And all of a sudden: Main Memory Is Less Expensive Than Disk

Martin Boissier, Carsten Meyer, Matthias Uflacker, Christian Tinnefeld

Hasso Plattner Institute,
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
{fistname.lastname}@hpi.de

Abstract. Even today, the wisdom for storage still is that storing data in main memory is more expensive than storing on disks. While this is true for the price per byte, the picture looks different for price per bandwidth. However, for data driven applications with high throughput demands, I/O bandwidth can easily become the major bottleneck. Comparing costs for different storage types for a given bandwidth requirement shows that the old wisdom of inexpensive disks and expensive main memory is no longer valid in every case. The higher the bandwidth requirements become, the more cost efficient main memory is. And all of sudden: main memory is less expensive than disk.
In this paper, we show that database workloads for the next generation of enterprise systems have vastly increased bandwidth requirements. These new requirement favor in-memory systems as they are less expensive when operational costs are taken into account. We will discuss mixed enterprise workloads in comparison to traditional transactional workloads and show with a cost evaluation that main memory systems can turn out to incur lower total costs of ownership than their disk-based counterparts.

Keywords: TCO, Mixed Workload, Bandwidth, In-Memory Systems

1 Introduction

Until today the wisdom for storage still is: storing data in main memory is more expensive than storing on disks. Especially with the recent rise of main memory-resident database systems, this cost comparison is often brought up as an argument for disk-based systems.

While this is true comparing the price per byte, the picture looks different if we compare the price per bandwidth. But the price of provided bandwidth from the primary persistence to the CPUs is of high importance as modern applications have increasing bandwidth requirements, e.g., due to demands for big data analytics, real-time event processing, or mixed enterprise workloads including operational reporting on transactional data. When handling such workloads bandwidth easily exceeds single disks. In such cases, local disk arrays or storage servers are employed with large RAID installations in order to meet bandwidth requirements by parallelizing I/O over multiple disks.

Big Data Benchmarking, 5th International Workshop, WBDB 2014, Potsdam, Germany, August 5-6 2014, Revised Selected Papers
The final publication is available at Springer via
http://dx.doi.org/10.1007/978-3-319-20233-4.

What is missing in our opinion is a thorough analysis that quantifies which costs are to be expected when bandwidth requirements exceed traditional database deployments. With increasingly large setups to provide sufficient I/O bandwidth it is often overlooked how expensive such systems can get, especially when including operational costs.

In this paper, we take a closer look at mixed workloads on relational databases. Mixed workloads combine transactional workloads (OLTP) with analytical workloads (OLAP) on a single system. Basically all major database vendors are currently adding analytical capabilities to their database products to allow operational reporting on transactional data and remove static materialized aggregates [10, 12, 13], underlining the necessity to research the rather new field of mixed enterprise workloads. Most of these solutions use main memory-resident data structure in order to provide sufficient performance. The reason is that disk-based data structures as they are used in most recent relational database have shown to be too slow compared to main memory-based solutions even when fully cached [7].

Disk-based databases are proven to be a good fit for traditional OLTP workloads, but their viability has to be re-evaluated for mixed workloads. The traditional approach of storing data on disk and caching the most recently used pages in memory is no longer viable as analytical queries often require to process vast numbers of tuples. The requirements of mixed workloads, which are e.g. analytical queries on transactional data – without pre-computed aggregates – increase bandwidth requirements significantly. We think that this trend will continue and steadily increase bandwidth requirements for modern database systems, rendering main memory-based systems increasingly feasible and viable.
Throughout this paper, we will make the following contributions:

- Estimation of the total cost of ownership (TCO) of several system configurations fulfilling certain bandwidth requirements (Section 2). We will make a simplified cost calculation to compare HDD-based (hard disk drive-based), SSD-based (solid state disk-based), and MM-based (main memory-based) servers including acquisition costs as well as operational costs.
- Evaluation of bandwidth requirements of a mixed workload using the CH-benCHmark (Section 3) for three relational databases.
- A discussion about the break-even point at which main memory-resident databases start to be less expensive than their disk-based counterparts. We project the usage of upcoming enterprise systems and combine it with bandwidth requirements discussed in Section 3 to analyze at which point a MM-based system will be the least expensive solution.

We think that this discussion is also important for the field of big data benchmarking. First, even big data veterans as Google recently added transactional features to their F1 database [15], showing that transactional safety might also be important for systems considered as being "big data". Hence, benchmarking enterprise workloads is of increasing relevance. Second, we argue that most benchmarks – TPC-* as well as other current benchmark proposals for big data – need to include operational costs, because our calculations for the three-year costs of large scale-out systems show that operational costs can become the dominating cost driver.

Last but not least, no matter how the application looks, a thorough analysis of bandwidth requirements and the resulting costs are of relevance for any larger server system.

2 Cost Calculations

In order to quantify the costs of bandwidth, we calculate the prices for three different server configurations in this Section: SSD-, HDD-, and MM-based servers. For this calculation, we started with a given bandwidth requirement and a data set size of 500 GB. Depending on the bandwidth we configured the servers using market-available components that are able to fulfil the bandwidth requirements. The HDD- and SSD-based calculations are done using vendor data for maximal bandwidth, not taking into account that these bandwidths are hard to achieve in real-life.

As we are arguing for MM-based systems, we will use more realistic assumptions here in contrast to the other system configurations. Consequently, we do not assume a maximal bandwidth of 80 GB/s per CPU (or 320 GB/s for a four socket node as stated in the technical specifications) but Intel's results for a standardized CPU benchmark.

To be capable of handling actual enterprise workloads, all server configurations have a high-performance PCI-e connected SSD for logging. Obviously, the main memory server also includes sufficient persistent storage to store at least one snapshot of the database.

The main memory size for the SSD- and HDD-based systems is set to ~10% of the whole data set. In our calculations, the data set size is 500 GB consequently the main memory size is 50 GB.

All server configurations are build using a modern four socket server blade. Since the discussed workloads in this paper are bandwidth bound, the processors for the SSD- and HDD-based servers are Intel Xeon E7-4850v2 CPUs. For the MM-based server we decided for a more expensive CPU with an improved main memory to CPU throughput.

We do not consider possible locking or contention for any of the configurations, neither disk-based systems nor main memory-based systems. We do also not include costs for networking (network cables, switches, et cetera).

High-Availability and Durability For all configurations and calculations we include both the costs for acquisition as well as operational costs. Furthermore, we include costs for high availability. For each configuration, we assume a system to be highly available when one server node can fail as there is an additional spare node in each configuration. I.e., for a single server node configuration high availability can increase total costs by a factor of two.

Durability issues and failures of single components as hard disks or DRAM chips are not considered.

2.1 Server Configurations

Main Memory-Based System The assumed main memory-based server is a four socket system equipped with four Intel Xeon E7-4890v2 CPUs. Recent benchmarks by Intel have shown that such a system achieves a bandwidth of up to ~246 GB/s for the STREAM Triad Benchmark[1]. Any bandwidth exceeding the maximum of 246 GB/s requires a multi-node setup to scale.

We assume that the memory size has to be at least a factor of two larger than the data set. This space is required for the operation system (OS), intermediate results, et cetera. Consequently, the main memory-based systems includes 1 TB of DRAM for the 500 GB data set, a PCIe-connected SSD for logging, and a 500 GB HDD for database persistence (e.g., snapshotting).

HDD-Based System The HDD-based server is a four socket node equipped with four Intel Xeon E7-4850v2 CPUs. The size of main memory is set according to Oracle's MySQL sizing documents [16], which recommend to reserve main memory to cache 5%-10% of the data set size. Consequently, we assume 25 GB main memory for database and 25 GB for the operation system (i.e., 50 GB).

The disks are put in direct-attached storage (DAS) units, where each DAS unit contains up to 96 disks in a RAID array. Two SAN controllers are used to connect each DAS unit. This setup yields a bandwidth of 6 GB/s (each SAN controller has a theoretical bandwidth of 3 GB/s) per DAS unit. Per server node up to eight SAN controllers can be used. Consequently, the peak bandwidth per HDD-based server is 24 GB/s ($4 * 2 * 3GB/s$).

The systems adapts to increasing bandwidth requirements first by adding DAS units and then by adding server nodes.

SDD-Based System The solid state disk-based systems are configured using SSDs that are connected via the PCI Express (PCIe) bus as the latest generation of SSD exceeds the bandwidth of SATA ports. Each SSD provides a read bandwidth of 3 GB/s. Using Intel Xeon E7-4850v2 CPUs we assume that each socket can directly connect two PCIe 16x SSDs at full bandwidth. For a four socket server a maximum of eight PCIe-connected SSDs can be used, yielding a peak bandwidth of 24 GB/s.

The systems adapts to increasing bandwidth requirements first by adding PCIe SSDs and then by adding server nodes.

2.2 TCO Calculations

We calculated the TCO for the three server configurations using varying bandwidth requirements.

For all configurations we assume a data set size of 500 GB. Even with low bandwidth requirements this configuration is already comparatively expensive

[1] Intel Xeon E7-4890v2 Benchmark – URL: `http://www.intel.com/content/www/us/en/benchmarks/server/xeon-e7-v2/xeon-e7-v2-4s-stream.html`

in a main memory-based configuration, because the server has to be able to store the whole data set in main memory.

Please note, that the sum for the high availability costs shown in Figure 1 represent the costs for the added server node including operational costs. The other two sums (acquisitional and operational costs) show the total costs for the system without high availability.

Configuration 1 - 10 GB/s Bandwidth In case of a required bandwidth of 10 GB/s a HDD-based system is the less expensive solution (see Figure 1(a)). The SSD-based server has significantly lower operational costs, but suffers from high availability costs as well as the MM-based solution. The MM-based server is comparatively expensive since – as stated above – main memory has to be sufficiently large to store the data set entirely in memory. In our calculations that results in a memory size of 1 TB.

For a fair comparison, it has to be said that the server configurations are built to scale to higher bandwidth requirements. Especially for a bandwidth requirement as 10 GB/s there are more price efficient configurations for disk-based setups.

Configuration 2 - 20 GB/s Bandwidth The picture looks different if we assume a bandwidth of 20 GB/s (see Figure 1(b)). The HDD-based solution is still the least expensive one, but the main memory-based solution is already less expensive than the SSD-based server.

The main cost driver for SSD-based over MM-based systems are expensive PCIe-connected flash drives of which seven are required to theoretically provide the required bandwidth. Even though PCIe-connected SSDs outperform HDDs, the performance to price ratio is not significantly improved compared to recent 15K HDDs.

Configuration 3 - 40 GB/s Bandwidth As shown in Figure 1(c), for a bandwidth requirement of 40 GB/s, the costs for a MM-based solution are lower than for the other two configurations. With this bandwidth requirement, costs for HDD-based servers are clearly dominated by the operational costs while SSD-based server are again dominated by the costs for PCIe-connected flash drives.

3 Bandwidth Requirements for Enterprise Workloads

To better understand the TCO for a given bandwidth requirement we measured the reading bandwidth of mixed enterprise workloads. Mixed workloads include transactional workloads as they are executed daily in business systems and further include analytical workloads, which are usually handled by data warehouses. Mixed workload systems are currently being researched intensively (e.g., HYRISE [5], H-Store [6], HyPer [9]) and are also the focus of several commercial products (e.g., SAP HANA [4], Oracle Times Ten, IBM DB2 BLU [13]).

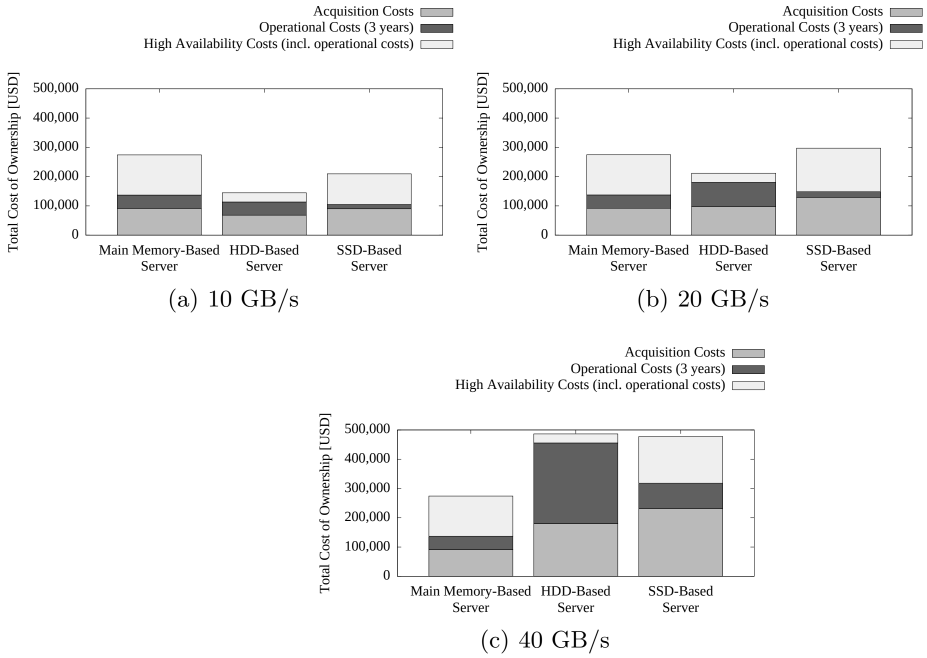

Fig. 1. Total costs of ownership for the server configurations (see Section 2.1) with varying bandwidth requirements.

They all aim at introducing analytical real-time capabilities into transactional systems.

To quantify how much read bandwidth is required executeing mixed workloads we chose the CH-benCHmark [2] as the assumed workload. The CH-benCHmark is a modification of the TPC-C benchmark. To include analytical characteristics of mixed workloads the CH-benCHmark runs analytical queries adopted from the TPC-H benchmark on the transactional TPC-C data set.

3.1 Setup

We created a TPC-C data set with a scale factor of 1000. The data set was generated using the OLTP-Bench framework [3] and has a size of ~70-85 GB (depending on the database used). We used *iotop*[2] to record reads from disk. Execution times for the analytical CH-benCHmark queries are not discussed in this paper as our focus is on the required bandwidth. We did not consider further characteristics as latency or IOPS (input/output instructions per second) due to our focus on mixed workloads, which are read-dominated. Also, we expect main memory-based systems to outperform disk-based systems for these characteristics anyhow.

[2] iotop – URL: `http://guichaz.free.fr/iotop/`

We evaluated three relational databases: MySQL version 5.6.4, PostgreSQL version 9.3.4, and the most recent release of a commercial relational disk-based database (referred to as *DBMS X*). For all installations, we left all settings to their default values.

The virtualized database server has one terabyte network connected storage, 32 GB of main memory, and is running SUSE Linux Enterprise Server 11 patch level 2 respectively Windows Server 2008 RC2.

3.2 CH-benCHmark

Because the CH-benCHmark is based on TPC-C, we decided to measure each analytical CH-benCHmark query on its own in order to quantify how much additional I/O is generated by running analytical queries on a transactional data set. We had to modify several queries to be executable on the three chosen databases. Whenever non-trivial changes were required (see Section 7.1) we skipped that particular query to avoid unfair comparisons.

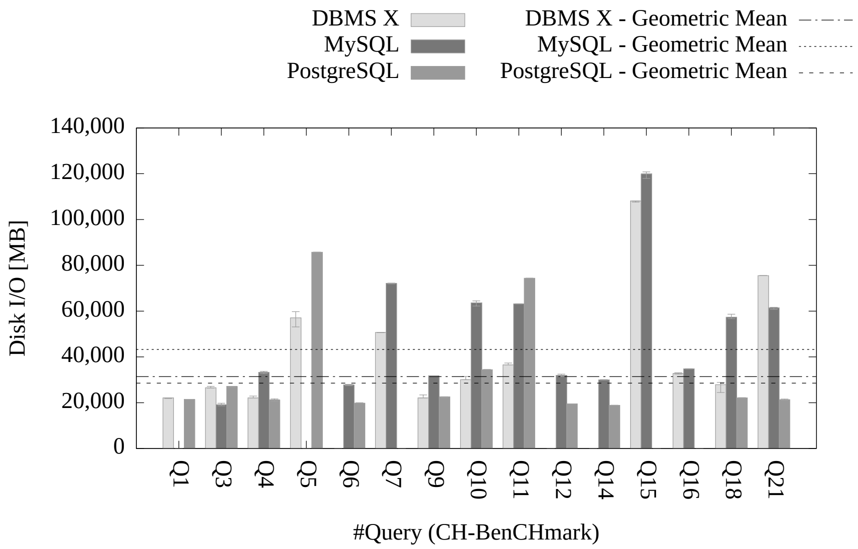

Fig. 2. Megabytes read from disk for MySQL, PostgreSQL, and DBMS X running the analytical queries of the CH-benCHmark. Geometric means are calculated using all queries that ran on all three databases (i.e., Q3, Q4, Q9, Q10, Q11, Q18, and Q21).

We executed each query five times with one non-included execution upfront to warm up the database and OS file caches. Figure 2 shows the measured average data transfers of each query execution with error bars showing the minimum/maximum measurement of the five executions. For each database, a vertical line shows the geometric means of read data for all queries that ran on all three databases (i.e., Q3, Q4, Q9, Q10, Q11, Q18, and Q21).

	Data Read from Disk (MB)	
Database	Geometric Mean	Average
DBMS X	31,368.7	34,373.7
MySQL	43,258.9	47,055.2
PostgreSQL	28,550.3	31,832.4

Table 1. Geometric mean and average of megabytes read from disk for MySQL, PostgreSQL, and DBMS X running the analytical queries of the CH-benCHmark.

4 Discussion

The next generation of enterprise systems will have vastly different characteristics compared to today's systems. In 2014, Plattner presented such an (already productive) enterprise system [12]. These new systems have to handle an increasing analytical pressure without materialized aggregates in order to provide higher flexibility and fewer restrictions on analytical queries, thus executing all calculations directly on the transactional data. Today, basically all major database vendors are working on techniques to provide sufficient performance for mixed workloads [4, 10, 13]. Hence, we think that mixed workload benchmarks as the CH-benCHmark are an important indicator of upcoming workload requirements and are well worth examining.

To discuss the break-even point of bandwidth requirements and main memory-based servers, let us assume an enterprise system with 500 parallel users and a data set size of 1 TB. Each user executes one analytical query on the database system every two hours, resulting in 250 analytical queries per hour. Assuming the average query reads of 31 GB (see Section 3.2), this scenario would have a bandwidth requirement of ~21.5 GB/s. As shown in Section 2, we see that SSD-based servers are less expensive in this scenario. However, neither the number of 500 parallel users and especially not the database size of 1 TB is what counts as a "large enterprise system" today. Especially the assumption of one analytical query per user every two hours is rather conservative. If we assume 500 parallel users executing analytical queries once an hour, the bandwidth requirements would already favor main memory-based solutions.

5 Related Work

To our best knowledge, there is no previous work that has yet discussed cost effectiveness of main memory databases for high-bandwidth workloads. However, there are several publications that discuss the link between architectural decisions and their impacts on costs.

Rowstron et al. presented a similar argument to ours for big data analytics [14]. The authors argue that scaled-up servers with large main memory volumes are often a better solution than disk-based Hadoop clusters. This observation is especially interesting as Rowstron et al. point out that the median MapReduce job size of Yahoo and Microsoft is smaller 14 GB and that 90% of Facebook's jobs process less than 100 GB of data.

Another related topic is how to improve operational costs for server systems. While this topic is well researched for traditional architectures, it is rather new to systems where massive main memory sizes are responsible for a large share of the energy consumption. Malladi et. al discussed this topic and found that energy can be decreased by a factor 3-5 without sacrificing too much performance [11]. However, lowering the power of main memory also lowers the maximal bandwidth.

Another very interesting aspect – one that is not covered in this paper – was discussed by Zilio et al. [17]. They argue that modern software systems have become increasingly sophisticated requiring several domain experts to handle and tune the systems: "These economic trends have driven the total cost of ownership of systems today to be dominated by the cost of people, not hardware or software". Of course, such a metric is hard to quantify. But it can also be seen as arguing in our favor because an often proposed advantage of main memory systems is the simplification of system architectures [1].

6 Future Work

There is still a considerable amount of work ahead of us. The main topics that we want to continue working on are:

Bandwidth evaluation To further evaluate bandwidth requirements we want to examine the CH-benCHmark from end to end, including bandwidth requirements for transactional queries as well as for analytical queries.
Besides the already evaluated databases, we are planning to benchmark other databases to gain a better overview. These alternatives include IBM DB2 with its columnar accelerator BLU [13] and MonetDB [8], both representing disk-based open sourced databases.
We expect columnar databases as MonetDB to require less I/O when executing analytical queries. However, it will be interesting how high the I/O overhead for tuple reconstruction using columnar data structures (and compression) is. Furthermore, it is interesting to see how the usage of compression effects read bandwidth.

General workload assumptions It is very hard to estimate and predict how workloads might look if databases are capable of mixed workloads and high performance analytics. We want to talk to experts and find out which workload assumptions are realistic for the next years and how workloads might look in the future.

Query cost model In the long run, we want to work towards a general cost model for bandwidth requirements of workloads. Even though we think that a holistic view over several different databases is already very helpful, there are still many variables in each implementation that are hard to factor out.

Emerging Memory / Storage Technologies The performance developments of disk-based storage and main memory will probably lower the break-even point even more as the gap between both is currently still widening. New developments as non-volatile memory (NVM) thrive to increase the density of byte-addressable storage significantly, potentially having huge impacts on the TCO of database systems with high bandwidth requirements.

7 Conclusion

The bandwidth evaluations of mixed enterprise workloads in this paper have shown that the requirements of upcoming enterprise systems might very well have completely different bandwidth requirements compared to current enterprise workloads. Comparing disk- and main memory-resident databases in respect to bandwidth shows that main memory-resident databases are not as expensive as often expected. In fact, main memory can be the least expensive storage medium. We think it is import to convey a new point of view in which main memory-based solutions are not "the most expensive solution only viable when everything else is too slow" but rather "the least expensive solutions when performance requirements are high".

References

1. M. Boissier, J. Krueger, J. Wust, and H. Plattner. An integrated data management for enterprise systems. In *ICEIS 2014 - Proceedings of the 16th International Conference on Enterprise Information Systems, Volume 3, Lisbon, Portugal, 27-30 April, 2014*, pages 410–418, 2014.
2. R. Cole, F. Funke, L. Giakoumakis, W. Guy, A. Kemper, S. Krompass, H. A. Kuno, R. O. Nambiar, T. Neumann, M. Poess, K.-U. Sattler, M. Seibold, E. Simon, and F. Waas. The mixed workload ch-benchmark. In *DBTest*, page 8. ACM, 2011.
3. D. E. Difallah, A. Pavlo, C. Curino, and P. Cudr-Mauroux. Oltp-bench: An extensible testbed for benchmarking relational databases. *PVLDB*, 7(4):277–288, 2013.
4. F. Färber, N. May, W. Lehner, P. Große, I. Müller, H. Rauhe, and J. Dees. The SAP HANA database – an architecture overview. *IEEE Data Eng. Bull.*, 35(1):28–33, 2012.
5. M. Grund, J. Krueger, H. Plattner, A. Zeier, P. Cudr-Mauroux, and S. Madden. Hyrise - a main memory hybrid storage engine. *PVLDB*, 4(2):105–116, 2010.
6. H-Store Documentation: MapReduce Transactions. `http://hstore.cs.brown.edu/documentation/deployment/mapreduce/`.
7. S. Harizopoulos, D. J. Abadi, S. Madden, and M. Stonebraker. Oltp through the looking glass, and what we found there. In *SIGMOD Conference*, pages 981–992. ACM, 2008.

8. S. Idreos, F. Groffen, N. Nes, S. Manegold, K. S. Mullender, and M. L. Kersten. Monetdb: Two decades of research in column-oriented database architectures. *IEEE Data Eng. Bull.*, 35(1):40–45, 2012.
9. A. Kemper, T. Neumann, J. Finis, F. Funke, V. Leis, H. Muehe, T. Muehlbauer, and W. Roediger. Processing in the hybrid oltp & olap main-memory database system hyper. *IEEE Data Eng. Bull.*, 36(2):41–47, 2013.
10. P. Larson, C. Clinciu, C. Fraser, E. N. Hanson, M. Mokhtar, M. Nowakiewicz, V. Papadimos, S. L. Price, S. Rangarajan, R. Rusanu, and M. Saubhasik. Enhancements to SQL server column stores. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, pages 1159–1168, 2013.
11. K. T. Malladi, B. C. Lee, F. A. Nothaft, C. Kozyrakis, K. Periyathambi, and M. Horowitz. Towards energy-proportional datacenter memory with mobile dram. *SIGARCH Comput. Archit. News*, 40(3):37–48, June 2012.
12. H. Plattner. The impact of columnar in-memory databases on enterprise systems. *PVLDB*, 7(13):1722–1729, 2014.
13. V. Raman, G. K. Attaluri, R. Barber, N. Chainani, D. Kalmuk, V. KulandaiSamy, J. Leenstra, S. Lightstone, S. Liu, G. M. Lohman, T. Malkemus, R. Müller, I. Pandis, B. Schiefer, D. Sharpe, R. Sidle, A. J. Storm, and L. Zhang. DB2 with BLU acceleration: So much more than just a column store. *PVLDB*, 6(11):1080–1091, 2013.
14. A. Rowstron, D. Narayanan, A. Donnelly, G. O'Shea, and A. Douglas. Nobody ever got fired for using hadoop on a cluster. In *Proceedings of the 1st International Workshop on Hot Topics in Cloud Data Processing*, HotCDP '12, pages 2:1–2:5, New York, NY, USA, 2012. ACM.
15. J. Shute, R. Vingralek, B. Samwel, B. Handy, C. Whipkey, E. Rollins, M. Oancea, K. Littlefield, D. Menestrina, S. Ellner, J. Cieslewicz, I. Rae, T. Stancescu, and H. Apte. F1: A distributed sql database that scales. *PVLDB*, 6(11):1068–1079, 2013.
16. Sizing Guide for Single Click Configurations of Oracles MySQL on Sun Fire x86 Servers. `http://www.oracle.com/technetwork/server-storage/sun-x86/documentation/o11-133-single-click-sizing-mysql-521534.pdf`.
17. D. C. Zilio, J. Rao, S. Lightstone, G. M. Lohman, A. J. Storm, C. Garcia-Arellano, and S. Fadden. Db2 design advisor: Integrated automatic physical database design. In *VLDB*, pages 1087–1097. Morgan Kaufmann, 2004.

Appendix

7.1 Execution of CH-benCHmark Queries

The following adaptions have been done to run the CH-benCHmark queries:

– when needed, the extract function (e.g., `EXTRACT(YEAR FROM o_entry_d)`) has been replaced by the year function (e.g., `YEAR(o_entry_d)`)
– for MySQL and PostgreSQL, query 15 has been modified to use a view instead of using SQL's `having` clause (code provided by the OLTP-Bench framework)
– when needed, aliases have been resolved in case they are not supported in aggregations

We set the maximal query execution time to 12h for each query, which excludes queries from our results even though they are executable. Due to their long execution time we assume that the execution of these queries does not terminate.

7.2 TCO Calculations

The following section lists the components for an assumed bandwidth requirement of 40 GB/s. The prices have been obtained from the official websites of hardware vendors and do not include any discounts. Energy costs are calculated using the technical specifications of the hardware. Cooling costs are calculated using an assumed Power Usage Effectiveness (PUE) of 1.8 according to the Uptime Institute 2012 Data Center Survey[3]. The cost of energy is $0,276 per kWh. Both energy and cooling costs are calculated for a timespan of three years.

For the hard disk and solid state disk based systems each node is a four processor server (4x Intel Xeon E7-4850v2 12C/24T 2.3GHz 24MB) with an estimated price of $30,000. For both configurations the size of main memory is set to ~10% of the database volume (i.e., 50 GB for the 500 GB data set).

All following exemplary calculations do not include costs for high availability.

HDD-Based System The HDD-based system adapts to higher bandwidth requirements by adding direct attached storage units. In this calculation, each node has eight SAS slots. Each DAS unit is connected to two SAS slots and is assumed to provide the maximal theoretical throughput of 6 GB/s and consists of 96 disks (10K enterprise grade) to provide the bandwidth. It is possible to reach 6 GB/s with fewer 15K disks, but a configuration with 10K is more price efficient.

Since two SAS slots are used to connect each DAS unit, each server node can connect to a maximum of four DAS units resulting in a peak bandwidth of 24 GB/s. Consequently, any bandwidth higher than 24 GB/s requires an additional server node.

The hardware setup for the 40 GB/s configuration and its TCO calculation is listed in Section 7.2.

[3] Uptime Institute 2012 Data Center Survey – URL: `http://uptimeinstitute.com/2012-survey-results`

Item	*Amount*	*Est. price per item ($)*	*Total ($)*
Server	2	30,000	60,000
DAS unit	7	4,500	31,500
SAS controller	14	500	7,000
Hard disk (10K)	672	150	100,800
Main Memory (16 GB)	4	340	1,360
Energy	-	-	143,110
Cooling	-	-	114,488
SSD for logging	1	5,000	5,000
TCO			**463,218**

Table 2. TCO Calculation for the HDD-based System

SSD-Based System The SSD-based system uses PCI-e connected solid state disks. Recent Intel Xeon CPUs have up to 32 PCI-e lanes per socket that are directly connected. Consequently, we assume a theoretical setup of up to eight PCIe-connected SSDs per server node.

For our calculations, we use an PCIe SSD that provide a peak read bandwidth of 3 GB/s and has a size of 1 TB. As of now, there are faster SSDs available (up to 6 GB/s), but these are more expensive by a factor of over 3x. We also calculated prices for another PCIe SSD vendor whose drives are almost a factor 2x less expensive in their smallest size of 350 GB. We did not include these calculations here, as these drives are currently not available.. However, even using these drives the 40 GB/s configuration is still more expensive than its main memory-based counterpart.

Item	*Amount*	*Est. price per item ($)*	*Total ($)*
Server	2	30,000	60,000
PCIe-connected SSD (3 GB/s)	14	13,100	183,400
Main Memory (16 GB)	4	340	1,360
Energy	-	-	34,240
Cooling	-	-	27,218
SSD for logging	1	5,000	5,000
TCO			**311,218**

Table 3. TCO Calculation for the SSD-based System

Main Memory-Based System The main memory-based server is equipped with Intel's latest XEON E7 CPU. A server with four CPUs (Intel Xeon E7-4890v2

15C/30T 2.8GHz 37MB) costs ~$63,000. The costs include a 600 GB enterprise-grade HDD for persistence.

Item	*Amount*	*Est. price per item ($)*	*Total ($)*
Server	1	63,000	63,000
Hard disk (15K)	1	300	150
Main Memory (16 GB)	63	340	21,420
Energy	-	-	25,304
Cooling	-	-	20,242
SSD for logging	-	-	5,000
TCO			**135,116**

Table 4. TCO Calculation for the Main Memory-based System