Towards the Prediction of the Performance and Energy Efficiency of Distributed Data Management Systems

Raik Niemann
Institute of Information Systems
University of Applied Science Hof, Germany
raik.niemann@iisys.de

ABSTRACT

The ability to accurately simulate and predict the metrics (e.g. performance and energy consumption) of data management systems offers several benefits. It can save investments in both time and hardware. A prominent example is the resource planning. Given a specific use case, a datacenter operator is able to find the most performant or most energy efficient configuration without performing benchmarks or aquiring the necessary hardware. Another possibility would be to study the effects of architectural changes without having them implemented.

In this paper, *Queued Petri Nets* were used to predict and to study the performance and energy consumption of a distributed data management system like *Cassandra*. The prediction accuracy was evaluated and compared to actual experimental results. On average, the predicted and experimental results differ only by 8 percent for the performance and 16 percent for the energy efficiency, respectively.

In addition to this, the experimental results of the used *Cassandra* cluster revealed a super-linear behavior for the performance and a sub-linear one for the energy consumption.

1. INTRODUCTION

Choosing the appropriate data management system (DMS) for a given use case depends on several factors such as existing hardware and licenses as well as performance and energy consumption indicators. Along with decision support, those factors also apply when it comes to resource planning. This is the case when the use case or the platform has changed in order to optimize the performance [5, 12]. However, evaluating the optimization gain by performing regression tests or benchmarks requires an important investment in time and hardware. Simulating a DMS could be a good approach to save those investments, for example by observing how horizontal or vertical scale-up scenarios affect the DMS performance. To achieve this, the simulation has to predict major metrics, for example the response time, performance, energy consumption and efficiency. For useful findings, the difference between simulated and real results has to be small. The simulation has also to include other platform components like the surrounding operating system and its communication network.

In [11], *Queued Petri Nets* (QPN) models were introduced that fully satisfy these requirements. They were used to simulate four *Yahoo Cloud Serving Benchmark* (YCSB) workloads running on top of the distributed DMS *Cassandra*. To prove and to evaluate the models, real experiments were performed on a *Cassandra* cluster consisting of 3 up to 7 nodes. The experimental and simulated results were compared to calculate the difference between the experimental and simulated workload response time and energy consumption, respectively. The difference for the workload response time was on average 21.07 percent. Regarding the energy consumption, the difference was 46.25 percent on average.

The short simulation execution times of 3 minutes and 46 seconds on average allowed to simulate the workload response time for *Cassandra* clusters beyond the test apparatus, i.e. for clusters having more than 7 nodes. This allows further findings without having investments in hardware. One finding was the optimal number of *Cassandra* cluster nodes with respect to the tested YCSB workloads. However, the capacity of the used bladecenter in terms of blade nodes was limited so this finding could not be validated.

In order to confirm the prediction of the optimal number of *Cassandra* cluster nodes, further experiments with an increased number of nodes were performed. Thanks to multiple optimizations and fine tuning of the models, the difference between the simulated and experimental results could be reduced.

The remaining paper is organized as follows: Section 2 gives a brief introduction into the QPN models. The test methodology and apparatus as well as their adjustments are described in Section 3. This section also briefly depicts the setup of the QPN models. The experimental and simulation results are compared in Section 4. Section 5 gives a conclusion about all observations.

2. RELATED WORK

In [10] and [11], *Queued Petri Nets*[1] models were introduced. The models are intended to reflect the data flow of a DMS in the form of marks among all components that could have

[1] A QPN combines *Colored Petri Nets*, *Generalized Stochastic Petri Nets* and adds queuing principles as well as scheduling strategies to the places. The formal definition for a QPN is fairly extensive and can be found in [6].

ICPE'16 Companion, March 12-18, 2016, Delft,Netherlands

DOI: http://dx.doi.org/10.1145/2859889.2859891

an impact on the performance. This includes the surrounding operating system, technical components (main memory, mass storage and CPU as visible in Figure 1) and the common components of a data management system as illustrated in Figure 2.

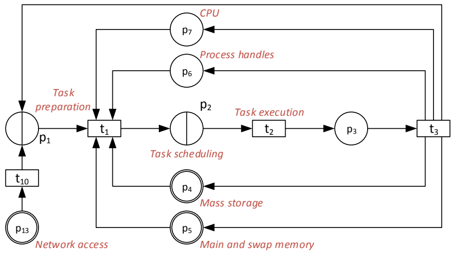

Figure 1: QPN model for a generic surrounding operating system

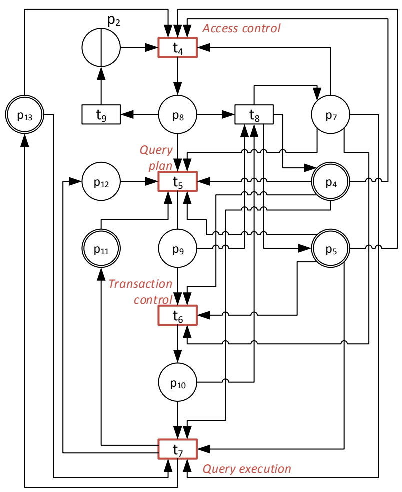

Figure 2: QPN model for a generic data management system

The models are also prepared to form a network. This enables to model the data flow of data management systems that are distributed over several independent servers, i.e. a database cluster. To achieve this, an additional QPN model is required that reflects the communication among the cluster nodes. Basically this model allows the exchange of marks and submarks within all incorporated models. An example of such a model is shown in Figure 4.

In essence, the models are able to simulate and to study the effects of both vertical and horizontal scale-out scenarios. They allow to observe the three main use cases of data management systems: data manipulation, data querying and the handling of multiple clients. In addition to this, all forms of communication topologies (bus, tree or star) as well as their characteristics can be modelled, for example throughput, latencies and performance metrics. The simulation runs are capable of predicting the performance metrics of a distributed DMS, the surrounding operating system and the communication network.

The models were evaluated and proven by simulating prominent database benchmarks (*TPC-H*, *StarSchema Benchmark* and *YCSB*). The simulated workload response times, energy consumption and efficiency values were compared with the experimental ones.

For all evaluations, the difference between them ranged from 23 to 46 percent on average[2]. In general the discrepancy between the simulated and experimental workload response times is smaller than the difference between the ones for the energy consumption.

3. EVALUATION SETUP

3.1 Test Methodology

To get comparable experimental results with the previous ones in [10], the same test methodology was used. In particular, four different YCSB workloads were performed to benchmark *Cassandra* in its enterprise version (*DataStax Enterprise Server*). The four workloads

1. `load`: the YCSB client executes 350 million insert operations. Each operation inserts one table row with 1 KByte of data. The resulting 350 GByte of data are equally distributed among all incorporated *Cassandra* nodes.

2. `read`: the YCSB client reads the previously inserted data. This workload executes 10 million read operations using the *uniform* distribution[3]. In total 10 out of 350 GByte of data are read.

3. `write`: the YCSB client executes 10 million write operations, i.e. 10 million existing table rows are modified. This represents 10 GByte out of 350 GByte of data to be modified.

4. `mixed`: the YCSB client executes at first 5 million read and then 5 million write operations. This represents 5 GByte of data to be read and 5 GByte to be modified, respectively.

were performed on a subsequent number of *Cassandra* nodes ranging from 3 to 13 nodes. In total 132 different YCSB experiments (4 workloads × 11 cluster sizes × 3 repetitions) were executed.

3.2 Test Apparatus

The test apparatus that was used to get the previous experimental results in [10] was modified to support the increased number of *Cassandra* cluster nodes. Previously only one *Fujitsu BX600 S3* bladecenter was sufficient to manage the required number of nodes. With respect to the objectives of this paper and due to the bladecenter limitations, an additional bladecenter was added to handle the increased number of cluster nodes. Table 1 shows the hard- and software characteristics.

Figure 3 illustrates the test apparatus in general. Bladecenter node B1 was primarily used to execute the YCSB benchmark as the client to access the *Cassandra* cluster. The other nodes of the bladecenter were used to create a *Cassandra* cluster with a variable number of participating nodes ranging from 3 to 13 nodes. The two hard drives of every node were combined to a striping RAID-0 array.

[2]Please note that to the best of the author's knowledge there is no other QPN model that is able to simulate the performance and energy consumption for the tested benchmarks. Therefore a comparison is infeasible.

[3]According to [3], the table rows are chosen at random and they have the same likelihood to be chosen.

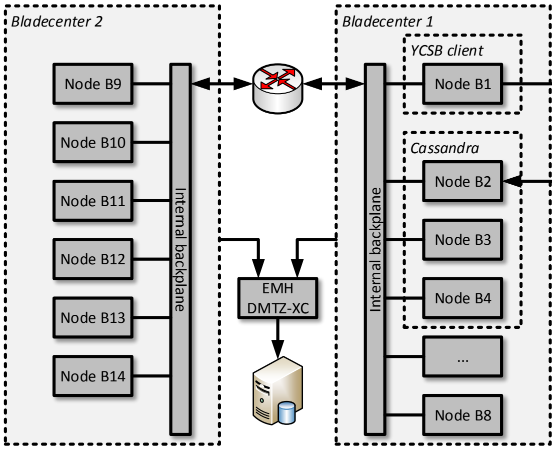

Figure 3: Modified test apparatus with two bladecenters to execute a *Cassandra* cluster with up to 13 cluster nodes (an exemplary cluster with 3 nodes is shown)

Table 1: Hard- and software characteristics

Component	Node B1	Node B2 to B14
CPU	2×AMD Opteron 890	2×AMD Opteron 870
Main memory	32 GByte DDR-2 reg.	16 GByte DDR-2 reg.
Hard drives	2×300 GByte	2×146 GByte
Operating system	*Ubuntu Server* 12.04 LTS 64 bit	
JRE	*Oracle* JRE 1.7.0.60-b19	
Cassandra cluster	*DataStax Enterprise Server* 4.5.0	

The blade nodes within a bladecenter used the internal backplane as the power source and to access the data network with a rate of 1 GBit/s. To connect both used bladecenters for data exchange, a *Linksys SRW2048* switch was used. Since both bladecenters offered the feature to combine multiple network ports to increase the transmission bandwidth, a rate of 6 GBit/s was enabled.

Regarding the measurement of the energy consumption, the same calibrated *EMH DMTZ-XC* measurement device was used as in the previous test apparatus. The measurement device has an inaccuracy of ± 1 percent and measured the overall energy consumption of both bladecenters throughout all experiments. Because the internal power connectors of the bladecenter were proprietary and no documentation was published, there were no other options. As before, the measurement device was queried every 15 seconds by an external server that stores the energy consumption values in a database. Lower query periods were impossible due to the query interface of the measurement device which has only a slow data transfer rate. The time clocks of the bladecenter and the external server were periodically synchronized to ensure identical timestamps for the experiments.

During all experiments, major characteristics of the incorporated blade nodes were recorded into log files, for example the usage of the CPU and mass storage devices. This also includes the fraction of free and used main memory.

In contrast to the previous test apparatus, the administrative functionality was not restricted. As a result, all blade nodes that were not required to execute a workload were shut down. The required ones were restarted after each repetition of a workload to prevent side effects such as using caches. This does not apply for the two bladecenter and the switch.

3.3 Simulation Setup and Execution

To simulate the real experiments according to the test apparatus and methodology, the QPN models described in Section 2 were used. To reflect the modified test apparatus, the QPN model shown in Figure 4 was created and used as the top tier model for all simulations. The place B_1 with its inner model for the surrounding operating system was used to simulate the YCSB client. The remaining places B_2 to B_{14} represent the *Cassandra* cluster nodes. This places also contain particular instances of the inner models for the surrounding operating system and the data management system. The places BP_1 and BP_2 represent the backplane of the two bladecenters and do not contain an inner model because no special network topology was used and therefore does not need to be modeled. Place Sw represents the switch that connects both backplanes (places BP_1 and BP_2) with each other and does not have an inner model, too.

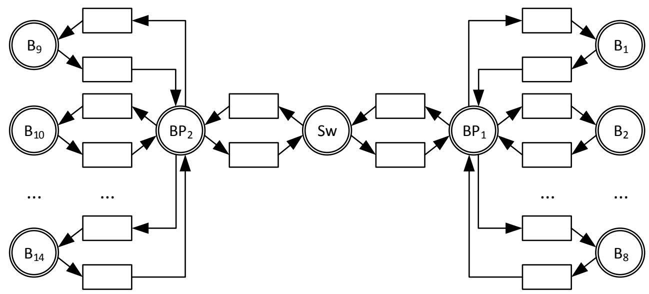

Figure 4: QPN model to simulate the test apparatus as shown in Figure 3. Queued place *Sw* represents the used switch to connect both internal backplanes of the bladecenters (queued places BP_1 and BP_2). All shown places have an up- and download path.

The transition times and queue sizes for the top tier QPN model were set up according to the test apparatus. By assuming a transitioned mark reflects 1.500 Byte of transmissioned data, a transmission bandwidth of 1 GBit/s corresponds to 87 marks per simulated second (1 GBit ÷ 1500 Byte = 87). This way, the queues for the places B_1 to B_{14} were set up to hold 96 marks (10 percent backlog). For the places BP_1 and BP_2 this queue size was set up to store 768 marks. The reason was the fact that the real backplanes are potent enough to ensure the transmission bandwidth for every node. For the place Sw the queue size was set to 576 (6 GBit/s transmission bandwidth plus 10 percent backlog). The transition shown in Figure 4 for the up- and download path were adjusted to fully support the full duplex transmission mode. This means that the transitions times for the places BP_1 and BP_2 were set to $1/87$, so 87 marks per simulated second can transition. The transitions for the places places BP_1, BP_2 and Sw were set up adequately ($1/696$ and $1/522$ respectively).

With respect to the inner models of the places B_1 to B_{14} (surrounding operating system and data management system), the same transition rules and transition times as for the previous test apparatus in [11] were used as a basis for optimization. Especially the transition rules got a fine tuning while the semantics of them stayed the same. As a re-

Table 2: Characteristics of the simulation server

Component	Description
CPU	4×*AMD Opteron 6380*, 16 cores per CPU with 2.5 GHz per core
Main memory	1.5 TByte (48×32 GByte, DDR3, ECC)
Operating system	*Ubuntu server* 15.04 64bit
JVM	*Oracle Java SE* 7u71

sult, the number of mark sets was reduced. Compared to the original ones, this allowed to reduce the complexity significantly. Basically the input of the models are mark sets that reflect the YCSB workloads in the inner model of place B_1 (simulated YCSB client). According to the transition rules and times those marks are processed involving all other models.

The simulation outputs are major stochastical values of the places, like the throughput of marks per simulated second or the minimum and maximum population of marks in the places.

All performed simulations were done using the same simulation server as for the evaluation of the former models described in [10] and [11]. The hardware characteristics are shown in Table 2. The top tier QPN model depicted in Figure 4 was designed and simulated with the QPN simulator *QPME* [7] version 2.

4. RESULT EVALUATION

4.1 Scalability and Amdahl's Law

The evaluation of the experimental results revealed a super-linear behavior for the YCSB workload performance and surprisingly a sub-linear one for the energy consumption. In the context of this paper, the speedup s for two different tested cluster sizes n and m are defined as

$$s_p(n,m) = \frac{Perf_n}{Perf_m} \qquad 3 \leq n < m \leq 13$$

for the performance[4] $Perf$ and

$$s_w(n,m) = \frac{W_m}{W_n} \qquad 3 \leq n < m \leq 13$$

for the electrical work W, respectively. $Perf_n$ and $Perf_m$ denote the YCSB workload performance for the tested *Cassandra* cluster size. Analogous, W_n and W_m denote the electrical work that the *Cassandra* cluster consumed to perform a YCSB workload. The optimal speedup for the performance is linear, i.e. $s_p(n,m) = {}^{n}/_{m}$. Any value below ${}^{n}/_{m}$ is sub-linear and any value above is super-linear, respectively. For example, doubling the number of *Cassandra* nodes from $n = 3$ to $m = 6$ would double the performance but would also double the energy consumption. *Amdahl's law* [1] states that the maximum speedup is linear and in reality rather sub-linear but the experimental results showed the opposite behavior.

First, *Amdahl's law* attracted a lot of discussion because it does not consider parallelization in clusters as well as caches. The evaluation of the experimental results showed for all tested YCSB workload a super-linear performance speedup. Using the example above, the performance was more than the double when the number of incorporated *Cassandra* cluster nodes was doubled. This also means that the time t it tooks to execute a YCSB workload was less than the half when doubling the number of cluster nodes. However, the difference between the linear and the super-linear speedup values are small and ranges between 2 and 8 percent. It was also observable that the speedup differences for the write intensive YCSB workloads `load` and `write` are higher than the ones for the read intensive workloads `read` and `mixed`. A deeper investigation of the log files that the YCSB client generates during a workload revealed an excessive use of caches. Together with the small cover rate[5] this explains the super-linear performance. This finding also corresponds to the observations of the test apparatus described in [9] where a cluster of up to 16 nodes was tested. The experiments running the *TPC-H* benchmark on top of a *Vertica* cluster also showed a super-linear performance due to cache effects.

Second, *Amdahl's law* does not directly consider the effects that relate to the energy consumption but have an effect on the performance, for example forced energy saving mechanisms like CPU frequency scaling [2]. In theory the optimal speedup for the energy consumption is also linear. However, the reality shows that it is rather super-linear. In contrast to this, the evaluation of the experimental YCSB results for the energy consumption showed a sub-linear behavior. This means that the energy consumption was less than the double when doubling the number of *Cassandra* cluster nodes for the same YCSB workload. The analysis of the log files that recorded the utilization of the cluster nodes during the experiments showed different power usage levels. In other words, the power consumption is not proportional to the utilization. In theory the power consumption increases linearly with the utilization. As visible in Figure 5, the analysis showed that between 52 and 76 percent of the utilization the power consumption was below this linearity. This means that the used cluster nodes are most efficient in this segment. It was also observed that the higher the number of incorporated *Cassandra* cluster nodes was, the lesser they were utilized. The reason is that the same YCSB workload was distributed over a subsequently increased *Cassandra* cluster size. In return the load for the incorporated cluster nodes decreases. As a result, the higher the number of cluster nodes the more were shifted into the utilization segment mentioned above. In summary, the super-linear performance speedup in conjunction with the under-utilized cluster nodes explains the sub-linear speedup for the energy consumption.

Third, the super-linear behavior of the performance and the sub-linear one for energy consumption has a dramatic consequence on the energy efficiency. The energy efficiency EE is the division of the performance $Perf$ and the mean electrical power P:

$$EE = \frac{W_{\text{YCSB}} \cdot t}{W_E \cdot t} = \frac{Perf}{P}$$

[4] Note that the performance is the quotient between the performed YCSB operations and the time t it tooks. The unit in this case is *ops* per second. For the `load` workload, this is 350.000.000 $ops \div t$. For the remaining workloads `read`, `write` and `mixed`, the performance is 10.000.000 $ops \div t$.

[5] The cover rate is a percentage to illustrate the affection. In the context of this paper the cover rate is the ratio between the number of operations and the overall number of operations. The cover rate is always 10 million÷350 million×100 = 2.8 percent.

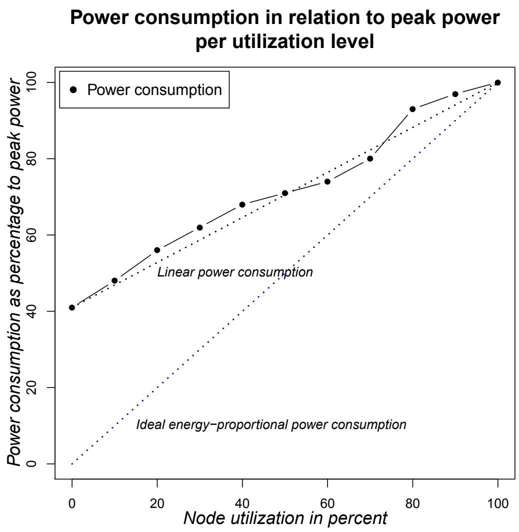

Figure 5: Power consumption of a cluster node in relation to the peak consumption for different utilization rates

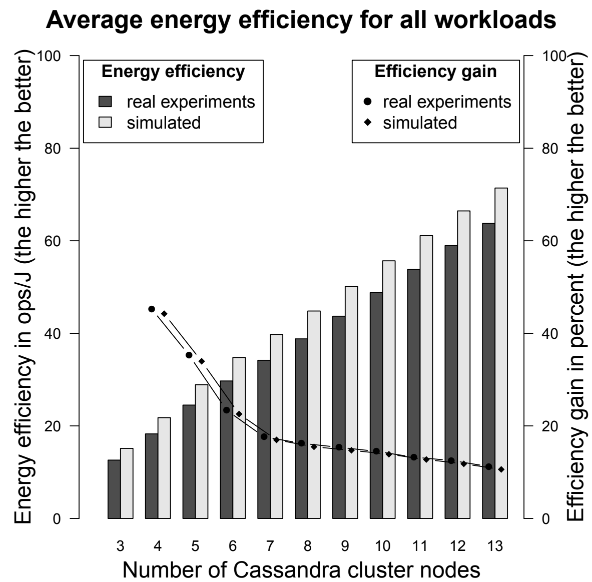

Figure 6: Average energy efficiencies for all YCSB workloads. The bars indicate the energy efficiency for the real and simulated workloads. The lines show the energy efficiency gain for subsequently increased cluster sizes.

This equation implies that optimizing the energy efficiency has two different goals. Either to optimize for the performance at the expense of the energy consumption or vice versa. In contrast to this, the evaluation of the experimental results showed that that both optimization goals are achieved at the same time. Figure 6 illustrates this fact. By considering only the energy efficiency values for the experiments (the dark bars), the energy efficiency rises with a subsequent number of incorporated *Cassandra* nodes. In other words, it is more efficient to perform the YCSB workloads with a higher number of cluster nodes even though they consume more energy. However, there is an observable tradeoff. Figure 6 shows the energy efficiency gain as a percentage for every subsequently added *Cassandra* cluster node (the line with the black dots). It can be seen that this curve is degrading. Beginning with a *Cassandra* cluster consisting of 11 nodes the energy efficiency gain is under 10 percent and decreasing. As a result the optimal *Cassandra* cluster size is 10 nodes.

4.2 Simulation accuracies

The experimental results were compared with the simulated ones to get the accuracies for the performed workloads. These accuracies are expressed as a percentage: 100 percent indicates that both the experimental and simulated workload response times or energy consumption do not differ. A percentage of zero percent indicates the fact that the simulated workload response time or energy consumption differs significantly from the experimental one, i.e. more than the double.

Table 3 shows the accuracies for all performed YCSB workloads. The last column labeled "∅" of this table show the average value per table row, i.e. the average accuracy for all tested *Cassandra* cluster sizes.

First, the accuracies for the response time are higher than the ones of the energy consumption. It is also noticeable that the accuracies rise with a higher number of *Cassandra* cluster nodes. The reason is that the input data of the simulations rises with every simulated cluster node. This results in a better precision of the calculations. In addition to this, the average accuracy for the performance for all tested YCSB workloads increased remarkably from 79.93 percent to 91.73 percent. The reasons were the improved transition rules and times that fitted more precisely to the used test methodology. It is also noticeable that the accuracies are higher for YCSB workloads involving only write operations, i.e. `load` and `write` vs. `read` and `mixed`. On average they differ by 3.28 percent. This conforms to other experimental results for *Cassandra* which is known for this behavior [3, 4, 8]. The simulations and the experimental results fully cover this fact.

Additionally, the statistics that were recorded during the simulations runs were evaluated and compared with the ones from the experiments. The comparison shows that *Cassandra*'s architecture in terms of cache utilization can be successfully simulated.

Second, the accuracies for the energy consumption showed the same behavior as previously described for the response time. This means that the accuracies also rise for *Cassandra* clusters with a higher number of incorporated nodes. Please note, that the simulated energy consumption is a calculated value and not a measured one. Based on the technical documentation and manuals of the simulated technical components (mass storage, main and swap memory, CPU, network), an energy consumption value in Watt per transitioned mark was calculated. This value multiplied by the recorded mean throughput rates for the resource places allowed to calculate the energy consumption for these components, which in sum is the overall energy consumption.

Based on this method, the calculated accuracies for the energy consumption were smaller than the ones for the performance. On average the accuracy for the energy consumption for all tested YCSB workloads is 76.79 percent. Compared to the previous one in [11], there is an increase of

Table 3: Accuracies for the response time and energy consumption in percent

		←				*Cassandra cluster size*					→	
Metric	**Workload**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**∅**
Response time	`load`	89.12	90.04	91.67	92.72	93.46	94.11	94.89	95.53	96.12	97.02	97.86	93.87
	`read`	85.81	86.75	87.93	88.87	89.56	90.07	90.84	91.47	92.18	93.25	93.98	90.06
	`write`	88.59	89.61	90.48	91.22	92.37	93.09	93.96	94.59	95.12	96.05	96.73	92.89
	`mixed`	87.95	88.46	88.89	89.26	89.84	90.20	90.63	90.95	91.21	91.68	92.02	90.10
Energy consumption	`load`	77.19	78.02	79.76	80.85	81.59	82.34	83.11	83.98	84.52	85.23	86.08	82.06
	`read`	65.44	66.58	67.61	68.33	68.89	70.51	71.23	71.91	72.67	73.62	74.28	70.09
	`write`	75.02	75.95	77.73	78.75	79.44	80.29	81.01	81.96	82.42	83.11	83.97	79.96
	`mixed`	70.23	71.26	72.67	73.54	74.16	75.40	76.12	76.93	77.54	78.36	79.12	75.03

nearly 23 percent. The reason were both the modified test apparatus and the more precise experimental results.
Third, a comparison of the energy efficiency values shows that the simulated ones are smaller that the experimental ones as visible in Figure 6. The difference is 16 percent on average. As mentioned above, Figure 6 also illustrates the energy efficiency gain as a percentage when the same YCSB workload was performed for a subsequently increased *Cassandra* cluster size. There is a small difference between the simulated and experimental percentages visible. This difference decreases with a higher number of *Cassandra* cluster nodes. On average this difference is 2.87 percent.

5. CONCLUSION AND FUTURE WORK

First, the evaluation of the experimental results revealed a super-linear behavior for the performance and a sub-linear one for the energy efficiency. This means that the performance was more than the double when doubling the number of *Cassandra* cluster nodes. Analogous, the energy consumption was less than the double when doubling the *Cassandra* cluster size. As a result, the energy efficiency rose with a higher number of nodes but its gain decreased with every subsequently added cluster node. Beginning with a size of 11 cluster nodes, this gain fells under 10 percent compared to a cluster size of 10 nodes.

Second, the experimental results fully prove the prediction of the optimal *Cassandra* cluster size with respect to the energy efficiency and the tested YCSB workloads. The simulation runs fully confirm the super-linear performance behavior that was observed in the experimental results. In contrast to this, the sub-linear behavior of the energy consumption was not completely reproduced.

Third, in the context of this paper the transition rules and transition times within the QPN models were optimized and rearranged while keeping the semantics intact. Together with a more precise test apparatus, the average accuracies for the performance and energy efficiency were drastically improved. Compared to the previous accuracies in [11], they rose by nearly 12 percent to 92 percent for the performance and by nearly 30 percent to 84 percent for the energy efficiency, respectively.

In conclusion, simulation runs using the QPN models showed that they are capable of predicting important metrics with a good accuracy. In contrast to traditional ways, they save investments in both time and hardware. One example is to study the effects of architectural changes for different hardware platforms without actually implementing these changes and to aquire the necessary hardware. Another one is resource planning where operators of datacenters use simulation runs to find the most performant or most energy efficient configuration for a given use case without having investments in those hardware.
In the future it is planned to measure the accuracies of other distributed data management systems using different benchmarks in order to demonstrate the benefits of the approach described above.

6. REFERENCES

[1] G. M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference*, pages 483–485. ACM, 1967.

[2] S. Cho and R. G. Melhem. Corollaries to amdahl's law for energy. *Computer Architecture Letters*, 7(1):25–28, 2008.

[3] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears. Benchmarking cloud serving systems with ycsb. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 143–154. ACM, 2010.

[4] T. Ivanov, R. Niemann, S. Izberovic, M. Rosselli, K. Tolle, and R. Zicari. Performance Evaluation of Enterprise Big Data Platforms with HiBench. In *9th IEEE International Conference on Big Data Science and Engineering*, Helsinki, 2015. IEEE.

[5] J. Klein, I. Gorton, N. Ernst, P. Donohoe, K. Pham, and C. Matser. Performance evaluation of nosql databases: A case study. In *Proceedings of the 1st Workshop on Performance Analysis of Big Data Systems*, pages 5–10. ACM, 2015.

[6] S. Kounev and A. Buchmann. Performance modelling of distributed e-business applications using queuing petri nets. In *Performance Analysis of Systems and Software, 2003. ISPASS. 2003 IEEE International Symposium on*, pages 143–155. IEEE, 2003.

[7] S. Kounev and C. Dutz. Qpme - a performance modeling tool based on queueing petri nets. *ACM SIGMETRICS Performance Evaluation Review (PER), Special Issue on Tools for Computer Performance Modeling and Reliability Analysis*, 36:46–51, 3 2009.

[8] J. Kuhlenkamp, M. Klems, and O. Röss. Benchmarking scalability and elasticity of distributed database systems. *Proceedings of the VLDB Endowment*, 7(13), 2014.

[9] W. Lang, S. Harizopoulos, J. M. Patel, M. A. Shah, and D. Tsirogiannis. Towards energy-efficient database cluster design. *Proceedings of the VLDB Endowment*, 5(11):1684–1695, 2012.

[10] R. Niemann. Evaluating the performance and energy consumption of distributed data management systems. In *Global Software Engineering Workshops (ICGSEW)*, pages 27–34. IEEE, 2015.

[11] R. Niemann and T. Ivanov. Evaluating the energy efficiency of data management systems. In *Proceedings of the Fourth International Workshop on Green and Sustainable Software*, pages 22–28. IEEE Press, 2015.

[12] R. Niemann and T. Ivanov. Modelling the Performance, Energy Consumption and Efficiency of Data Management Systems. In *Big Data, Smart Data and Semantic Technologies INFORMATIK 2015*. GI, 2015.