Distributed Information Management

Daniel Kocher

Salzburg, Summer term 2021

Department of Computer Sciences
University of Salzburg

Part III

Emerging Trends in Data Management

Literature, Sources, and Credits

Literature:

- Recent research papers (references given on the respective slides).

Self-Designed and Learned Data Systems

Motivation [1]

Rapidly Changing System Requirements: New applications and workload patterns occur, new hardware is developed, systems keep refining/changing.

System Assumptions: A system that is implemented under particular assumptions can only be fine tuned wrt. to these assumptions.

Goal: Automatically design a system for a particular problem.

[1] Idreos and Kraska. From Auto-tuning One Size Fits All to Self-designed and Learned Data-intensive Systems. ACM SIGMOD, 2019. https://stratos.seas.harvard.edu/files/stratos/files/selfdesignedandlearnedsystems.pdf

Self-Designed Data Systems

Systems that **know the possible design choices** (and combinations thereof) for critical system components, and are able to **automatically choose the most appropriate design for a given problem.**

Design Space: All designs that can be described as combinations of fundamental design concept. Intuitively, we collect all fundamental design concepts that have been introduced in the past to derive new valid designs (analogous to the periodic table of elements in chemistry).

Example: The Data Calculator [2] to design key-value storage.

---

[2] Idreos et al. The Data Calculator: Data Structure Design and Cost Synthesis from First Principles and Learned Cost Models. ACM SIGMOD, 2018. `https://www.eecs.harvard.edu/~kester/files/datacalculator.pdf`

Self-Designed Data Systems

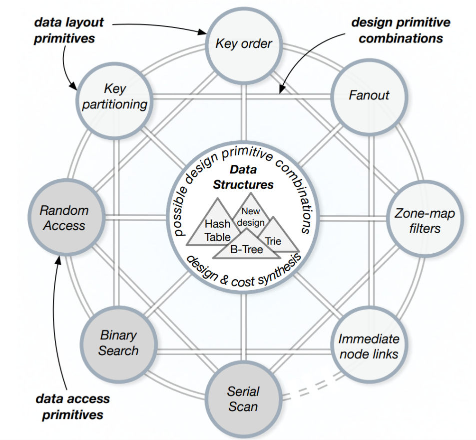


Figure taken from Idreos et al. The Data Calculator: Data Structure Design and Cost Synthesis from First Principles and Learned Cost Models. ACM SIGMOD, 2018. `https://www.eecs.harvard.edu/~kester/files/datacalculator.pdf`

Learned Data Systems

Systems that **replace critical system components with (learned) models.**

Model: Captures properties of the data and can be anything (a simple linear model or a complex neural network model).

Example: SageDB [3] is a database system where learned components are first-class citizens in its design.

[3] Kraska et al. SageDB: A Learned Database System. CIDR, 2019. http://cidrdb.org/cidr2019/papers/p117-kraska-cidr19.pdf

Learned Data Systems

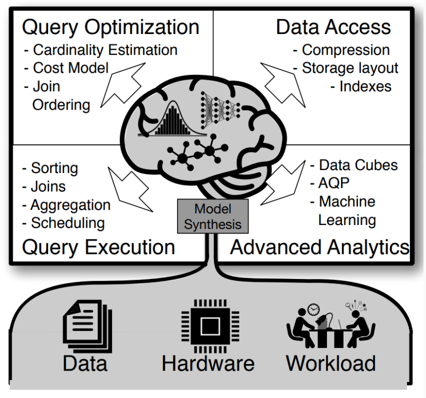


Figure taken from Kraska et al. SageDB: A Learned Database System. CIDR, 2019. http://cidrdb.org/cidr2019/papers/p117-kraska-cidr19.pdf

Self-Designed and Learned Data Systems

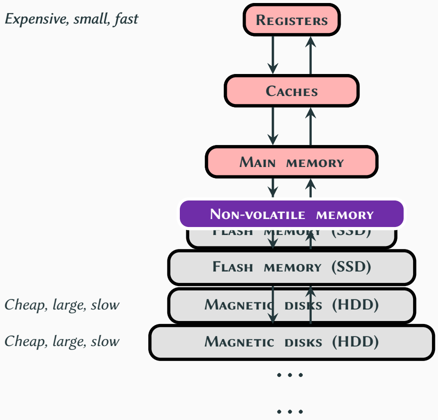
Opportunities: Design data systems that provide a wider range of performance behaviors than systems with a fixed design.

AI4DB: Make database systems more intelligent using artificial intelligence (AI).

DB4AI: Optimize AI models using database techniques.

Modern Hardware

Non-Volatile Memory (NVM)



Non-Volatile Memory (NVM) [5]

Also referred to as NVMe/NVMM/NVRAM, **storage-class memory (SCM)**, and **persistent memory (PM).**

Speed and capacity: Speed is similar to (D)RAM (byte addressability), storage capacity is similar to SSDs.

Leveraging the full power of **NVM** is **not easy.** Reexamination of database systems components is required.

Joy Arulraj received the **Jim Gray Dissertation Award** [4] for this dissertation on **how to build NVM-based database systems.**

---

[4] https://sigmod.org/sigmod-awards/citations/2019-sigmod-jim-gray-doctoral-dissertation-award/

[5] Arulraj and Pavlo. How to Build a Non-Volatile Memory Database Management System. ACM SIGMOD, 2017. https://doi.org/10.1145/3035918.3054780

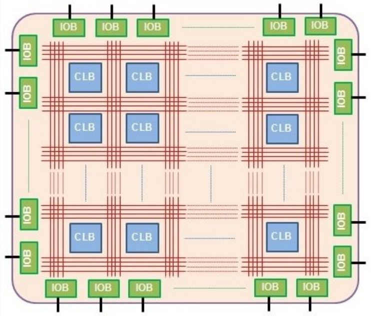
Field Programmable Gate Array (FPGA) [6]

A **programmable hardware device** that can be configured to act as a specific hardware component (i.e., electronic circuit). It consists of a large number of logic blocks that can be “re-wired” (i.e., its functionality is reconfigurable).

CPUs/GPUs are instruction-based, i.e., the programmer writes code and the CPU/GPU execute the corresponding instructions on the data. **FPGAs represent a hardware circuit** that implements a specific functionality, the **data flows through this circuit**, and the **circuit transforms the data** to produce the desired output.

[6] Fang et al. In-memory database acceleration on FPGAs: a survey. VLDB Journal, 2020. https://doi.org/10.1007/s00778-019-00581-w

Field Programmable Gate Array (FPGA)



FPGAs can be used in **multiple roles** in a (database) system including (a) bandwidth amplifier, (b) IO-attached accelerator, or (c) co-processors.

Picture taken from https://www.allaboutcircuits.com/uploads/articles/techart_FPGA_2.jpg

Remote Direct Memory Access (RDMA) [7]

A new mechanism that is supported by modern networking hardware, therefore particularly interesting to **improve the performance of distributed (database) systems.**

In a standard **TCP/IP** connection, the **data** is being **copied** by the operating system **through multiple layers. RDMA** allows the networking hardware to **directly access a main memory (RAM) location** without involving CPU or operating system.

Intuitively, a node can directly access the memory of a remote node without the remote node knowing about it ⇒ A cluster can be viewed as **one large portion of RAM.**

[7] Binnig et al. The End of Slow Networks: It's Time for a Redesign. PVLDB, 2016. https://doi.org/10.14778/2904483.2904485

Remote Direct Memory Access (RDMA)

Distributed Join Algorithms on Thousands of Cores

Claude Barthels, Ingo Müller[‡], Timo Schneider, Gustavo Alonso, Torsten Hoefler
Systems Group, Department of Computer Science, ETH Zurich
{firstname.lastname}@inf.ethz.ch

The End of a Myth: Distributed Transactions Can Scale

Erfan Zamanian, Brown University, erfanz@cs.brown.edu
Carsten Binnig, Brown University, carsten_binnig@brown.edu
Tim Harris, Oracle Labs, timothy.l.harris@oracle.com
Tim Kraska, Brown University, tim_kraska@brown.edu

Using RDMA Efficiently for Key-Value Services

Anuj Kalia, Michael Kaminsky[†], David G. Andersen
Carnegie Mellon University, [†]Intel Labs
{akalia,dga}@cs.cmu.edu michael.e.kaminsky@intel.com

Designing Distributed Tree-based Index Structures for Fast RDMA-capable Networks

Tobias Ziegler, TU Darmstadt, tobias.ziegler@cs.tu-darmstadt.de
Sumukha Tumkur Vani, Brown University, sumukha_tumkur_vani@brown.edu
Carsten Binnig, TU Darmstadt, carsten.binnig@cs.tu-darmstadt.de
Rodrigo Fonseca, Brown University, rfonseca@cs.brown.edu
Tim Kraska, MIT, kraska@mit.edu

High-Speed Query Processing over High-Speed Networks

Wolf Rödiger, TU München, Munich, Germany, roediger@in.tum.de
Tobias Mühlbauer, TU München, Munich, Germany, muehlbau@in.tum.de
Alfons Kemper, TU München, Munich, Germany, kemper@in.tum.de
Thomas Neumann, TU München, Munich, Germany, neumann@in.tum.de

Rethinking Database High Availability with RDMA Networks

Erfan Zamanian[1], Xiangyao Yu[2], Michael Stonebraker[2], Tim Kraska[2]
[1] Brown University [2] Massachusetts Institute of Technology
erfanz@cs.brown.edu, {yxy, stonebraker, kraska}@csail.mit.edu

Similarity Search