Informix por revisar e melhorar este artigo.

### Mais leituras

- **VLDB 2008:** “Main-Memory Scan Sharing for Multi-core CPUs”, Lin Qiao, Vijayshankar Raman, Frederick Reiss, Peter Haas, Guy Lohman
- **VLDB 2008:** “Row-Wise Parallel Predicate Evaluation”, Ryan Johnson, Vijayshankar Raman, Richard Sidle, Garret Swart
- **VLDB 2006:** “How to wring a table Dry: Entropy Compression of Relations and Querying Compressed Relations”, Vijayshankar Raman, Garret Swart
- **SIGMOD 2007:** “How to barter bits for chronons: compression and bandwidth trade offs for database scans”, Allison L. Holloway, Vijayshankar Raman, Garret Swart, David J. DeWitt
- **ICDE 2008:** “Constant-time Query Processing”, Vijayshankar Raman, Garret Swart, Lin Qiao, Frederick Reiss, Vijay Dialani, Donald Kossmann, Inderpal Narang, Richard Sidle
- **BTW 2009:** Bringing BLINK Closer to the Full Power of SQL. Knut Stolze, Vijayshankar Raman, Richard Sidle, Oliver Draese