MAGPIE: A Scalable Data Storage System for Efficient High Volume Data Queries

Thomas Lindemann[1] Patrick Brinkmann[2] Fadi Dalbah[2] Christian Hakert[2] Philipp-Jan Honysz[2] Daniel Matuszczyk[2] Nikolas Müller[2] Alexander Schmulbach[2] Stefan Petyov Todorinski[2] Oliver Tüselmann[2] Shimon Wonsak[2] Jens Teubner[1]

Abstract: Modern challenges in huge sized data storage and querying require new approaches in the field of data storage systems. With *MAGPIE*, we are introducing a hardware-software-co-design, which is efficient in querying data by distributed storage with storage-near pre-processing and designed to be scalable up to large dimensions.

Keywords: data analysis, big data, database, distributed computing, modern hardware

1 Introduction

Beyond the "Big Data" age, data is keeping on growing further year after year in all areas such as business, science and social media. Especially in the scientific field, the data sizes have exceeded all previous data sizes and are therefore often referred to as high-volume data.

The challenge is to develop new storage solutions which can store big amounts of data and make it accessible for fast analyses. With *MAGPIE*, we present our approach to aim the goals of scalable storing huge amounts of partitioned data and allowing efficient parallel distributed pre-processing. Figure 1 shows a comparison between classic database configuration and the approach that *MAGPIE* is based on. To achieve this, we have created a distributed hardware-software-co-design, which consists of SSD storage devices with storage-near embedded systems on the hardware side and a software framework, which allows to pass all operators that are able to process independently on the partitioned data in form of predicates to the storage nodes. Thus, we got a system that is scalable because of arbitrary partitioning on independent storage nodes and allows efficient querying on distributed storages through lightweight embedded systems in the storage layer.

[1] TU Dortmund University, Databases and Information Systems Group, {firstname.lastname}@cs.tu-dortmund.de
[2] TU Dortmund University, Student Project Group PG614, {firstname.lastname}@tu-dortmund.de

doi:10.18420/btw2019-41

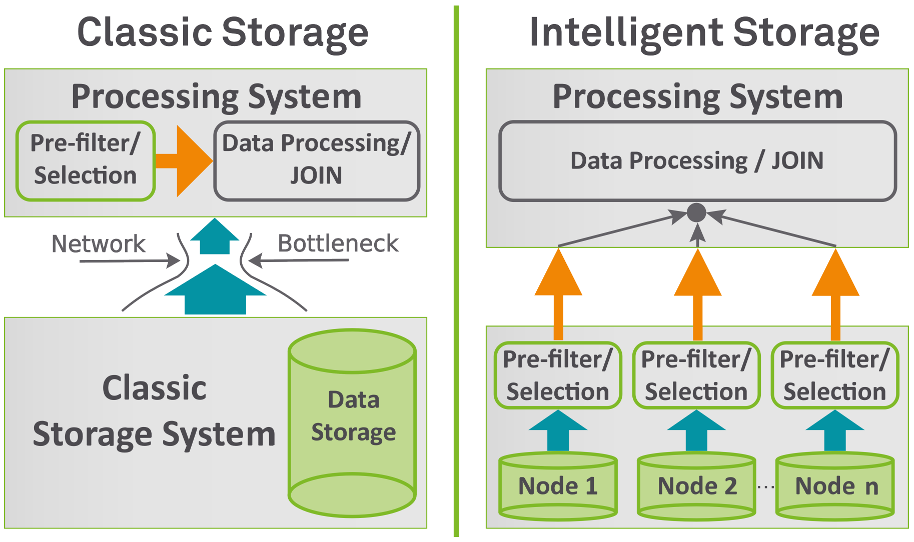

Fig. 1: Differences between the classical configuration and intelligent storage approach of *MAGPIE*

2 Hardware Construction Test Platform

To test our approach, we created a hardware platform in shape of a 2U 19"Blade with integrated embedded systems, SSD storage devices and power supplies with a custom designed integrated power control and energy consumption measurement. The system is by design scalable by just adding more similar blades through a fast network backend. We have used power efficient embedded systems for our evaluation system, because we expect that high performance processors have no advantage for the scans and filter operations. Figure 3 shows the test platform which has been created to make our experiments with the *MAGPIE* Framework on modern embedded hardware.

- AAEON UP2 Embedded Nodes:
  - 32 (8x 4) Cores N4200
  - 8x 2 MB L2 Cache
  - Base Clock: 1.1 GHz
  - Boost Clock: 2.4 GHz
  - 8x GPU Intel HD505
  - 64GB (8x 8GB) LPDDR4
  - Inter-Network: 8x 2 Gbit/s
  - Uplink to Client: 2x 10 GBit
  - Storage: 16x SATA
- SSD Storage:
  - 8 TB (16x 500 GB) SSD
  - Read speed: 16x 550 MB/s
  - Write speed: 16x 520 MB/s
  - Cache: 16x 512MB LPDDR4

Fig. 2: *MAGPIE* System Specifications

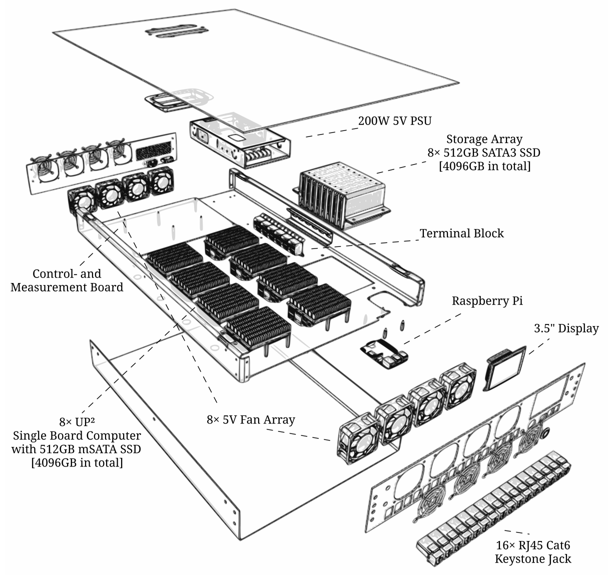
200W 5V PSU
Storage Array 8× 512GB SATA3 SSD [4096GB in total]
Terminal Block
Control- and Measurement Board
Raspberry Pi
3.5" Display
8× UP² Single Board Computer with 512GB mSATA SSD [4096GB in total]
8× 5V Fan Array
16× RJ45 Cat6 Keystone Jack

Fig. 3: *MAGPIE* Hardware Construction

3 Storage-sided Distributed Data Pre-processing with *MAGPIE*

There's already a lot of research on distributed storage near data processing done my commercial companies, one of the most popular examples is IBM's PureData System for Analytics, which is based one the rebranded Netezza Architecture. It all comes down to the AMPP (Asymmetrical Massive Parallel Processing) engine. [IB14] A similar approach is shown by Oracle in their Oracle Exadata Database Machines, which has been engineered to be the highest performing and most available platform for running Oracle Database. [Or16] The disadvantage is that these systems are of a commercial nature and not open for research on different processing strategies. An energy-aware approach is shown in [LWA13], the authors introduce IBEX, an FPGA accelerator that allows a number of more complex operators to be pushed down into the storage engine of a database. The result of this is increased performance for certain queries and reduced power consumption. Compared to the FPGA approach of IBEX, *MAGPIE* is much more flexible but still energy-aware.

Figure 4 shows a schematic of the *MAGPIE* Framework. The software concept is designed of three main components, the user node (client), the master worker node (master) and the worker nodes (slaves). The client is supposed to run on the workstation, it is the interface between the user/application querying the database and the *MAGPIE* cluster and receives the query result from the user. Selected operations which are possible to run distributed on all worker nodes are forwarded to the master node. The master is also a worker node itself but keeps track of the tables in a catalog, after doing the lookup which workers have partitions of the tables involved in the sub-queries, it forwards the queries to the respective worker node. It is also possible to maintain the catalog redundantly to avoid having a single point of failure. In theory, every worker node can be a master node in case of a malfunction of the current active master. To avoid the master worker to become a bottleneck, the results of the queries of all worker slaves are sent directly to the user client.

After the worker have completed the scans and filters on all their partitions for the current query id, they send an acknowledge to the master, the master worker collects all the acknowledge messages and forwards one commit to the user client when all slaves completed their tasks for this id. We implemented the scans algorithms, the predicate filter, the buffer manager, the catalogs and the TCP communication from scratch, due to performance optimization.

4 Comparison to state-of-the-art Hardware and Software

To evaluate the performance of the Hardware-Software-Co-Design of *MAGPIE*, we have run similar test workloads on state-of-the-art hardware and common used software. Thus, we executed different test scenarios with a generated 100GB TPC-H dataset on a dual-socket AMD EPYC 7501 32-Core Server with two CPUs, 32 cores each, one thread per core and stored the input data on a ramdisk to avoid a bottleneck with the hard drives.

The comparison software of choice was PostgreSQL 9.6.10 on x86_64-pc-linux-gnu, compiled by gcc (Debian 6.3.0-18+deb9u1) 6.3.0 20170516, 64-bit. We tested different filters and aggregations, as well as a modified TPC-H Q6 scan query. The plots in Figure 5 show the results on *MAGPIE* and the comparison system.

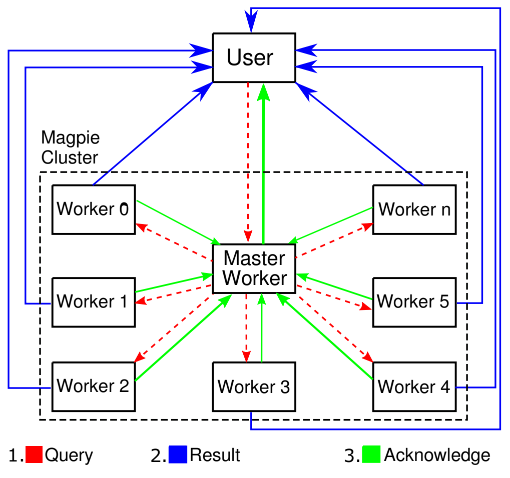

Fig. 4: *MAGPIE* Software Architecture

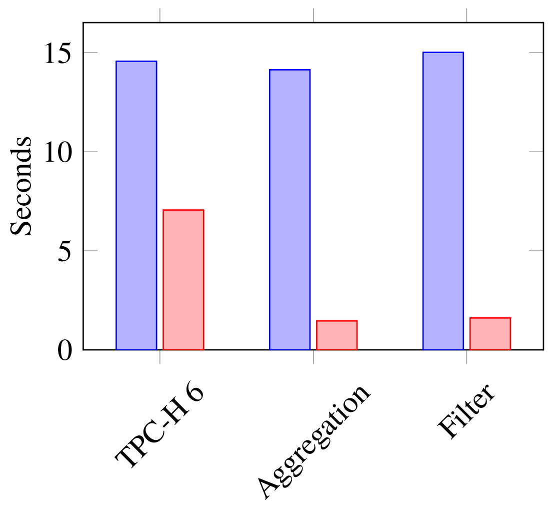

Fig. 5: Execution Time of Queries Comparison on Magpie vs. Reference Server w/ PostgreSQL

5 Demo Outline

For our demo, we are planning to bring along our *MAGPIE* test system as an exhibition piece. In our demo we want to show how *MAGPIE* processes different queries. We can also query a schema listing of the distributed TPC-H data which is stored on the system nodes. Moreover, we created a graphical interface for our demo that allows to execute queries on the *MAGPIE* system interactively and makes an output of the results. For a running query, we will log the system's execution time and power consumption. Furthermore, we want to observe the system load during execution.

References

[IB14] IBM PureData System for Analytics Architecture, `https://www.redbooks.ibm.com/redpapers/pdfs/redp4725.pdf`.

[LWA13] Louis Woods, Jens Teubner; Alonso, Gustavo: Less Watts, More Performance: An Intelligent Storage Engine for Data Appliances. Proceedings of the 2013 ACM SIGMOD Conference on Management of Data, April 2013.

[Or16] Oracle Exadata Database Machine, `https://www.oracle.com/technetwork/database/exadata/exadatatechnicaldeepdive-3518309.pdf`.