Cheap Data Analytics using Cold Storage Devices

Renata Borovica-Gajic, Raja Appuswamy, and Anastasia Ailamaki

renata.borovica@epfl.ch, raja.appuswamy@epfl.ch, anastasia.ailamaki@epfl.ch

Cold Data and Cold Storage

Proliferation of cold data

"**Enterprise data** is growing at a rate of **40% to 60% per year** and is projected to grow 50-fold – from under one zettabyte in 2010 to 40 zettabytes by 2020. " [GigaOM]

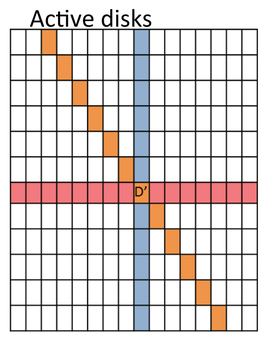
"**Archival data** presently represents approximately 43-60% of all data stored online, making it the **largest category** and at **> 60% CAGR** is the fastest growing data classification segment. " [Horison]

"Although **cold data** is infrequently accessed, it is still **incredibly valuable**. Businesses are increasingly investing in "big data" analytics to identify customer and operational trends, and to gain business insights. Cold storage must therefore provide the performance and capabilities required to **enable analysis.** " [Intel]

Cold Storage Devices (CSD)

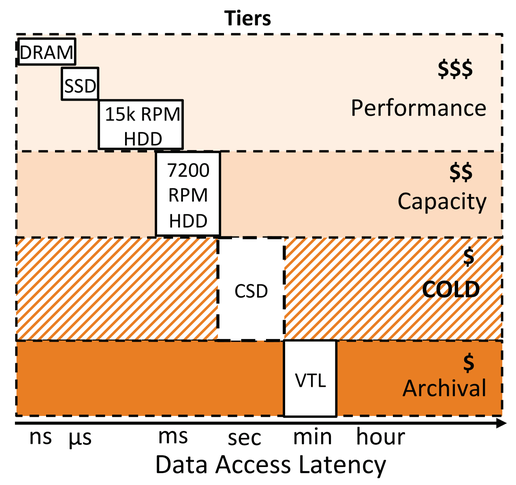
[Wiwynn]

PB-size rack based on **High density HDD** organized in **MAID**



Constraints on number of active disks
Group switch latency **~10sec**

CSD in storage tiering hierarchy

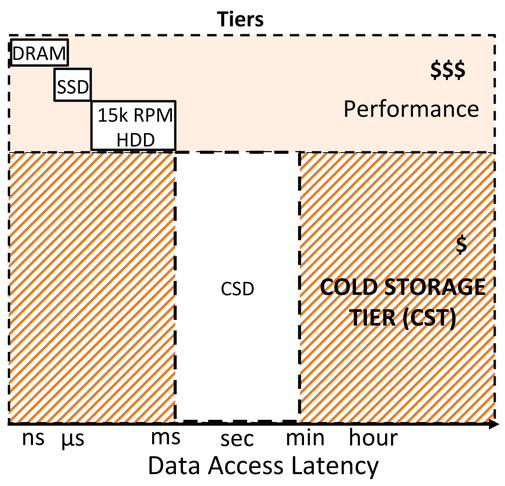


Cost ~ tape, latency ~ disks (ms – secs)

Fast for its price, cheap for its speed

A case for Cold Storage Tier

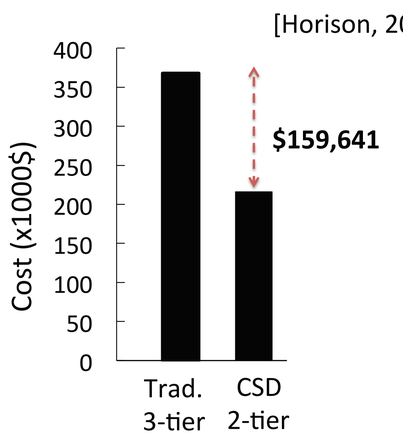
Shrinking storage hierarchy



How far can we go without sacrificing performance?

Price benefit of CST

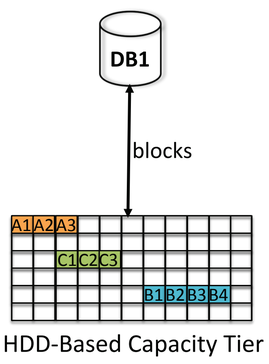
Setting: 3-tier (performance, capacity, archival) vs 2-tier(performance, CST), 100TB



Substantial cost savings (40%)

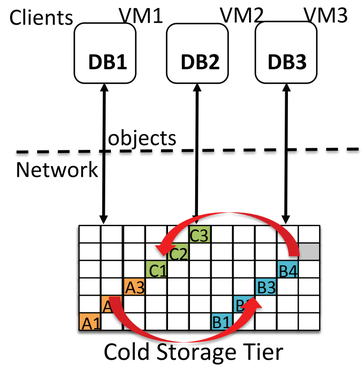
Query execution over CSD

Traditional setting



Control layout ✓ Uniform access ✓
Static (pull-based) execution ✓

Virtualized enterprise data center

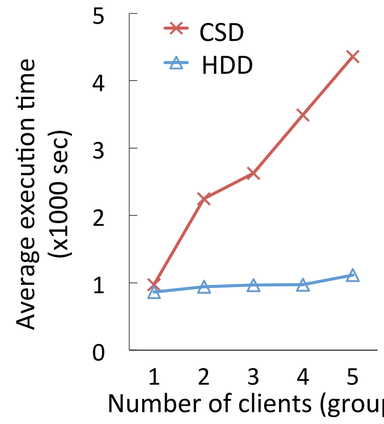


Control layout ✗ Uniform access ✗
Static (pull-based) execution ✗

Traditional execution incompatible with CSD

Performance implications of CST

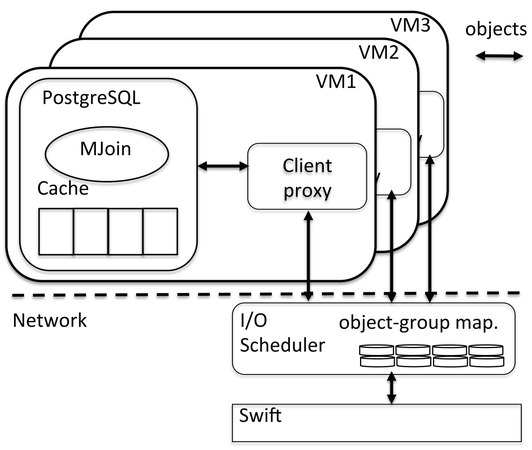
Setting: virtualized enterprise datacenter, clients: PostgreSQL TPCH 50, Q12, CSD: shared, layout: one cl. per group



Performance drop due to group switch latency

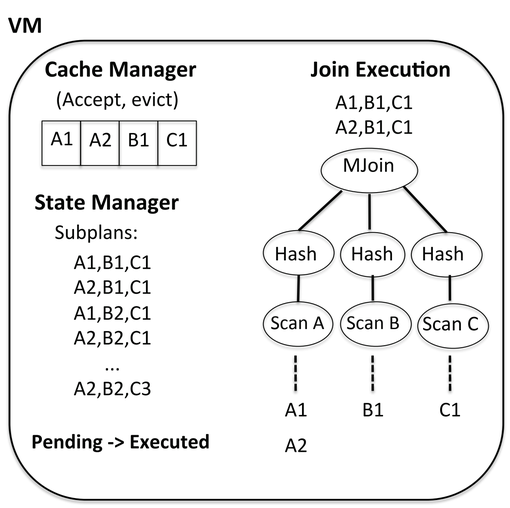
Skipper in a Nutshell

Skipper Architecture



CSD-driven execution combined with smart caching and I/O scheduling

Multi-way joins in PostgreSQL



Out-of-order opportunistic execution

Rank-based scheduling

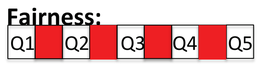
Which group to switch to ?

Group	Queries
G1	Q1, Q3, **Q6,Q8**
G2	Q2, Q4, **Q7,Q9**
G3	Q5

#Q1:2
#Q2:2
#Q3:1

TIME

Fairness:



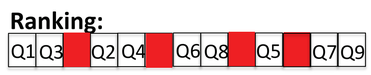
Efficiency:

Group	Queries
G1	Q1, Q3, **Q6,Q8**
G2	Q2, Q4, **Q7,Q9**
G3	Q5

RANKING= #Q + R

provides efficiency — provides fairness

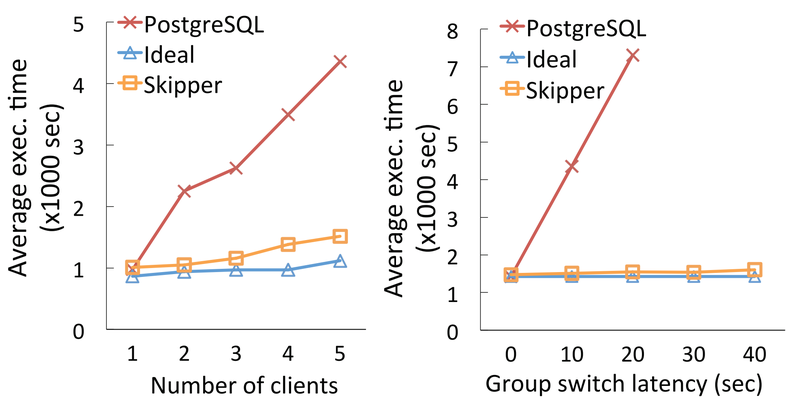
Ranking:



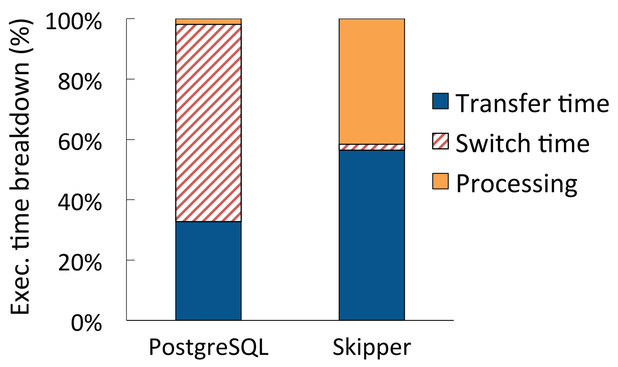
Balances efficiency and fairness

Skipper in Action

Setting: multitenant enterprise datacenter, clients 1- 5: TPCH 50, Q12, CSD: shared, layout: one client per group



Skipper approximates capacity tier by 20% avg.
Skipper is resilient to group switch latency



Skipper substantially reduces overhead of group switches

Summary

- Cold storage can substantially reduce TCO
  - But DBMS performance suffers due to pull-based execution
- Skipper enables efficient query execution over CSD
  - Out-of-order execution based on multi-way joins
  - Novel progress based caching policy
  - Rank based I/O scheduling
- Skipper enables data analytics over CSD as a service
  - Providers reduce cost by offloading data to CSD
  - Customers reduce cost by running inexpensive data analytics over CSD

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

DATA-INTENSIVE APPLICATIONS AND SYSTEMS