Processamento Distribuído de Operações Hash Join em Switches Programáveis: Uma Análise via Modelo de Custo

Marisa S. Franco, Simone Dominico, Tiago R. Kepe,
Luiz C. P. Albini, Eduardo C. de Almeida, Marco A. Zanata Alves

Apoio:

1

Introdução

64,2 zettabytes

Volume de dados gerados globalmente em 2020

181 zettabytes

Estimativa do volume de dados gerados globalmente em 2025

Fonte: HOLST (2021)

Contextualização

- **Aumento da relevância** de pesquisas sobre **bancos** de dados **distribuídos**.

- O custo do processamento de uma consulta em sistemas de banco de dados distribuídos liga-se diretamente ao **custo da transferência de dados na rede**.

- Latência de rede é um **gargalo** para ganho de desempenho em consultas nas quais a comunicação entre servidores é exigida. Ex.: **operações de hash join**.

Contextualização

- A Software-Defined Wide Area Network (**SD-WAN**) é uma tecnologia que permite **(re)programar dispositivos de rede** via software.

- Sua **programabilidade** da rede traz novas possibilidades para gerenciar topologias de forma dinâmica e **mitigar latências**, possibilitando ainda o **processamento** de dados em **dispositivos de rede**.

Objetivo

- **Avaliar o processamento distribuído da operação de hash join em dispositivos de rede, via modelo de custo.**

- A análise contempla **seis cenários** de execução de hash join distribuído, sendo dois deles com processamento da consulta em um **switch**.

- Assim, buscou-se avaliar o **potencial** do processamento de dados em dispositivos de rede para **ganho de desempenho**.

2

Bancos de Dados Distribuídos e Operação Hash Join

Bancos de Dados Distribuídos

- Um sistema de gerência de banco de dados distribuídos gerencia uma **coleção de banco de dados** espalhada em **diversos servidores** conectados via **rede** de computadores.

- **Servidor** ou nó armazena e processa dados. Arquitetura geral = **“modelo cliente-servidor”**.

- Processamento colaborativo e transferência de dados via rede dedicada para suportar alto fluxo de dados permutados entre dispositivos de rede, como **switches** (agentes **passivos**).

Custo das consultas

Diretamente ligado ao custo da transferência de dados na rede (KOSSMANN, 2000).

Assim, busca-se gerar **planos de consulta** que **minimizem** a quantidade de **dados trafegados**.

Join e Hash Join

- Operação de **join** em uma consulta: modo de recuperar dados de várias tabelas de banco de dados relacionais por combinações lógicas entre as tabelas. Em um ambiente distribuído, podem existir **várias abordagens** de join.

- A **cardinalidade** é um dos fatores levados em consideração pelo otimizador para a escolha do algoritmo de junção (hash, merge, NL...).

- Optou-se por focar nas operações de **hash join** visando entender o impacto de tabelas de **alta cardinalidade** que cabem em memória.

Hash Join

- Hash join consta como **um dos quatro operadores** que mais contribuem para o **tempo de execução** e **memória utilizada** do benchmark TPC-H em bancos de dados colunares (KEPE et al., 2019).

- Algoritmo básico de **hash join**:

```
build hash table HT_R for
R
foreach tuple s ∈ S
    output, if h_1(s) ∈
HT_R
```

3

Metodologia e Modelagem Teórica

Metodologia

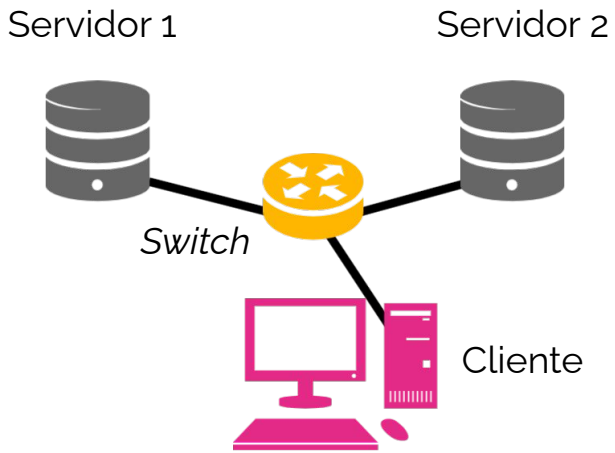
- Simplificação das consultas **query-10** e **query-11** do benchmark **TPC-H** (COUNCIL, 2020), incluindo apenas a operação de join, implementadas em C.

```
query 10:
SELECT C_CUSTKEY, C_NAME
FROM CUSTOMER, ORDERS
WHERE C_CUSTKEY = O_CUSTKEY
```

```
query 11:
SELECT PS_PARTKEY
FROM PARTSUPP, SUPPLIER
WHERE PS_SUPPKEY = S_SUPPKEY
```

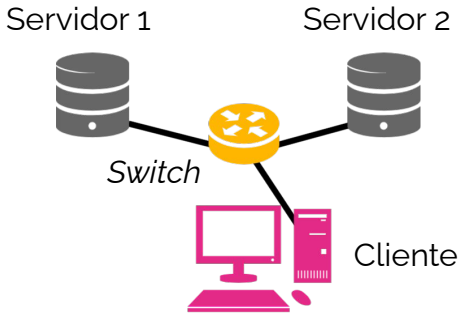
Metodologia

- Adoção de **seis cenários** de transmissão de dados baseados na seguinte **topologia estrela**:



Cenários analisados de processamento distribuído de hash join.

Cenário	(A) Armazenamento inicial, (C) Comunicação e (P) Processamento
1	**A:** Servidor 1 possui a relação maior. Servidor 2 possui a relação menor. **C:** Relação **maior** é transmitida. **P:** Processamento no servidor 2.
2	**A:** Servidor 1 possui a relação maior. Servidor 2 possui a relação menor. **C:** Relação **menor** é transmitida. **P:** Processamento no servidor 1.
3	**A:** Servidor 1 possui a relação maior. Servidor 2 possui a relação menor. **C: Ambas** as relações são transmitidas ao switch. **P: Processamento no switch.**
4	**A:** Cada servidor possui uma metade das duas relações. **P:** Cada servidor processa a fase de construção usando a relação menor. **C:** Compartilhamento das **tabelas hash** entre servidores. **P:** Cada servidor faz parte da análise, usando as duas hashes e sua metade da relação maior.
5	**A:** Cada servidor possui uma metade das duas relações. **C: Todas** são transmitidas ao switch. **P: Processamento no switch.**
6	**A:** Cada servidor possui uma metade das duas relações e dados **relevantes co-localizados**. **P:** Processamento local nos servidores de cada uma das metades.



Metodologia

- Cada um dos cenários foi avaliado com três cargas de trabalho (**1 GB**, **10 GB** e **100 GB**) e usando **quatro tecnologias de rede** (Ethernet 100 Gb, Ethernet 200 Gb, Ethernet 400 Gb e InfiniBand HDR 12X), com comunicação **full-duplex**.

- Nos **cenários 3 e 5**, em que há processamento no roteador, foram avaliadas as seguintes **velocidades de processamento**: **igual às CPUs** dos servidores, **5% e 10% mais lentas** do que as CPUs.

Metodologia

- Os dados foram armazenados no **formato DSM** (Decomposition Storage Model).
- Funções hash avaliadas: **FNV-1a** (FOWLER et al., 2013) e **MurmurHash3** (APPLEBY, 2016).
- Para cada um dos 24 experimentos-base foram realizadas **20 repetições**, totalizando 480 execuções.

Modelo de custo

- O **modelo de custo** proposto neste trabalho é **inspirado** na máquina teórica e **modelo LogP**, de Culler et al. (1993) – um modelo tradicional em programação paralela.

- Quanto à nomenclatura usada nas fórmulas:
  - **PD:** tempo de **processamento** dos dados;
    - Cenários 1, 2, 3 e 5, análise e construção sequenciais. Cenários 4 e 6, análise e construção paralelas;
  - **N:** tamanho da **relação maior**;
  - **n:** tamanho da **relação menor**;
  - **H:** tamanho da **tabela hash** – espalhamento perfeito;
  - **R:** tamanho da **relação de saída**;
  - **BW: largura de banda**.

Obs: "tamanho" refere-se ao número de tuplas da tabela multiplicado pela soma dos tamanhos das colunas pertinentes à consulta.

Cálculo de custo utilizado em cada cenário.

Cenário	Cálculo de tempo total
1	$PD + \left(\frac{(N+R)}{BW}\right)$
2	$PD + \left(\frac{(n+R)}{BW}\right)$
3	$PD + \left(\frac{(max(N;n)+R)}{BW}\right)$
4	$PD + \left(\frac{((H/2)+R)}{BW}\right)$
5	$PD + \left(\frac{((N/2)+(n/2)+R)}{BW}\right)$
6	$PD + \left(\frac{R}{BW}\right)$

PD: tempo de **processamento** dos dados;
Cenários 1, 2, 3 e 5, análise e construção sequenciais.
Cenários 4 e 6, análise e construção paralelas;
N: tamanho da **relação maior**;
n: tamanho da **relação menor**;
H: tamanho da **tabela hash** – espalhamento perfeito;
R: tamanho da **relação de saída**;
BW: Largura de banda.

Obs: "tamanho" refere-se ao número de tuplas da tabela multiplicado pela soma dos tamanhos das colunas pertinentes à consulta.

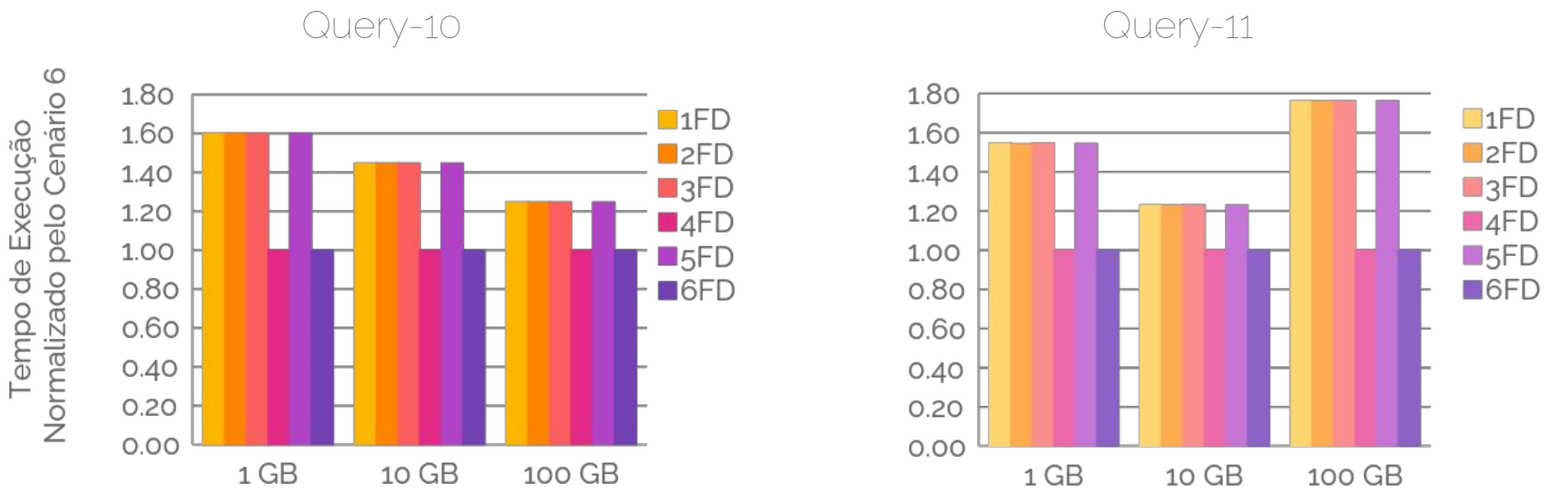
Limitações do modelo

- Consideraram-se apenas as **capacidades máximas** teóricas da **largura de banda** de cada tecnologia.
- Trabalhou-se com o envio do **volume bruto** de dados, sem discriminar cabeçalhos e divisão das mensagens em pacotes.
- **Não** foram consideradas as **latências de rede** e as **distâncias** entre os nós. Nas estimativas feitas, a latência inicial de comunicação é negligenciável (poucos envios com grandes volumes de dados).
- Considerou-se o **espalhamento perfeito** dos dados nas tabelas **hash** para mensurar o volume de dados transferido no cenário 4.

4

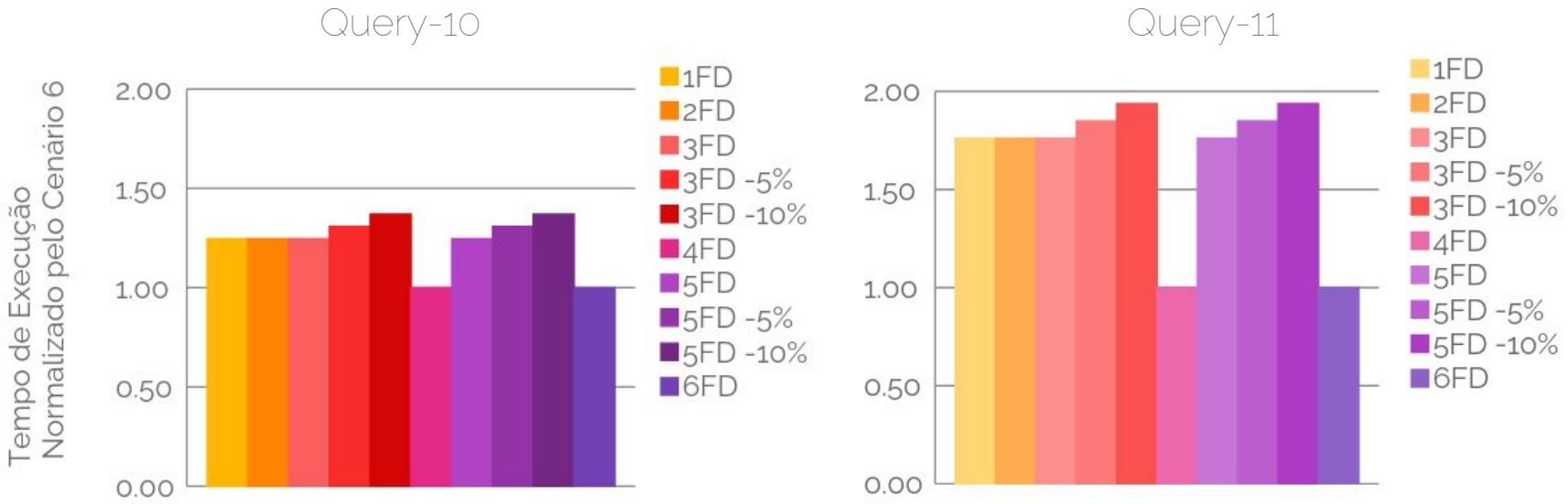
Análise dos Resultados

Análise dos Resultados



Tempos conforme variação da carga de trabalho, usando InfiniBand HDR com switch CPU 100% e função MurmurHash3. Valores normalizados pelo cenário 6.

Análise dos Resultados



Tempos conforme variação na capacidade de processamento do switch, usando InfiniBand HDR com carga de trabalho de 100 GB e função MurmurHash3. Valores normalizados pelo cenário 6.

Análise dos Resultados

Porcentagem do tempo de cada etapa para a query-10, considerando o pior cenário (1) e melhor cenário (6), com Ethernet 100 Gb.

Cenário	Consulta	Carga de trabalho (GB)	Construção (%)	Análise (%)	Transferência (%)
1	10	1	3,71	95,49	0,80
6	10	1	2,57	96,28	1,16
1	10	10	5,88	93,88	0,24
6	10	10	1,99	97,70	0,31
1	10	100	7,36	92,51	0,13
6	10	100	4,73	95,11	0,15

Análise dos Resultados

Porcentagem do tempo de cada etapa para a query-11, considerando o pior cenário (1) e melhor cenário (6), com Ethernet 100 Gb.

Cenário	Consulta	Carga de trabalho (GB)	Construção (%)	Análise (%)	Transferência (%)
1	11	1	2,00	95,94	2,06
6	11	1	1,55	97,38	1,07
1	11	10	1,75	96,92	1,33
6	11	10	0,97	98,48	0,55
1	11	100	1,49	98,37	0,14
6	11	100	0,43	99,49	0,08

O modelo de processamento de dados em switches apresenta performance equiparável com processamento tradicional em servidores com tráfego de dados similar.

Os dois melhores tempos de execução, entre os seis cenários apresentados, foram obtidos por cenários em que há menos transferência de dados e o processamento acontece em paralelo nos servidores locais.

Em todos os experimentos, o cenário 6 apresentou o melhor desempenho em relação ao tempo de execução.

5

Conclusões

Conclusões

- Variações básicas de distribuição dos dados e processamento foram suficientes para diferenciar os desempenhos e indicar melhores cenários a serem explorados para avançar na pesquisa.

- As velocidades de transmissão das tecnologias de rede não crescem na mesma proporção que o volume de dados transferidos anualmente. Logo, a latência na transferência de dados segue um gargalo importante em sistemas distribuídos.

- Porém, vale ressaltar que o modelo e os resultados apresentados demonstram que o **paralelismo** no processamento é um **ponto crítico** extremamente relevante para o ganho de **desempenho em tempo de execução** nas consultas que envolvem processamento **distribuído** da operação de **hash join**.

Conclusões

- Quanto aos riscos e desvantagens de se usar redes programáveis para o processamento distribuído de operações hash join, o ambiente de rede considerado foi o privado. Mas questões ligadas à tolerância falhas e limitações de memória nesses sistemas devem ser discutidas em trabalhos futuros.

- Processamentos de **consultas recorrentes** podem **se beneficiar** do processamento em **redes programáveis**.

- Espera-se contribuir para um conhecimento mais amplo sobre diferentes estratégias de processamento distribuído de dados e o potencial do processamento de dados em dispositivos de rede para ganho de desempenho, motivando novos estudos.

Agradecimentos

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

HiPES
High Performance and Efficient Systems

serrapilheira

CAPES

CNPq
Conselho Nacional de Desenvolvimento
Científico e Tecnológico

Dúvidas, comentários, críticas?






{masf18, sdominico, trkepe, albini, eduardo, mazalves}@inf.ufpr.br

Obrigada ;)

Créditos: template por SlidesCarnival.