

Advances in Scalability of Databases

Technology Landscape

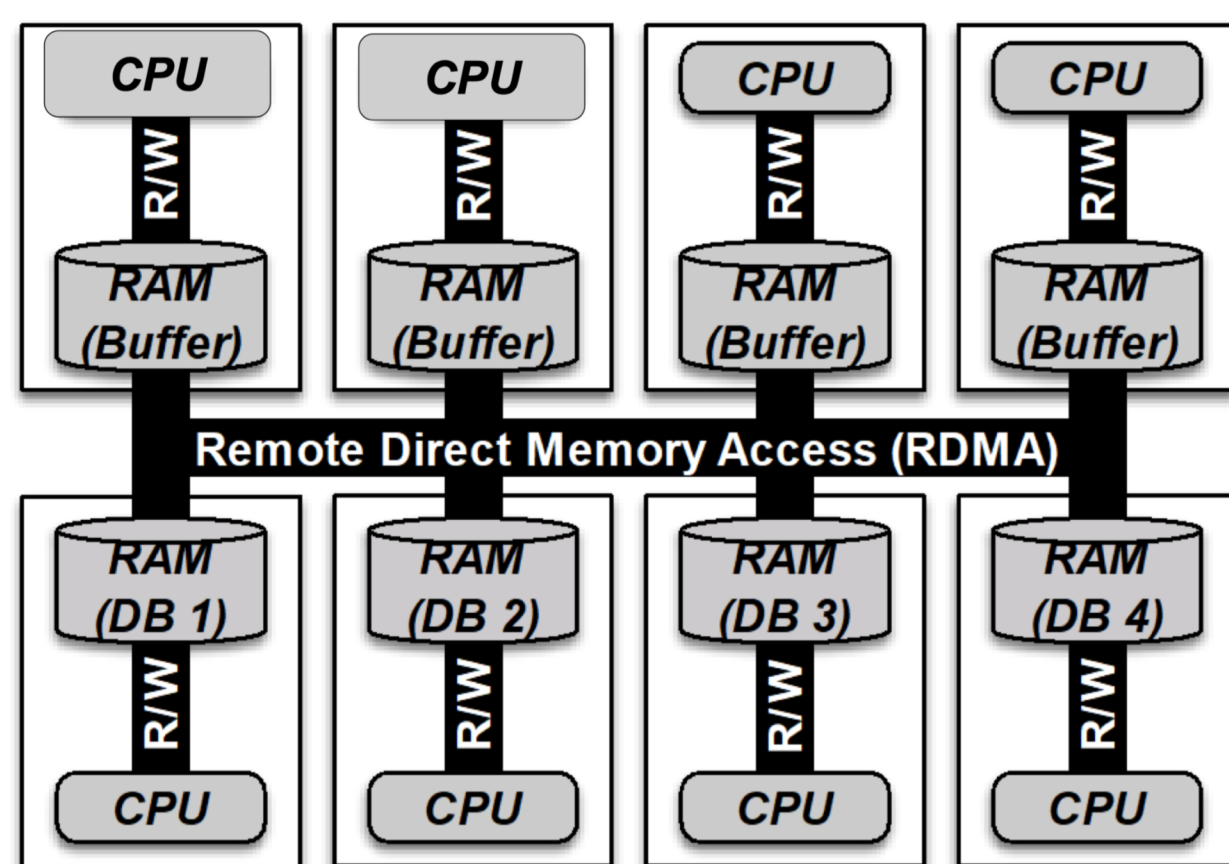
Scalability of databases is becoming more and more important as the amount of data to handle is growing and the database systems are growing with it. As simply scaling up a database usually runs into performance problems there has been a great amount of research going on to combat this problem. Here we want to take a look at 4 different approaches.

RDMA *Remote Direct Memory Access*

- Bottleneck when using more machines: network
- Prices for **faster networks** are dropping
- **Speed of main memory** can be matched
- Overhead due to TCP/IP messages
- Solution: use **RDMA instead of TCP/IP**

RDMA **bypasses the operating system**, which means that data from another machine can be read (or written) without going through the operating system stack, including the TCP/IP stack, of the other machine.

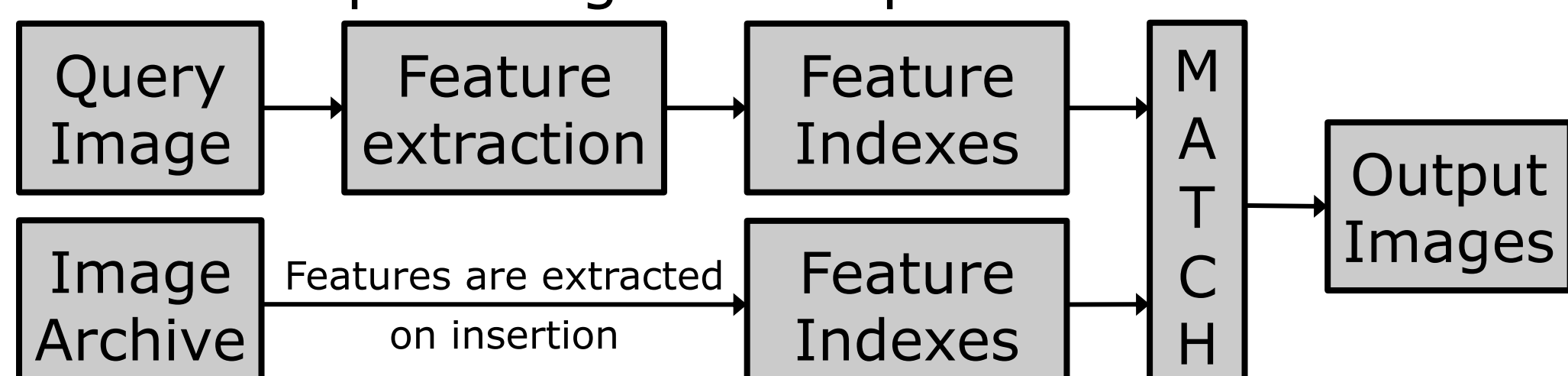
Taking this concept an architecture can be designed where all compute nodes can access all memory nodes freely. Compute power and memory can even be scaled separately.



Indexing of images

- During **content based image retrieval** images with a similar semantic content to the input image content shall be retrieved
- Not doable solely based on image data
- **Semantic descriptors** have to be generated
- Descriptors of query images and archive images can be matched

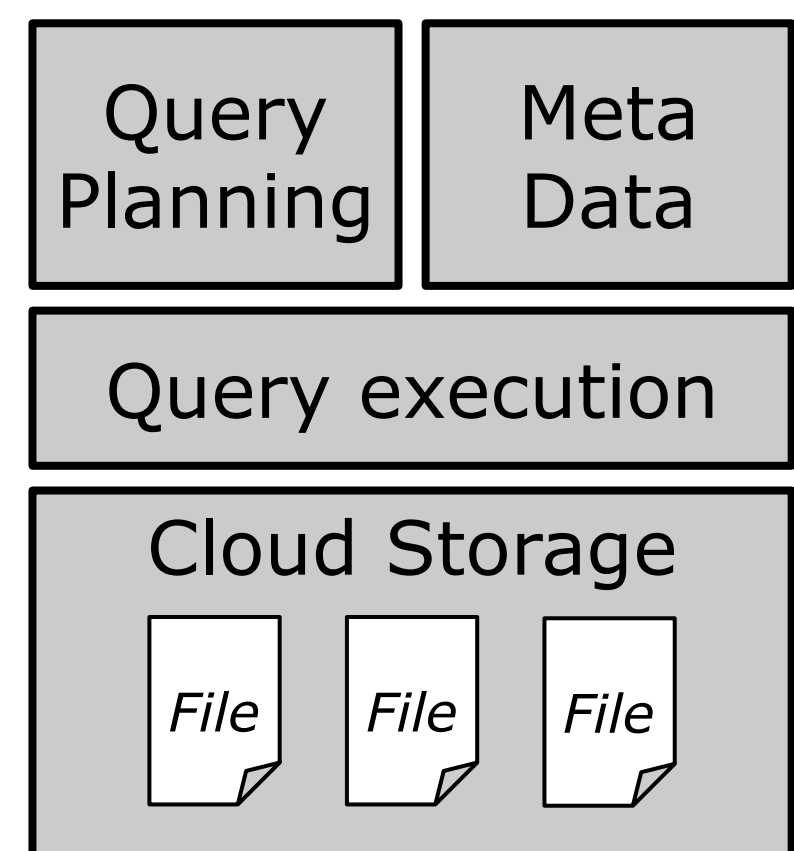
This approach is easily applicable for small data sets, using exhaustive search. Once data sets grow, **exhaustive search** is **not scalable** anymore. To combat this **image indexing** with content based hash codes can be used. Generating multiple local hash codes instead of one global hash code per image can improve this even more.



Meta Data for small immutable files

- In the cloud there is virtually unlimited storage
- Files don't need to change (**immutable**)
- To update a file a new one can be created
- **Meta data** can reflect these changes
- Meta data contains **statistics** about content

The statistics in the meta data can be used to perform **pruning** while accessing the database. This means that if the statistics about a file rule its content out for a particular selection, the file does not have to be loaded at all, which can save lots of time. Because deleted files can also be kept and just **marked as deleted in the meta data**, the meta data can also be used to reconstruct the state of the database at a previous point in time. Meta data can also be used to **clone a table** without actually having to clone data.



FPGAs *field-programmable gate arrays*

- Adding more machines to a system to scale it up leads to many problems and bottlenecks
- Another approach is to **scale in**, to improve the processing power of the existing machines with **hardware acceleration**
- Options for this are upgrading the CPU or GPU, installing ASICs or using **FPGAs**

While upgrading the CPU and even more so developing ASICs is expensive and using GPUs comes with memory problems FPGAs are **cheaper** and more **flexible** as their logic can be reprogrammed. Integrating FPGAs into servers gives the ability to efficiently and parallelly handle requests, which increases scalability without adding machines.

