CIDR 2023, Amsterdam, The Netherleands

Data Processing in the Hardware Era

Gustavo Alonso

Systems Group

Department of Computer Science

ETH Zurich, Switzerland

The many waves in database evolution

YESTERDAY

Pre-relational

The relational years

The object oriented years

The distribution years

The data mining years

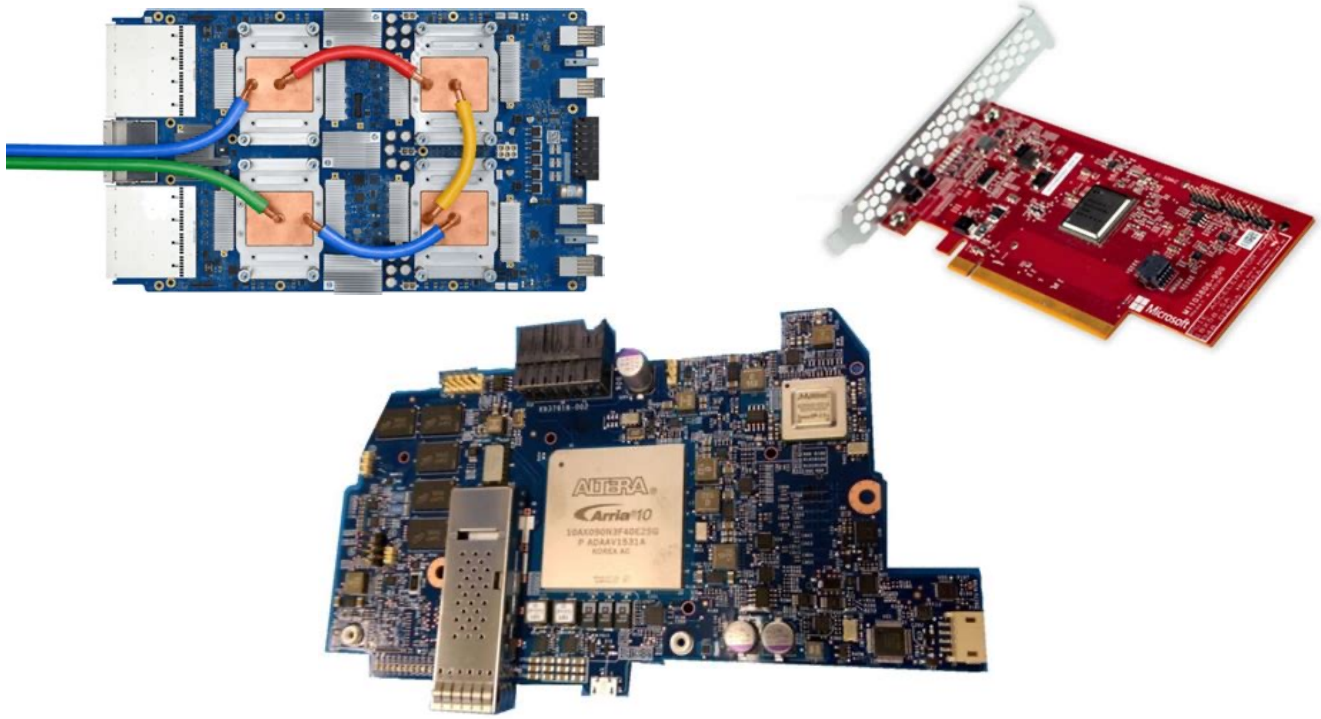
The web search years

The cloud years

…

TODAY

The hardware years



With every wave, we went into crisis mode

MapReduce: Simplified Data Processing on Large Clusters

CACM January 2008

by Jeffrey Dean and Sanjay Ghemawat

Abstract

MapReduce is a programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks. Users specify the computation in terms of a *map* and a *reduce* function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks. Programmers find the system easy to use: more than ten thousand distinct MapReduce programs have been implemented internally at Google over the past four years, and an average of one hundred thousand MapReduce jobs are executed on Google's clusters every day, processing a total of more than twenty petabytes of data per day.

First, we say it sucks …

MapReduce: A major step backwards

By David DeWitt on January 17, 2008 4:20 PM | Permalink | Comments (44) | TrackBacks (1)
[Note: Although the system attributes this post to a single author, it was written by David J. DeWitt and Michael Stonebraker]

1. A giant step backward in the programming paradigm for large-scale data intensive applications
2. A sub-optimal implementation, in that it uses brute force instead of indexing
3. Not novel at all -- it represents a specific implementation of well known techniques developed nearly 25 years ago
4. Missing most of the features that are routinely included in current DBMS
5. Incompatible with all of the tools DBMS users have come to depend on

Then we tell them what they have to do …

A Comparison of Approaches to Large-Scale Data Analysis

Andrew Pavlo
Brown University
pavlo@cs.brown.edu

Erik Paulson
University of Wisconsin
epaulson@cs.wisc.edu

Alexander Rasin
Brown University
alexr@cs.brown.edu

Daniel J. Abadi
Yale University
dna@cs.yale.edu

David J. DeWitt
Microsoft Inc.
dewitt@microsoft.com

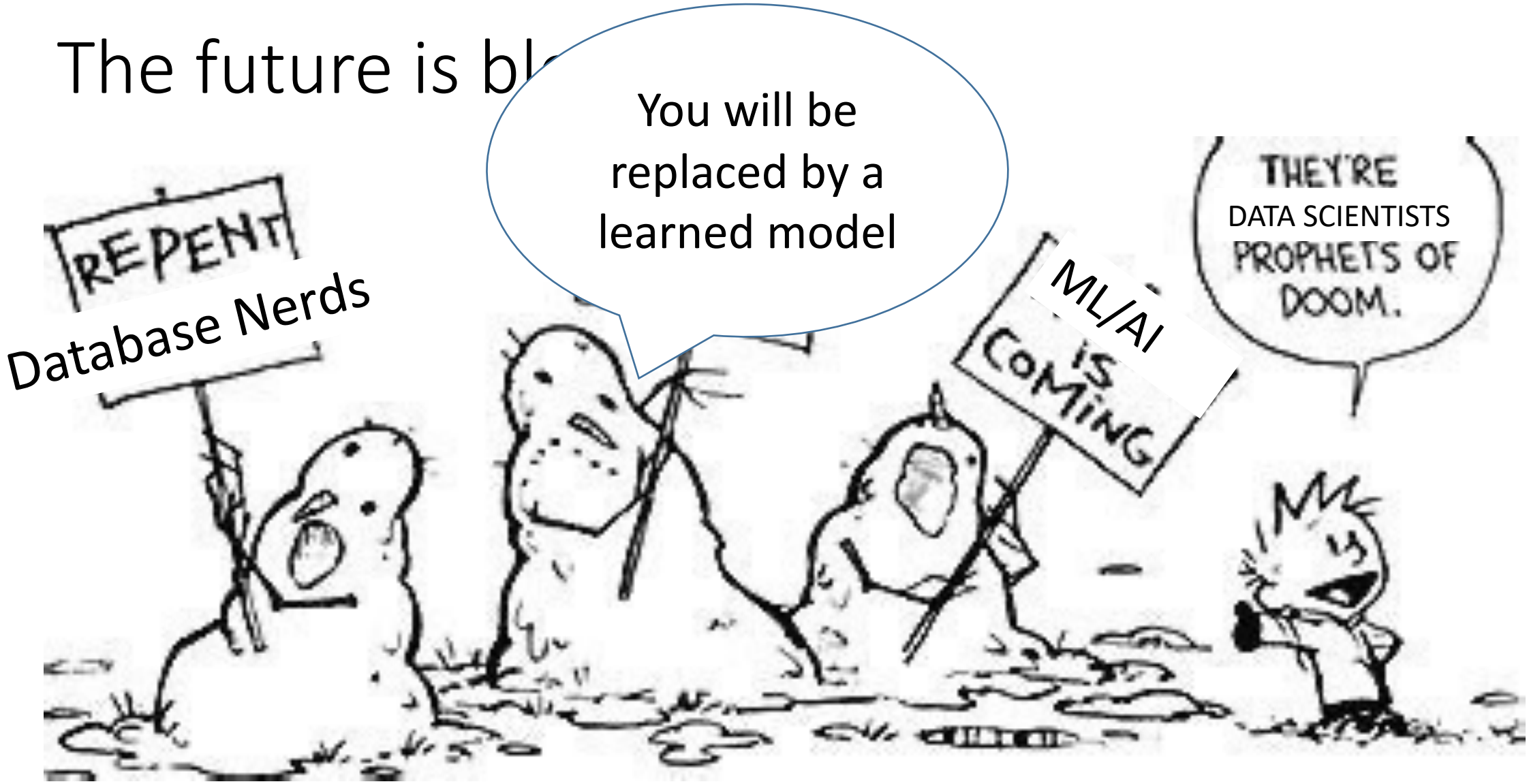
Samuel Madden
M.I.T. CSAIL
madden@csail.mit.edu

Michael Stonebraker
M.I.T. CSAIL
stonebraker@csail.mit.edu

SIGMOD 2009

The world quietly ignores us …

The future is bl



The real crisis today

The Netherlands has basically reinvented the tomato

By **Netherlands Foreign Investment Agency** | Published February 17, 2017

The price of performance

Out of flavour: why tomatoes have lost their taste

After exhaustive studies, an international team of scientists has worked out why tomatoes don't taste like they used to

The scientists found that tomato flavour was inadvertently lost as the industry sought to maximise yields. Photograph: Steven Senne/AP

The Guardian, January, 2017

TIME

SCIENCE

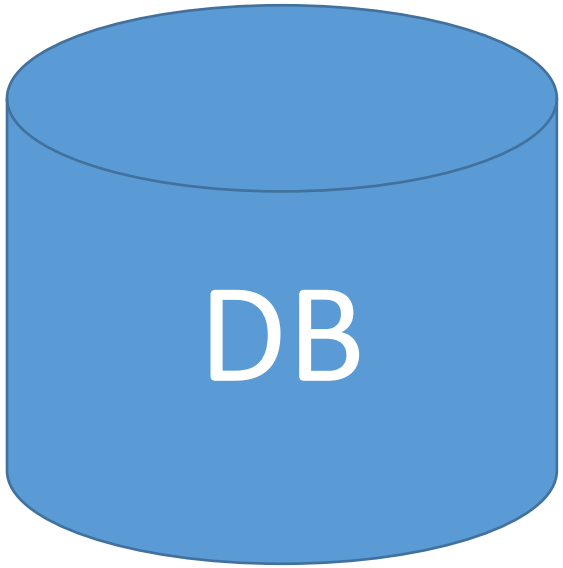
Your Tomatoes Are Flavorless, Right? Here's Why

By Jeffrey Kluger | Monday, July 02, 2012

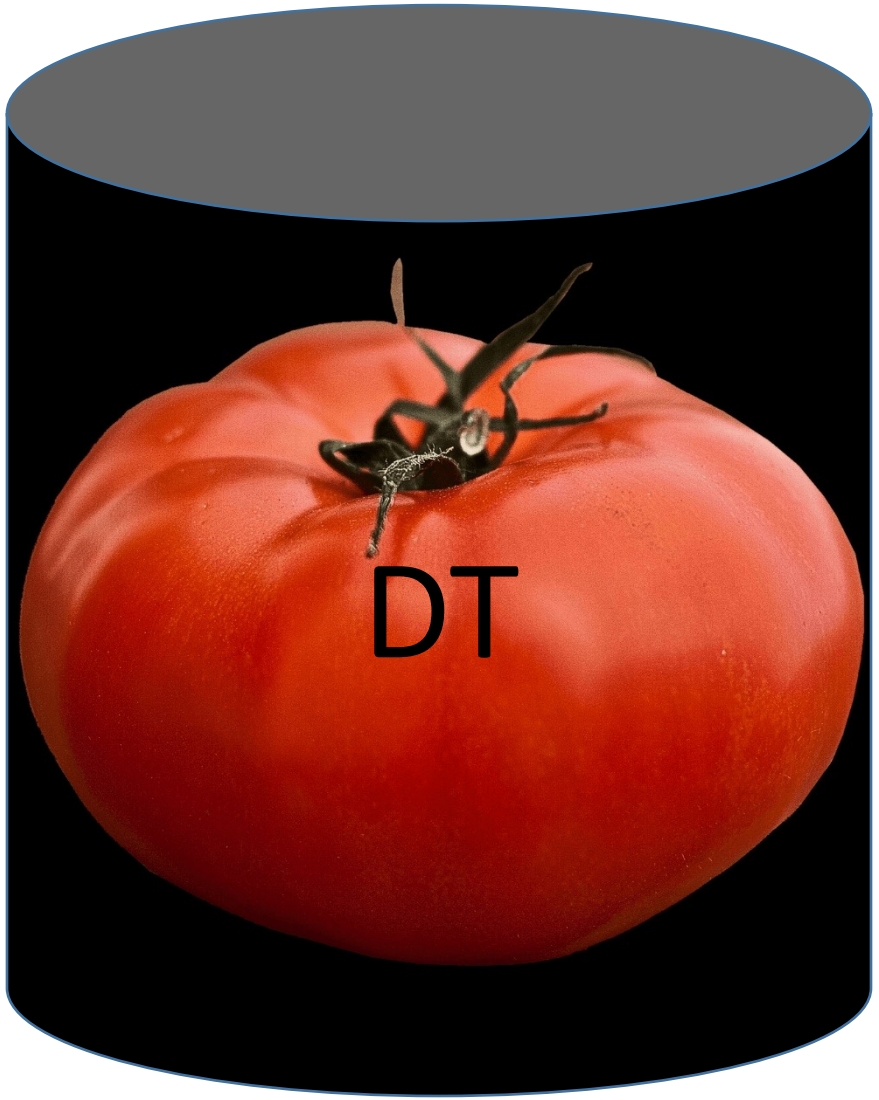
Tweet Read Later

There are two pieces of late-breaking news on the tomato beat this week. First of all, tomatoes have shoulders. Second, tomatoes taste lousy. If you're younger than 70, you probably already know about the lousy part. The shoulders are surely more of a surprise — but these are both key parts of a new study published in *Science* that explains what's going on in the sorry world of supermarket tomatoes and why they taste nothing like their sweet, flavorful cousins in the wild.

DBs are the new DTs



=



Don’t keep polishing the Dutch Tomato

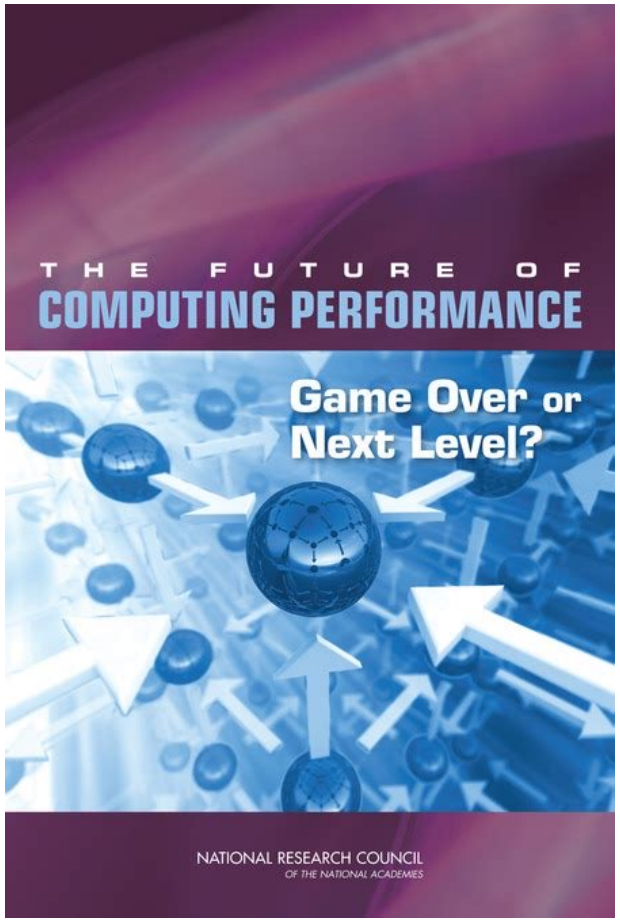
- Can we push query response times into the few microseconds range?
  - Make them compatible with the rest of the infrastructure
- Can we make databases truly elastic rather than server based?
  - How fast can we get a database up and running?
  - How fast can we migrate a database engine from one machine to another?
- Can we make database architecture dynamic?
  - Deploy only what is needed, pay only for what is used
- Can a database benefit from progress elsewhere?
  - TPUs, GPUs, FPGAs, smart NICs, ASICS …
- Beyond SQL
  - Can we move away from single pass operators?

Despair not: the best of times for research!

We are in an era where most of the established assumptions, rules of thumb, and accumulated wisdom about data processing no longer hold and need to be revisited.

The Hardware Era

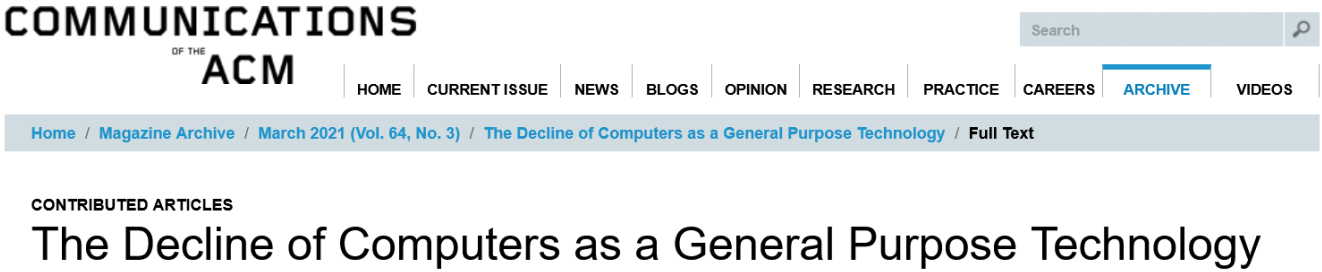
Not a new concept …

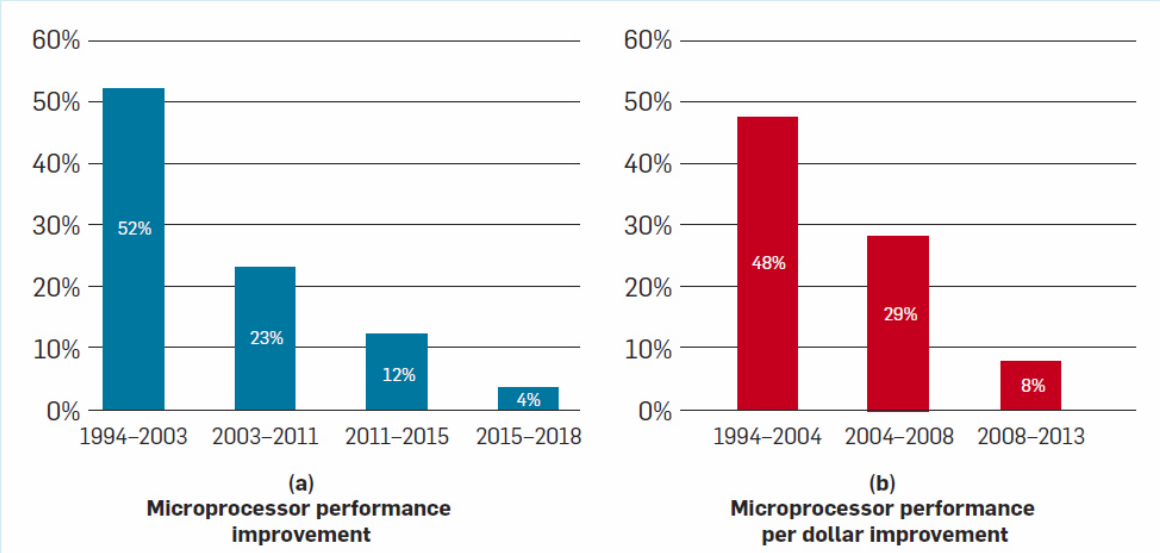


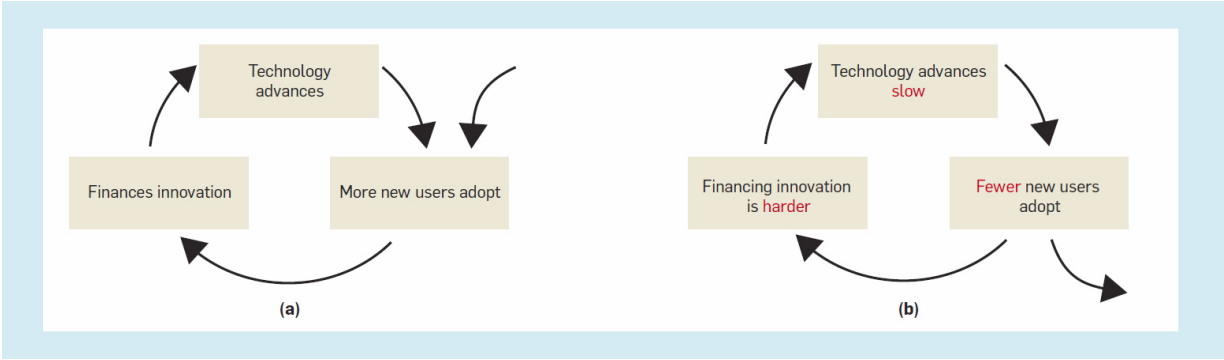
- 2011 Report
- Exponential growth for several decades
- Exponential growth no longer possible
- Switch to multicore and parallelism
  - Energy consumption becomes an issue
  - Multicore introduces parallelism that we do not know how to exploit well
- Situation will not change in near future
- Alternative is specialization
- Either somebody comes up with a new great invention or there is a problem

General purpose computing

Slow improvements lead to specialization

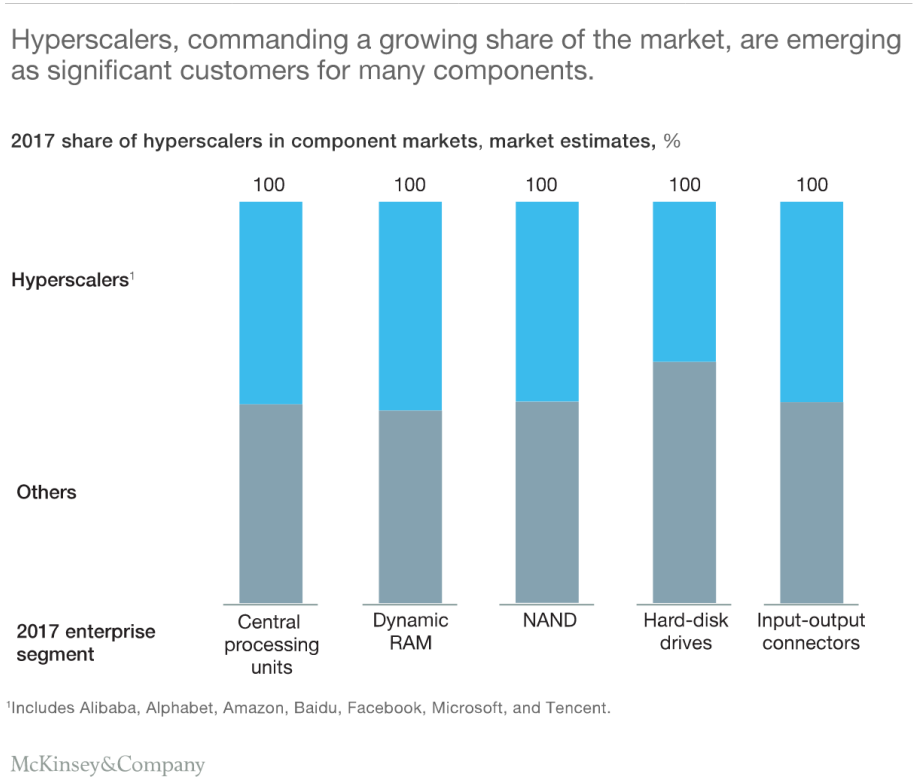






Driving specialization

- The cloud is the big game changer:
  - New business model
  - Economies of scale
  - Very large workloads
- Every hyper scaler is its own “Killer App”
  - The scale makes many things feasible
  - The gains have a very large multiplier



https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/how-high-tech-suppliers-are-responding-to-the-hyperscaler-opportunity

HW Specialization for databases

Have we been here before?

The Gamma Database Machine Project

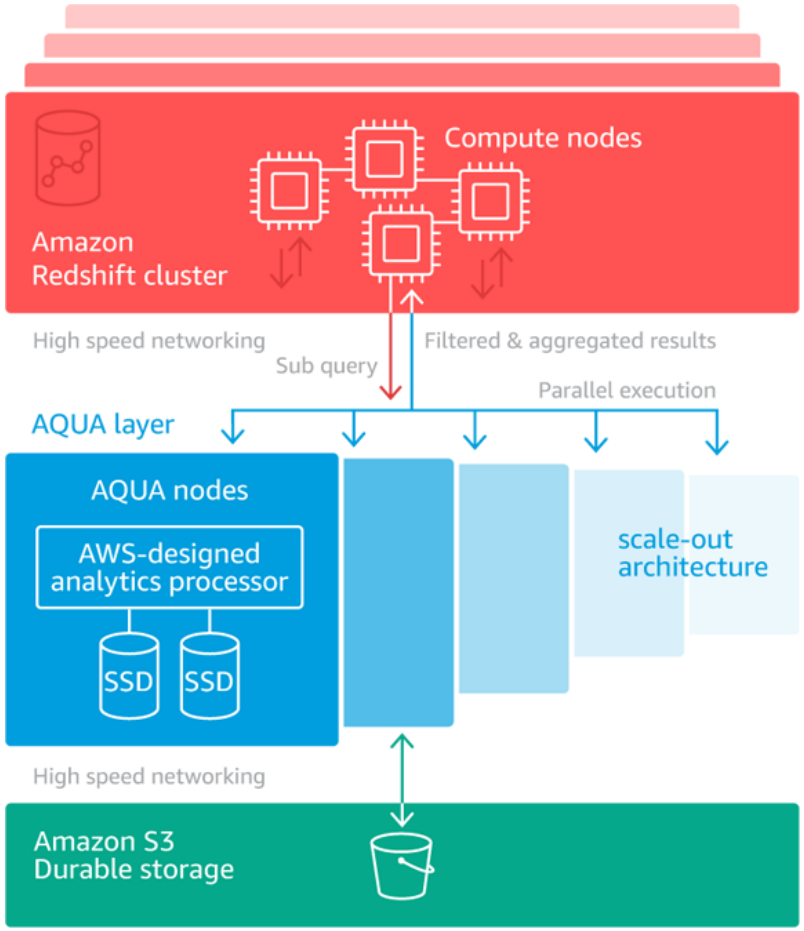
DAVID J. DEWITT, SHAHRAM GHANDEHARIZADEH, DONOVAN A. SCHNEIDER, ALLAN BRICKER, HUI-I HSIAO, AND RICK RASMUSSEN

Abstract—**This paper describes the design of the Gamma database machine and the techniques employed in its implementation. Gamma is a relational database machine currently operating on an Intel iPSC/2 hypercube with 32 processors and 32 disk drives. Gamma employs three key technical ideas which enable the architecture to be scaled to hundreds of processors. First, all relations are horizontally partitioned across multiple disk drives enabling relations to be scanned in parallel. Second, novel parallel algorithms based on hashing are used to implement the complex relational operators such as join and aggregate functions. Third, dataflow scheduling techniques are used to coordinate multioperator queries. By using these techniques it is possible to control the execution of very complex queries with minimal coordination—a necessity for configurations involving a very large number of processors.**

shared memory and centralized control for the execution of its parallel algorithms [3].

As a solution to the problems encountered with DIRECT, Gamma employs what appear today to be relatively straightforward solutions. Architecturally, Gamma is based on a shared-nothing [37] architecture consisting of a number of processors interconnected by a communications network such as a hypercube or a ring, with disks directly connected to the individual processors. It is generally accepted that such architectures can be scaled to incorporate thousands of processors. In fact, Teradata database machines [40] incorporating a shared-nothing ar-

Cloud caches (Amazon Aqua)



"AQUA is designed to deliver up to 10X performance on queries that perform large scans, aggregates, and filtering with LIKE and SIMILAR_TO predicates. Over time we expect to add support for additional queries."

https://aws.amazon.com/blogs/aws/new-aqua-advanced-query-accelerator-for-amazon-redshift/

Lesson Learned

- Database engines are bad at operations that have now become very important
- This is not because they are bad at it but because the underlying hardware (CPU) makes it expensive
- Other devices enable these operations, let's embrace them …

Accelerating Pattern Matching Queries in Hybrid CPU-FPGA Architectures

David Sidler Zsolt István Muhsen Owaida Gustavo Alonso
Systems Group, Dept. of Computer Science
ETH Zürich, Switzerland
{firstname.lastname}@inf.ethz.ch

SIGMOD 2017

X-Engine (Alibaba)

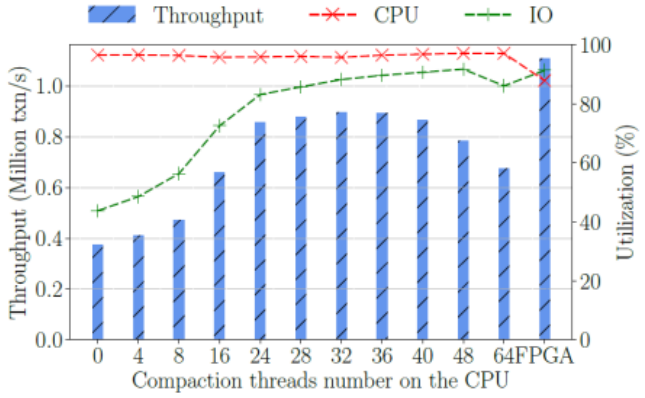
FPGA-Accelerated Compactions for LSM-based Key-Value Store

Teng Zhang[*,†], Jianying Wang[*], Xuntao Cheng[*], Hao Xu[*], Nanlong Yu[†], Gui Huang[*], Tieying Zhang[*], Dengcheng He[*], Feifei Li[*], Wei Cao[*], Zhongdong Huang[†], and Jianling Sun[†]

[*]Alibaba Group
[†]Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Zhejiang University
{jason.zt,beilou.wjy,xuntao.cxt,haoke.xh,qushan, tieying.zhang, dengcheng.hedc,lifeifei,mingsong.cw}@alibaba-inc.com
{yunanlong,hzd, sunjl}@zju.edu.cn

Abstract

Log-Structured Merge Tree (LSM-tree) key-value (KV) stores have been widely deployed in the industry due to its high write efficiency and low costs as a tiered storage. To maintain such advantages, LSM-tree relies on a background compaction operation to merge data records or collect garbages for housekeeping purposes. In this work, we identify that slow compactions jeopardize the system performance due to unchecked oversized levels in the LSM-tree, and resource contentions for the CPU and the I/O. We further find that the rising I/O capabilities of the latest disk storage have pushed compactions to be bounded by CPUs when merging short KVs. This causes both query/transaction processing and back



18th USENIX Conference on File and Storage Technologies (FAST'20)

Lesson Learned

- Even modern designs of database engines consume too many resources for operations internal to the engine
- Some of these operations are better done somewhere else than in a CPU
- Accelerators are available in data centers and the cloud, let’s embrace them …

Data Compression (Microsoft Zipline/Corsica)

https://azure.microsoft.com/en-us/blog/improved-cloud-service-performance-through-asic-acceleration/

Lesson Learned

Everything that is demanding enough and common enough will move to dedicated accelerators

Performance is not the only story …

- These deployments have one goal:
  - Free up the CPU for other tasks!

- In traditional database engines and data processing systems, the CPU does everything
  - No longer efficient!

First baby steps …

- Software evolves slower than hardware
- Current deployments focus on elements that can be migrated to hardware without major changes to the engines
- But once the hardware is available …
  - Engines will be developed for that hardware
  - Pressure will increase to take advantage of heterogeneous architectures
  - That will be the next wave of truly cloud native database engines

Emerging themes

- Reduced CPU utilization
- Accelerate common operations
- Accelerate the infrastructure supporting the system
- Processing data on the fly
- Near data processing (memory, storage, …)
- On demand servers and functionality
- …

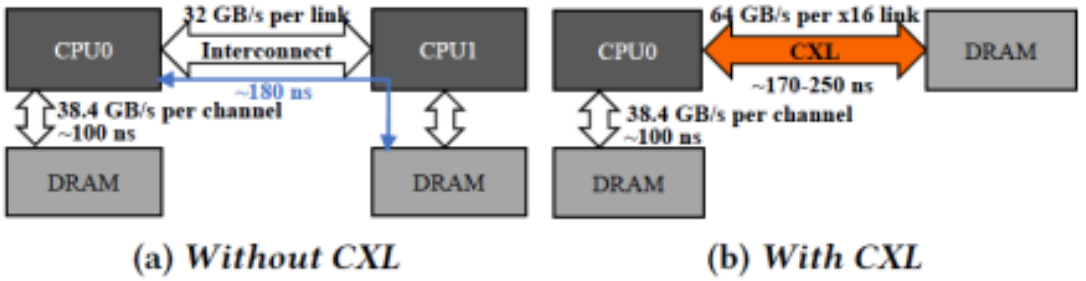
Ignore hardware developments at your own peril

The future of accelerators

TPP: Transparent Page Placement for CXL-Enabled Tiered Memory

Hasan Al Maruf*, Hao Wang†, Abhishek Dhanotia†, Johannes Weiner†, Niket Agarwal†, Pallab Bhattacharya†, Chris Petersen†, Mosharaf Chowdhury*, Shobhit Kanaujia†, Prakash Chauhan†

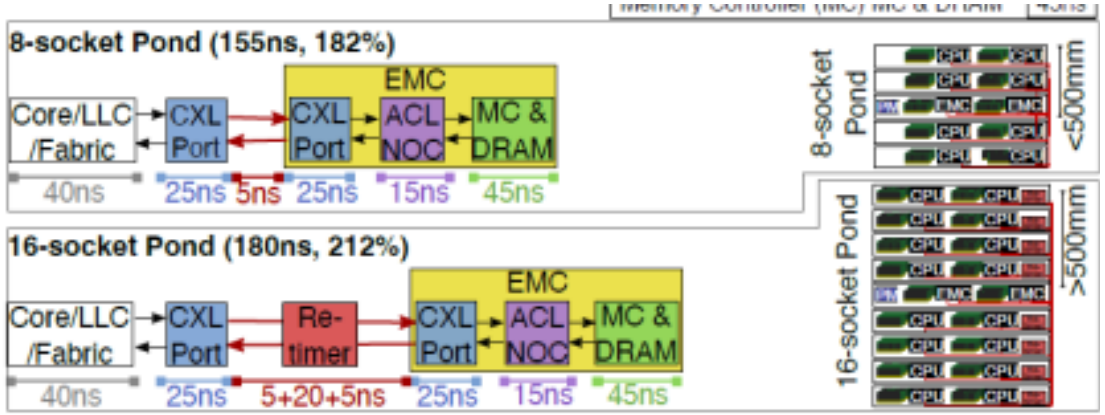
University of Michigan* Meta Inc.†



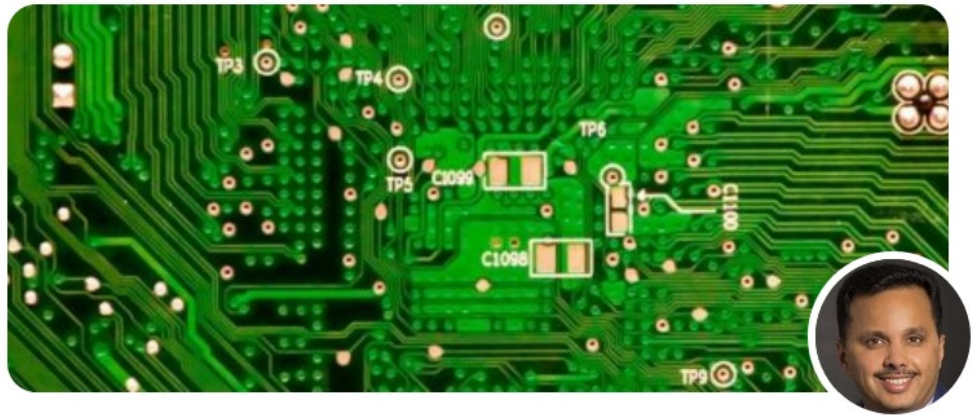
Pond: CXL-Based Memory Pooling Systems for Cloud Platforms

Huaicheng Li†, Daniel S. Berger*‡, Stanko Novakovic*, Lisa Hsu*, Dan Ernst*, Pantea Zardoshti*, Monish Shah*, Samir Rajadnya*, Scott Lee*, Ishwar Agarwal*, Mark D. Hill*°, Marcus Fontoura*, Ricardo Bianchini*

†Virginia Tech and CMU *Microsoft Azure ‡University of Washington °University of Wisconsin-Madison



February 16, 2022



Data Center

Disaggregating Memory May be the Most Exciting Trend for Computing System Design Right Now

Dattatri Mattur

https://blogs.cisco.com/datacenter/disaggregating-memory-may-be-the-most-exciting-trend-for-computing-system-design-right-now

MemLiner: Lining up Tracing and Application for a Far-Memory-Friendly Runtime

Chenxi Wang[†♣] Haoran Ma[†♣] Shi Liu[†] Yifan Qiao[†] Jonathan Eyolfson[†] Christian Navasca[†]
Shan Lu[‡] Guoqing Harry Xu[†]
University of California, Los Angeles[†] *University of Chicago*[‡]

Carbink: Fault-Tolerant Far Memory

Yang Zhou[†*] Hassan M.G. Wassel[‡] Sihang Liu[§*] Jiaqi Gao[†] James Mickens[†] Minlan Yu[†‡]
Chris Kennelly[‡] Paul Turner[‡] David E. Culler[‡] Henry M. Levy[‖‡] Amin Vahdat[‡]
[†]*Harvard University* [‡]*Google* [§]*University of Virginia* [‖]*University of Washington*

AIFM: High-Performance, Application-Integrated Far Memory

Zhenyuan Ruan Malte Schwarzkopf[†] Marcos K. Aguilera[‡] Adam Belay
MIT CSAIL [†]*Brown University* [‡]*VMware Research*

Rethinking Software Runtimes for Disaggregated Memory Extended Abstract

Irina Calciu[1], Talha Imran[2], Ivan Puddu[3], Sanidhya Kashyap[4], Hasan Maruf[5], Onur Mutlu[3], Aasheesh Kolli[2]
[1]*VMware Research,* [2]*Penn State University,* [3]*ETH Zürich,* [4]*EPFL,* [5]*University of Michigan*

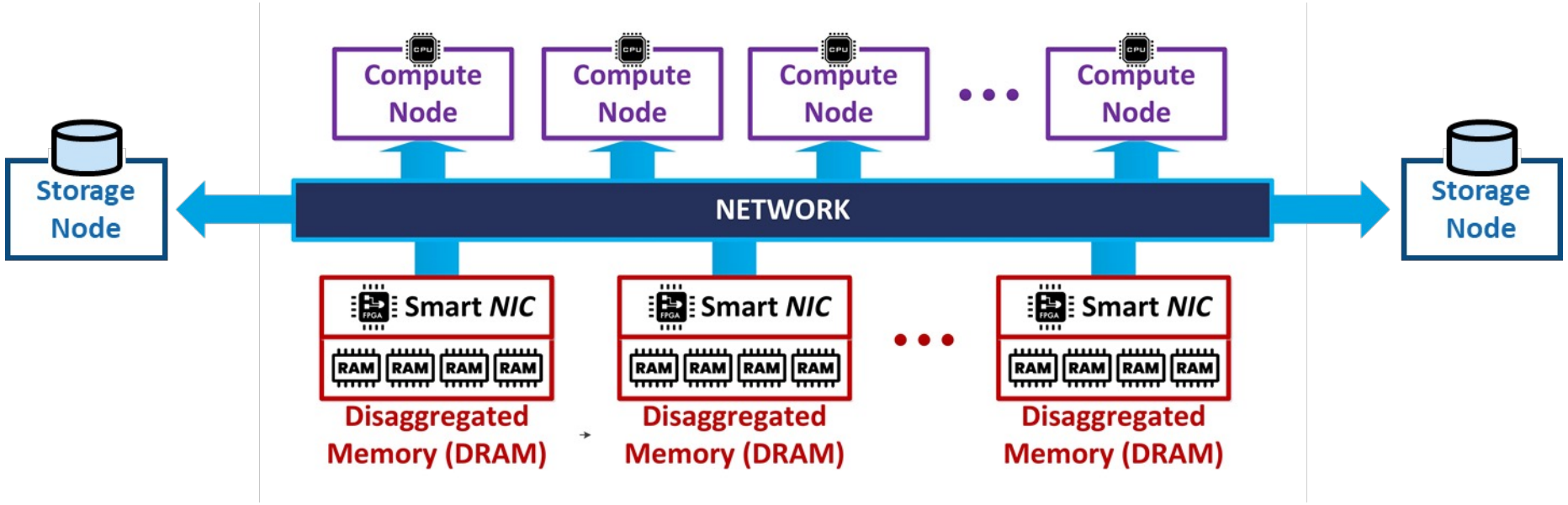
Far memory systems

- DRAM is the most expensive part of clusters these days
- Often, memory is “orphaned”: all the CPUs are taken but there is still memory available
- Disaggregated DRAM: use memory available through the network (RDMA)
  - Remote Memory: Use the memory of other machines
  - Disaggregated Memory: Network attached memory modules
- Most research focused on remote memory
- **Remote memory is inadequate for databases!!!**

Why we need to work in this area

- Database Systems 101:
  - Function vs data shipping
  - Remote memory implements data shipping
  - In databases, function shipping is more efficient but cannot be done in remote memory systems (the CPUs are busy and would interfere with whatever is in the other machine!)
- Disaggregated memory would be great but how would it look like?

Smart Disaggregated Memory (Farview)



Korolija et al. *Farview: Disaggregated Memory with Operator Off-loading for Database Engines*, CIDR 2022

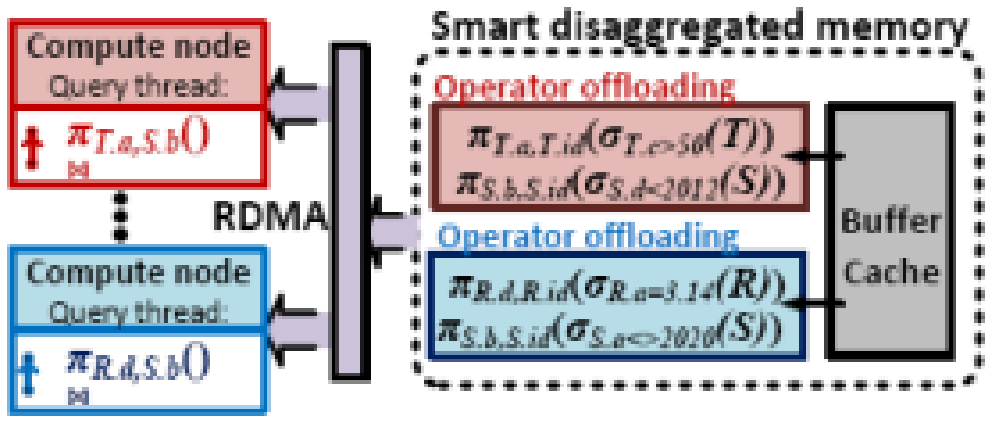
Smart Disaggregated Memory (Farview)

```
SELECT T.a, S.b
FROM T, S
WHERE T.id = S.id
AND T.c > 50 AND S.d < 2012;
```

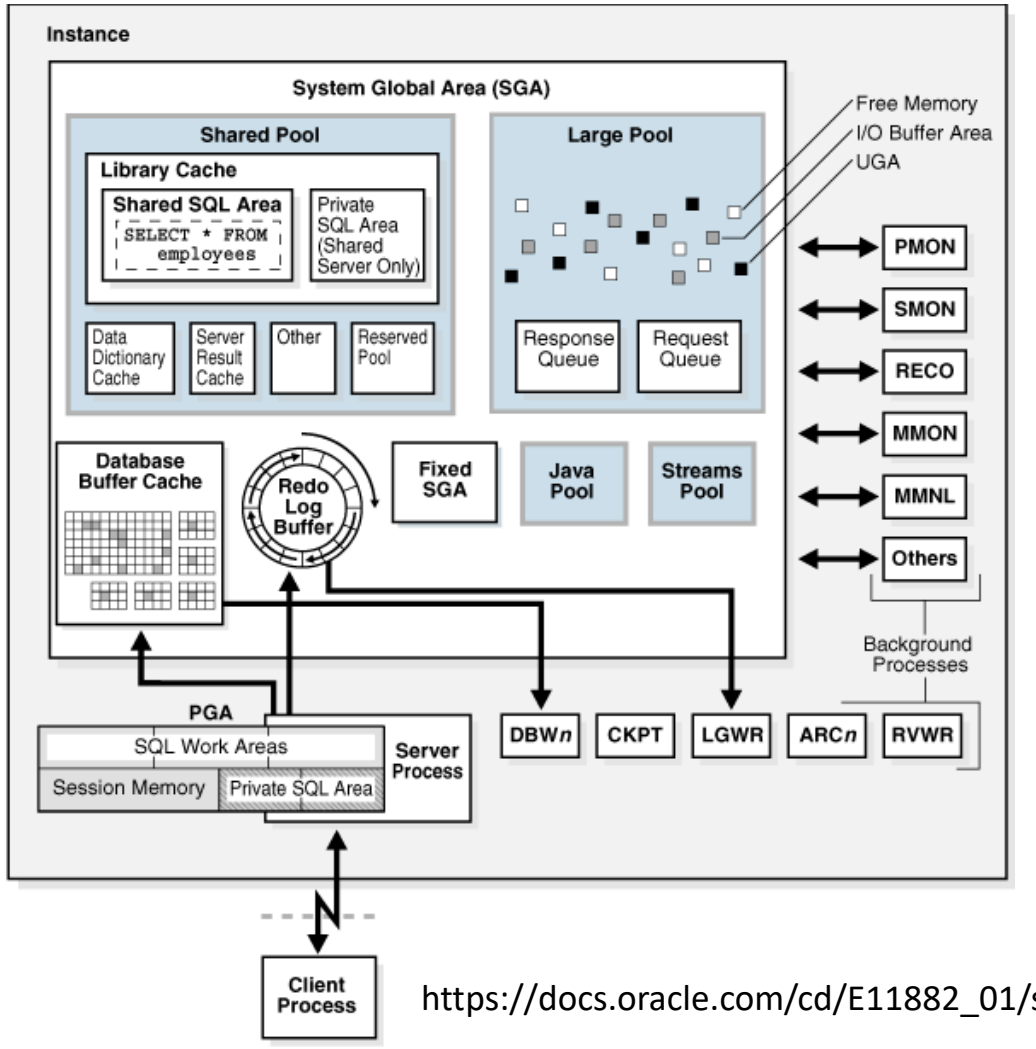
$\pi_{T.a,S.b}() \bowtie$ over $\pi_{T.a,T.id}(\sigma_{T.c>50}(T))$ and $\pi_{S.b,S.id}(\sigma_{S.d<2012}(S))$

```
SELECT R.d, S.b
FROM R, S
WHERE R.id = S.id
AND R.a = 3.14 AND S.a <> 2012;
```

$\pi_{R.d,S.b}() \bowtie$ over $\pi_{R.d,R.id}(\sigma_{R.a=3.14}(R))$ and $\pi_{S.b,S.id}(\sigma_{S.a<>2020}(S))$



Database Architecture



Research questions:

- What else can be disaggregated?
- Cache management
- Operator offloading
- Streaming operators
- Query optimization
- Updates
- …

https://docs.oracle.com/cd/E11882_01/server.112/e40540/process.htm#CNCPT902

One example: cache replacement policies

1986

An Evaluation of Buffer Management Strategies for Relational Database Systems

Hong-Tai Chou
David J. DeWitt

Computer Sciences Department
University of Wisconsin

2022

CacheSack: Admission Optimization for Google Datacenter Flash Caches

Tzu-Wei Yang, Seth Pollen, Mustafa Uysal, Arif Merchant, and Homer Wolfmeister

Google Inc.

{twyang, pollen, uysal, aamerchant, wolfmeister}@google.com

The majority of Colossus Flash Cache traffic comes from Google's database systems like BigTable and Spanner where categories can be well-defined. For database traffic, CacheSack defines a category as the combination of the table name, locality group [11, 13], and type for BigTable and Spanner, and a similar combination for other databases. Since Colossus

Conclusions

These are the best of times!

- Databases are a crucial component on every development
- But they are not ready for the new hardware and must be redesigned
- Go beyond current cloud native databases
- Embrace the new hardware era
- Let’s influence the new hardware, the opportunity is there!!

Take away mssg:

Do not work on polishing Dutch tomatoes no matter how many others do

20,000 people throw old tomatoes during huge food fight in Spain

Comment

Jen Mills
Wednesday 30 Aug 2017 10:07 pm

Share

Smart people wore goggles (AP Photo/Alberto Saiz)

https://metro.co.uk/2017/08/30/20000-people-throw-old-tomatoes-during-huge-food-fight-in-spain-6891426/