Wimpy Node Clusters: What About Non-Wimpy Workloads?

Willis Lang
University of Wisconsin
wlang@cs.wisc.edu

Jignesh M. Patel
University of Wisconsin
jignesh@cs.wisc.edu

Srinath Shankar
Microsoft Corp.
srinaths@microsoft.com

ABSTRACT

The high cost associated with powering servers has introduced new challenges in improving the energy efficiency of clusters running data processing jobs. Traditional high-performance servers are largely energy inefficient due to various factors such as the over-provisioning of resources. The increasing trend to replace traditional high- performance server nodes with low-power low-end nodes in clusters has recently been touted as a solution to the cluster energy problem. However, the key tacit assumption that drives such a solution is that the proportional scale-out of such low-power cluster nodes results in constant scaleup in performance. This paper studies the validity of such an assumption using measured price and performance results from a low-power Atom-based node and a traditional Xeon-based server and a number of published parallel scaleup results. Our results show that in most cases, computationally complex queries exhibit disproportionate scaleup characteristics which potentially makes scale-out with low-end nodes an expensive and lower performance solution.

1. INTRODUCTION

Datacenter deployment is a big investment for any enterprise. Facilities often cost hundreds of millions of dollars while datacenters are populated with many thousands of servers. Such datacenter costs are ultimately factored into the company bottom line through the monthly amortized costs. The largest proportions of this monthly total cost of ownership (TCO) are the server costs, power distribution and cooling, and the actual energy costs. Hamilton states that the amortized (3 year server, 10 year infrastructure) monthly costs for a 50,000 machine datacenter breaks down to 54% for servers, 21% for power distribution and cooling, 13% for energy, and the balance for remaining networking and infrastructure costs. Thus, energy and energy related costs can account for a third of the monthly TCO, and several studies have shown that these costs are projected to increase as a proportion of the monthly TCO [6, 5]. Consequently, there has been expanding interest in reducing datacenter power costs [4] (see Section 4).

In this paper, we study the price/performance characteristics of parallel scale-out clusters where our notion of price includes server costs *and* direct energy costs thereby accounting for 67% of the monthly TCO.

To reduce the monthly TCO, a growing number of recent studies have focused on redesigning datacenter server clusters with low-cost, low-power "wimpy" nodes [3, 23]. The argument for wimpy nodes is that they are relatively well-balanced [24]. With low-end CPUs and the use of low-power components, these nodes are claimed to be more energy-efficient. However, low-end nodes lag far behind traditional nodes in performance. Therefore, a small cluster of traditional nodes must be replaced by a larger cluster of low-end nodes.

These previous studies have generally come to the conclusion that a scale-out deployment of large clusters of "wimpy" nodes is the most effective solution. However, these published results focused on simple key-value workloads [3] and web-search environments [23], where near-perfect performance scaleup complements the poor performance of individual wimpy nodes. In [3], the authors admit that their approach targets *"data-intensive, computationally simple applications"*. Consequently, their system is described as a *datastore* instead of a *database* to emphasize that they do not provide transactional and relational interfaces. Such simple data lookup environments lack the complexities of more complex data processing workloads, and the focus of this paper is to consider if the same conclusions apply for complex data processing workloads.

Recently, proponents of wimpy node clusters theorize that such architectures will also be able to handle more complex workloads such as sorting [32]. However, these arguments are based on single node, performance/Watt comparisons of Atom and Nehalem-based servers. Without the analysis of how performance is affected with increasing scale-out, which is the focus of this paper, such arguments are largely based on theoretical ideal performance. The purpose of this paper is to show that for complex database workloads, previously observed results show that parallel data processing overhead is significant enough to dilute the benefits of large wimpy node clusters.

There are two factors that come into play when evaluating the efficiency of traditional versus low-end cluster deployments: (1) The individual node price/performance when processing the partitioned data; and (2) The effects of diminishing returns when undergoing *parallel scaleup* (constant response time when the datasize and computing resources increase proportionally) due to startup-interference-skew factors (see Section 2 for further discussion).

Consider Figure 1(a), where our price and performance metrics are plotted for various types of nodes and different TPC-H workloads on a commercial DBMS. The amortized monthly TCO is calculated as the sum of the amortized node cost as well as the monthly energy costs for the node continuously running TPC-H (see Sec-

Proceedings of the Sixth International Workshop on Data Management on New Hardware (DaMoN 2010), June 7, 2010, Indianapolis, Indiana.

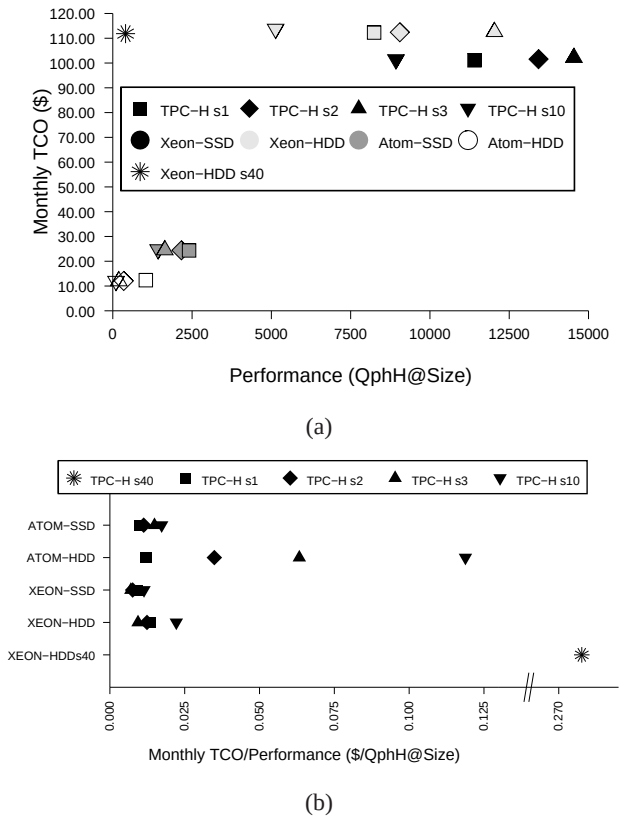


Figure 1: (a) Amortized Monthly TCO (includes energy costs) as a function of Performance over various hardware configurations and TPC-H scale factors. (b) Price/Performance metric of Amortized Monthly TCO ($) and Performance (QphH@Size) System details can be found in Section 3.1. QphH@Size is the unit for the TPC-H Power Test.

tion 3). (Here, for the node cost, we have simply divided the purchase cost over 36 months. More complex interest-based amortization can also be applied.) Performance is provided by the TPC-H Power Test [30]. 'Atom' and 'Xeon' represent Atom (low-end node) and Xeon (traditional node) processors respectively. Both types of nodes were then outfitted with either SSD or mechanical HDD disks (see Section 3.1 for detailed node specs). Figure 1(a) plots costs versus performance for different node configurations and TPC-H scale factors. There are two interesting patterns to observe. First, SSDs provide better performance than regular disks. Second, the Atom nodes lag far behind the Xeon nodes in performance, which means that we must deploy significantly more Atom nodes in a cluster to equal the performance of a Xeon cluster.

We combined the two metrics in Figure 1(a) into a single price/performance metric as shown in Figure 1(b). When we consider the more expensive SSD configurations of the Atom and Xeon, we notice that all their results for various TPC-H scale factors are tightly clustered together at the cheapest end of the price/performance spectrum. Even though the purchase cost for the systems increase with SSDs, the performance increase outpaces the cost increase and so we see better price/performance (similarly seen in [2]). This suggests that if we partition the TPC-H workload and use multiple Atom nodes, we can achieve similar performance as a Xeon node at a similar price point (assuming perfect parallel scaleup).

For example, assuming ideal scaleup, we could run a TPC-H scale 10 workload partitioned on 5 Atom-SSD nodes (i.e., 2GB per node) in the same time as a scale 2 workload on a single Atom-SSD node. This can also be done with even smaller partition sizes per Atom-SSD node, thereby changing the cluster size. If we used a 1GB partition size, we would need 10 Atom nodes. Since the measured performance for TPC-H scale 1 on the Atom is greater than 1000 QphH, ideal scaleup would infer a cluster performance greater than 10000 QphH. This would outperform a single Xeon-SSD running TPC-H scale 10 (9000 QphH).

The problem is that parallel database research has already shown two decades ago that such ideal proportional scaleup is far from guaranteed [9]. We show published examples of deviations in Section 2, Figure 2. Essentially, replacing traditional clusters with increasing numbers of low-power nodes may result in diminishing returns. Further, such scaleup profiles are largely determined by the query being run and the parallel processing system.

Therefore, while we have shown real price/performance results for the traditional server versus modern low-power wimpy node (detailed results in Section 3.3- 3.4), the interesting problem that we focus on in this paper is how these results fit into various scaleup models. Since the parallel scaleup for a given workload is dependent on the parallel software system and node hardware configuration, we need to examine the effects of different scaleup models on a low-power cluster versus a traditional cluster. To this end, we will present several real scaleup profiles from various published parallelized data processing systems (see Section 2) and use our results in Figure 1 to show that traditional server clusters may be more cost effective than a massively scaled-out wimpy node clusters.

The remainder of this paper is organized as follows: Section 2 reviews the fundamental goals, metrics, and pitfalls in parallelizing queries in a parallel DBMS, we present our results in Section 3. Section 4 discusses related work, and we present our conclusions in Section 5.

2. PARALLEL DATABASES AND SCALEUP

This section recalls the lessons learned from more than two decades of parallel database research. Specifically, we discuss the factors that impact the ability to achieve ideal parallel scaleup when deploying larger clusters.

To start, we clarify the parallelism goals that often get misused with the overloaded term cluster "scale-out". Often, scale-out is used as a blanket term for increasing the size of a parallel data processing cluster to achieve ideal performance benefits. However, as defined in [9], this idea can be divided into two distinct goals: linear **speedup** and linear **scaleup**.

For example, given 100 machines processing 1TB of data in 1min, if the parallel system has the ideal linear *speedup* property, 400 machines could process the same 1TB in 15sec. On the other hand, given the same cluster nodes, a parallel system exhibiting ideal linear *scaleup* could process 10TB with 1000 machines in 1min. Scale-up can be further broken up into transactional or batch which essentially describe throughput or latency-based definitions of performance respectively.

DeWitt and Gray [9] identified three main threats to successful scaleup behavior of a parallel DBMS: startup, interference, and skew. **Startup** costs refer to overhead in time needed to start the parallel processing jobs. For example, if synchronization is required across hundreds of nodes for starting a short query, then this cost can make up a significant fraction of the total response time. The impact of such costs often diminishes with long running queries. **Interference** costs are those caused by processes that need to share resources such as memory or disk. Finally, the last impediment of **skew** refers to the behavior that with increased parallelism and decreasing per node computation time, the the variance of node

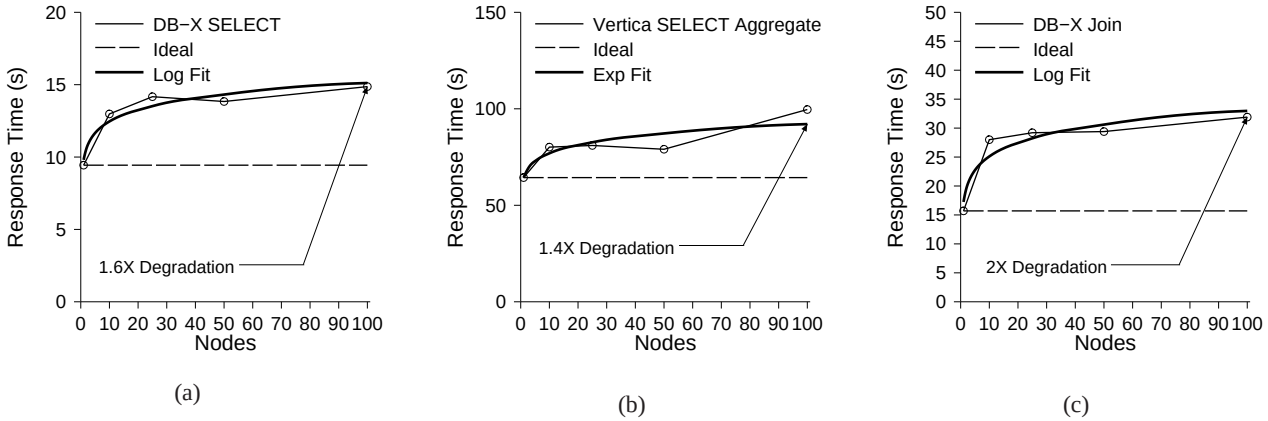


Figure 2: (a) DB-X running a 535MB/node SELECT query [19]. (b) Vertica running a 20GB/node SELECT aggregate query [19]. (c) DB-X running a join query (large table 20GB/node, small table 1GB/node) [19].

computation time can start to dominate the average runtime.

Consider Figure 2 where we have plotted the scaleup profiles of a simple selection query, another selection query but with an aggregate operation, and a third query with a join and an aggregation operation. These queries are taken from the recent paper by Pavlo et al. [19]. We present these real scaleup results, not for comparison purposes, but to illustrate the point that in practice, scaleup is often not ideally proportional for complex data processing workloads.

Figure 2(a) reports scaleup response time for a commercial parallel database, DB-X, running a simple SELECT scan where each node had 535MB partitions shows that there is significant response time degradation as the cluster size increases.

```
SELECT * FROM Data WHERE field LIKE '%XYZ%';
```

Table 'Data' has two rows 'key' and 'field' with sizes 10 and 90 characters respectively. Even for a workload as simple as a scan, the response time degradation at a 100X scaleup factor is 1.6 times worse than ideal.

Figure 2(b) shows an aggregate selection query with a scaleup from one node to 100 nodes. On a different commercial database, Vertica [33], the following query was run on a 233B wide table:

```
SELECT sourceIP, SUM(adRevenue)
FROM UserVisits GROUP BY SUBSTR(sourceIP,1,7);
```

Each node stored 20GB partitions of the table. This result showed that a different commercial parallel database also exhibits diminishing returns when the environment is scaled up; at 100X scaleup, there is a 1.4X performance degradation from the ideal performance.

Finally, Figure 2(c) shows a scaleup model for a complex join query on DB-X between the 'UserVisits' table in the above query, to a 108B table (1GB/node). The query also has an aggregation and date range predicate:

```
SELECT INTO Temp sourceIP, AVG(pageRank) as avgPageRank,
       SUM(adRevenue) as totalRevenue
FROM Rankings AS R, UserVisits AS UV
WHERE R.pageURL = UV.destURL
       AND UV.visitDate BETWEEN Date('2000-01-15')
       AND DATE('2000-01-22')
GROUP BY UV.sourceIP;
```

For DB-X, at 100X scaleup, the response time degradation is 2X worse than the ideal scaleup performance. Full details of the queries can be found in [19].

All three scaleup models show that there is a large startup cost between the one node 'cluster' and any multinode cluster. As a result, we have fitted logarithmic models to the DB-X results and an exponential model to the Vertica result to account for the initial drop in performance. The chosen models provided the best correlation coefficient of various regression models we applied.

The core point of this discussion is to show that scaleup performance is not ideally constant for complex data processing workloads; in which case *wimpy node scale-out to save energy and purchasing costs may not be more cost effective than traditional servers if equivalent performance is sought*. Equivalently, in real (as opposed to ideal) scaleup environments, price/performance degrades as the scaleup factor is increased (i.e., it gets more expensive to achieve the same level of performance). Next, we will show experimental results incorporating our price/performance results in Figure 1 and analyze the cost effectiveness of traditional clusters versus low-power/low-cost clusters.

3. EXPERIMENTAL EVALUATION

In this section, we discuss our experimental results which includes energy measurements of our nodes under different workloads using a commercial database system as well as scaleup experiments using published scaleup results. We start by describing the node characteristics and costs, followed by the measurement methods, and finally we present our results.

3.1 Server Costs and Specifications

In our tests, we use an Atom node (wimpy) and a typical high-end server-class Xeon node. Both ran the same commercial database system on Windows 7 Pro (Atom) and Windows Server 2008 (Xeon) which share the same kernel [26].

Atom Node: The Atom node had a dual core, hyper-threaded Intel D510 Atom processor with accompanying Intel motherboard ($80). The motherboard was filled with the maximum allowed 2x2GB GSkill DDR2 memory ($95). Our power supply was an 80plus certified Corsair VX450 ($65). We tried two different power supplies units (PSU); a cheap 120W PSU and a 450W Corsair. Even though the Corsair can provide almost 4X more power than the cheap PSU, we found that the power drawn by the system with the Corsair was almost half of that when using the cheap PSU. Given this, we chose the larger, but more efficient Corsair. We had two disk configurations: (1) the **SSD** configuration had an

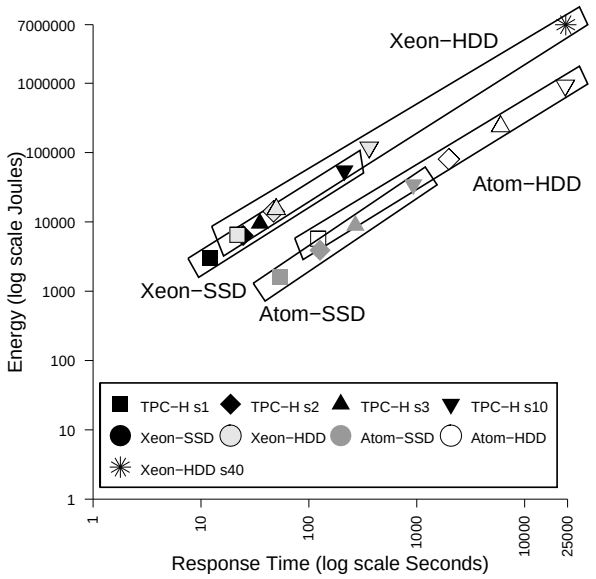


Figure 3: Raw measured response time and energy consumption of the TPC-H Power Test under various scale factors. TPC-H scale factors are represented by the shapes square, diamond, upward triangle, and downward triangle for scale factors 1, 2, 3, and 10 respectively. Shading of shapes correspond to different system configurations.

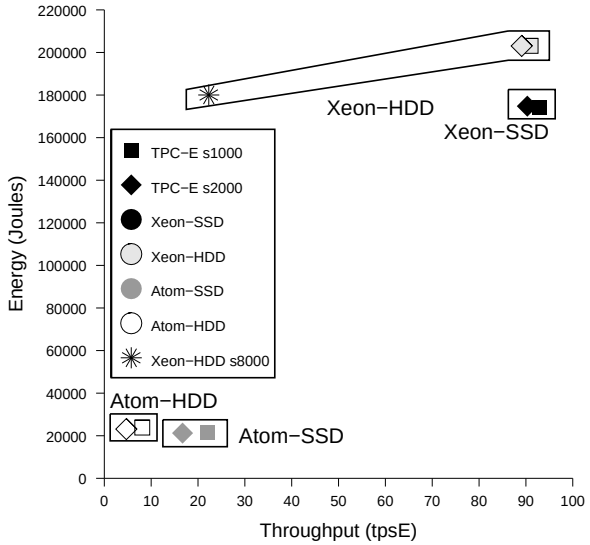


Figure 4: Raw measured throughput and energy consumption of a 10min window running TPC-E.

OCZ SATA2 64GB drive ($200) for the OS and DBMS applications and an Intel X-25E 32GB drive ($383) as the database data storage drive; (2) the mechanical **HDD** configuration used two WD Caviar Green SATA2 32MB Cache drives each with 500GB storage where both drives were used for database data storage ($120). The mechanical HDD configuration costs $360 while the SSD configuration costs $823.

Xeon Node: The Xeon node is an HP Proliant DL380GS with two quad core Xeon E5410 processors and 16GB of memory. The server has eight 146GB 10K RPM SAS drives with two in RAID1 for the OS and DBMS applications, another two in RAID1 for the DBMS log, and four drives for database data storage. This configuration cost approximately $3500. Each SAS drive can be purchased at $270 a drive. Our SSD configuration consists of removing the four data drives and replacing them with two Intel X25-E SSDs (priced as above). The server cost now is approximately $3186. This drop in cost between the high capacity SAS configuration and SSD configuration is similar to [2].

To keep this study manageable, we only explored a limited number of IO hardware configurations. Tuning this IO system for price, performance and energy is an interesting topic for future research, and beyond the scope of this paper.

3.2 Energy Measurement

AC current was measured at the wall outlet using a Fluke i200s AC current clamp (1.5% accuracy at 0.5A). The Fluke clamp was connected to an NI USB-6008 Multifunction DAQ and collected using NI's LabView, sampling at 1KHz. RMS current was calculated using a sliding window of 16 sample points (1 period) given an AC frequency of 50Hz. The RMS voltage was measured at 118V. Power was calculated as the product of the RMS current and the RMS voltage. Finally, energy consumption was calculated by summing the time discretized real power values over the length of the workload.

3.3 Single Node TPC-H

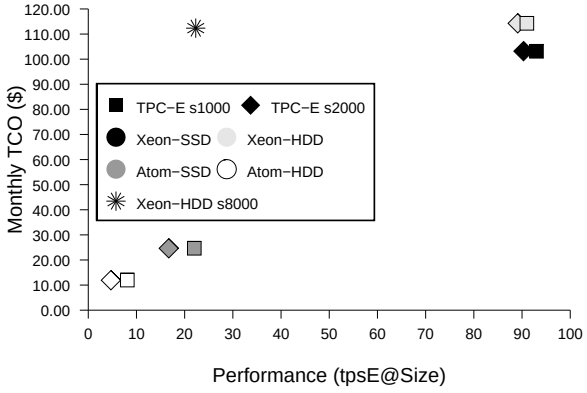
The raw energy and response time data that was used to create Figure 1 is shown in Figure 3. In this figure, we plotted the measured response time to complete the TPC-H Power Test against the energy consumed by the node during the run. There are a number of interesting features to note in Figure 3. First, for any given scale factor of TPC-H, the Atom node with the SSD configuration always consumes the least amount of energy. This is unsurprising given the low voltage of the Atom processor and SSDs. Second, the Xeon node with any storage configuration always finishes faster than the Atom-SSD node. This is also unsurprising given that the Xeon node has eight cores while the Atom only has two (four w/hyperthreading) and the Xeon node also has four times the memory of the Atom node.

Using these results, we calculated the amortized monthly total cost of ownership (TCO) using the node purchase prices (see Section 3.1) and a $0.07kWh datacenter energy cost [11]. Furthermore, we calculated the TPC-H Power@Size metric using the definition provided in [30]. Figure 1 shows our single node TPC-H results after transformation to a price/performance metric.

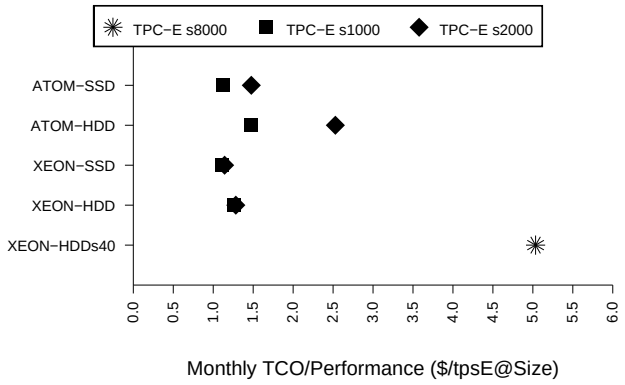
3.4 Single Node TPC-E

While TPC-H is a benchmark for decision support system (DSS), we also wanted to understand the performance and energy consumption profiles for OLTP workloads. Figure 4 shows a TPC-E throughput versus energy consumption plot similar to Figure 3. Here we have measured the throughput (in transactions per second/E – tpsE [31]) over a 10min period. This measurement period was preceded by a 10min warm up time where both systems stabilized. The energy measurements represent the energy consumption over 10min. Here we notice that the Xeon nodes always provide very high tpsE for both 1000 and 2000 scale factors because the system is essentially CPU bound at scale 1000 and 2000. The Atom nodes have significantly lower throughput but also about an order magnitude lower energy consumption.

Figure 5 shows the similar price/performance plot as Figure 1, but now for TPC-E. We notice that while Figure 4 shows that there was massive differences in energy consumption between the Xeon



(a)



(b)

Figure 5: (a) Amortized Monthly TCO (includes energy costs) as a function of Performance over various TPC-E scale factors. (b) Price/Performance metric of Amortized Monthly TCO ($) and Performance (tpsE@Size)

nodes and the Atom nodes, Figure 5 shows that the actual price/performance values for the nodes is only factors in difference and in the case of scale 1000, the Atom-SSD node has the same price/performance as the Xeon nodes.

TPC-E is a workload that was designed to reflect a more realistic benchmark compared to TPC-C, and is well-known to not have the easily-partitionable characteristics of TPC-C. Thus the impediments to perfect scaleup (see Section 2) will likely affect the scale-out story for TPC-E like workloads. In this paper, we do not consider TPC-E further, and focus only on the DSS TPC-H workload.

3.5 TPC-H Parallel Scaleup

Now, we examine the price and performance metrics for various scale-out clusters built from either traditional server nodes, or low-power Atom nodes. As we have seen in Figure 2(a-c), since scaleup characteristics are largely determined by the query workload, we apply these example scaleup models of Figure 2 to examine cluster price/performance. While these models are from various parallel systems and hardware, *the goal here is to show how the scale-out of our nodes would affect scaleup price/performance given various possible scaleup models*.

3.5.1 Modeling Cluster Performance

This section will describe the way we have applied the published scaleup models to our single node measurements in order to get cluster performance for a parallel data processing workload. First, the response time models we presented in Figure 2 are converted to give scaleup factors. Consider this example that illustrates how this is done: given that the DB-X Join response time model shows that using two nodes the workload response time will be 1.14X the response of one node, then the scaleup factor will be $2 \times 1/1.14 = 1.75$. This scaleup factor can then be used with our measured performance data to provide cluster performance.

Performance for a xy GB TPC-H dataset using x nodes at y GB partition each is calculated as $P(y)M(x)$ where $P(y)$ is the performance for the single node (Section 3.3) running TPC-H at scale y and $M(x)$ is the modeled scaleup factor given a cluster of x nodes (using the models in Figure 2). The ideal scaleup factor for x nodes is $M(x) = x$. For example, given a single node 10GB performance value of $P(10) = 9000QphH$, the ideal cluster performance for $M(2)$ nodes is $18000QphH$ while the modeled performance is $9000QphH \times 1.75 = 15840QphH$ for the Join model.

3.5.2 Wimpy Clusters vs One Xeon

The purpose of this section is to examine the effect of the startup costs, seen in Figure 2, on the cost of the Atom-SSD clusters as compared to a single Xeon-SSD node (which has no startup or any other parallel scale-out).

In Figure 6(a-c), we have applied each of the scaleup models of Figure 2 to the results shown in Figure 1. We have also included the data points for the ideal (constant) scaleup model. Since Figure 1 has shown us that outfitting nodes with SSDs typically decreases price/performance for both Atom and Xeon nodes, for our analysis here, we have used the SSD configurations of our nodes. To begin, consider Figure 6 which shows the results of scaleup when one Xeon node is compared to a cluster of Atom nodes.

For this discussion, we use the price and performance results for the Xeon-SSD node when running a TPC-H workload at scale 10. Based on the results shown in Figure 1(b), the Atom-SSD running TPC-H scale 2 has a similar price/performance rating as the Xeon-SSD at scale 10. Figure 1(b) shows that the Atom-SSD at TPC-H scale 2 (diamond) and the Xeon-SSD at TPC-H scale 10 (downward triangle) both have a price/performance of $0.011.

To match the Xeon-SSD workload size, Atom-SSD cluster will be made up of 5 Atom-SSD nodes each with 2GB partitions. This means that the monthly TCO of the Atom cluster will be 5 times the single node monthly TCO (at TPC-H scale 2). If we had ideal scaleup, then this Atom-SSD cluster would also provide 5 times the performance of a single Atom-SSD node thereby retaining a price/performance of $0.011. However, Figure 2 shows that this ideal scaleup does not happen and we calculate the performance of such a cluster using the methods described in Section 3.5.1.

In Figure 6(a), we show the price as a function of performance for two setups when applying the DB-X Scan scaleup model of Figure 2(a). Since we are only using a single Xeon-SSD, there is no scaleup effects and the modeled price and performance is identical to the ideal. The Atom cluster price/performance is 18% worse than the Xeon node for a 10GB workload.

Similarly, for Figure 6(b) and (c), we have applied the Vertica Aggregate scaleup (Figure 2(b)) and DB-X Join scaleup (Figure 2(c)) respectively. The Atom cluster price/performance is 13% and 31% worse than the Xeon node for the Vertica Aggregate and DB-X Join models respectively. It is clear that if the scaleup behavior, such as those in Figure 2(a-c), has poor degradation, this will be reflected in the cluster performance.

The results in this section are for a single Xeon node and an Atom cluster. The next section compares multi-node Xeon clusters to larger Atom clusters.

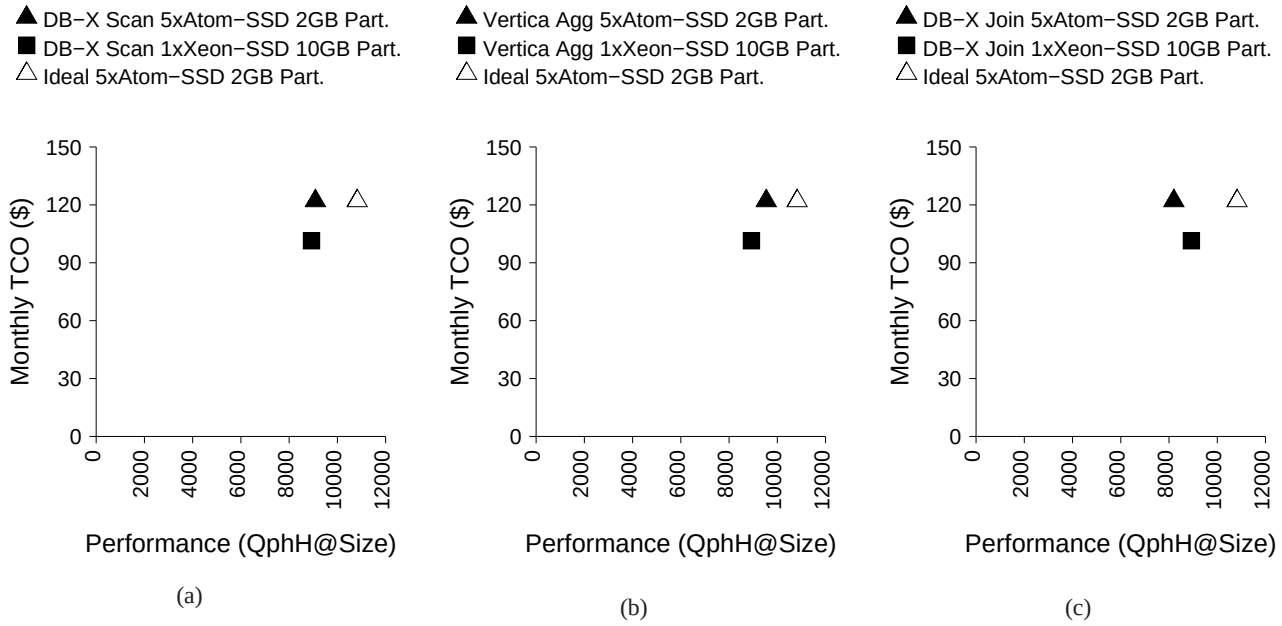


Figure 6: Parallel scaleup of a 10GB TPC-H workload on Atom and Xeon clusters using different published scaleup models (5 Atom-SSD nodes and 1 Xeon-SSD node respectively). Atom nodes have 2GB partitions and the Xeon node has the entire 10GB dataset. For all figures, the effects of an ideal constant scaleup is shown. (a): DB-X Scan scaleup model (Figure 2(a)). (b): Vertica Aggregation scaleup model (Figure 2(b)). (c): DB-X Join scaleup model (Figure 2(c)).

3.5.3 Wimpy Clusters vs Traditional Clusters

This section compares various levels of Atom-SSD scaleup for a 60GB TPC-H workload where the Xeon-SSD cluster is made of 6 nodes, each running 10GB partitions. We applied the DB-X Join scaleup model from Figure 2(c) as it represented the most complex workload of the three we discussed. Modeling was done as described in Section 3.5.1.

Figure 7(a-c) shows the price as a function of performance similar to Figure 6. In these figures, the amount of data (partition size) per Atom-SSD node decreases from 3GB (Figure 7(a)) to 1GB (Figure 7(c)).

In the progression of decreasing partition size for the Atom cluster, as the partition size decreases, the size of the Atom cluster increases. First, we notice that the 20 node Atom cluster is cheaper than the Xeon cluster in Figure 7(a). However, its performance is about half of the Xeon-SSD cluster. If we consider the price/ performance, the 20 node Atom cluster is 55% higher than the 6 node Xeon cluster.

Next, in Figure 7(b), where the Atom cluster is 30 nodes large, both clusters have roughly equivalent performance. However, we notice that the cost of the Atom cluster is 20% higher than the Xeon cluster. In this case, the 30 node Atom cluster is 23% higher in price/performance than the 6 node Xeon.

Finally, in Figure 7(c), we notice that as the Atom cluster increases with decreasing partition size, the performance increases. However, this is an effect of the increasing performance of the Atom-SSD as it works on a smaller, in memory dataset (at 1GB partition). It is quite clear in Figure 7(c), where the 60 node Atom-SSD cluster has higher performance than the 6 node Xeon-SSD cluster, that the cost to deploy such a wimpy node cluster is significantly higher than the Xeon-SSD cluster. While the Atom cluster's performance is 2X better than the Xeon, it is 2.4X more costly. This translates to a 19% increase in price/performance over the Xeon cluster.

Since this analysis has held the Xeon-SSD cluster size constant while we varied the Atom-SSD cluster size, it is necessary to compare the cluster price/performance when both clusters vary in size.

Consider Figure 8, where we plot the (a) performance, (b) price, and (c) price/performance as a function of the dataset size for both clusters when the Join scaleup model is applied (Figure 2(c)). Here we partition the data so each Atom-SSD has 2GB partitions and each Xeon-SSD has 10GB partitions.

Figure 8(a) shows that as the clusters scaleup, the Atom-SSD cluster starts to exhibit higher performance than the Xeon-SSD cluster. This is because the Xeon cluster is growing and starts to become affected by the DB-X Join scaleup degradation. This performance difference in Figure 8(a) can be explained by the models in Figure 2 that show that parallel scaleup behavior flattens out as the cluster sizes increases. While the Atom cluster has slightly better performance with larger scaleup, Figure 8(b) shows that this is accompanied by higher cluster cost. Finally, Figure 8(c) shows the price/performance of both clusters under scaleup. It shows that the increase in performance and cost of the Atom-SSD cluster over the Xeon-SSD cluster is largely proportional.

3.5.4 Discussion

We have shown the effects of various scaleup models on two types of clusters running TPC-H workloads: an Atom-SSD cluster and a Xeon-SSD cluster. While these models are not directly drawn from a TPC-H workload, the purpose of using these models is to show how scale-out of different cluster architectures can be affected by different scaleup behavior. As such we have applied a variety of published scaleup models to the TPC-H measurements we collected.

With computationally simple workloads, such as those of [3], "wimpy" node clusters are claimed to be more effective than clusters made from traditional nodes. However, this analysis has shown that for data-intensive workloads (Figure 6, 7, and 8), large wimpy node clusters suffer from poor scaleup effects and are therefore po-

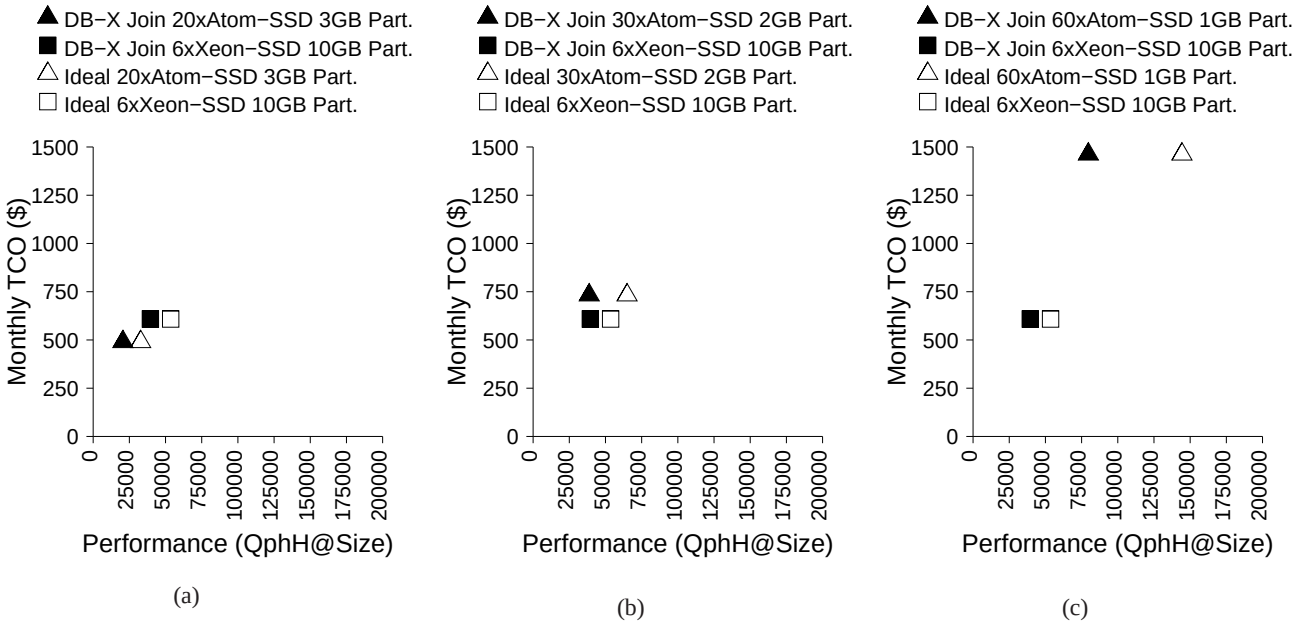


Figure 7: Parallel scaleup of a 60GB TPC-H workload on Atom (with different partition sizes) and Xeon (10GB partitions) clusters using the join scaleup model from Figure 2(c). (a) 3GB Atom-SSD partitions, (b) 2GB Atom-SSD partitions, (c) 1GB Atom-SSD partitions.

tentially slower and a costlier solution than smaller Xeon clusters. The reason for this is because larger wimpy clusters are more affected by a diminishing return scaleup effect than a smaller traditional cluster.

Furthermore, small relative gains in performance or price/ performance by a larger low-power cluster over a smaller traditional cluster may not be worth the increase in mean-time-to-failure for the entire cluster [27, 28]. Providing fault tolerance over a large number of wimpy nodes requires replication and over-provisioning, thereby increasing the price/performance of such clusters.

Finally, increasing the cluster size by migrating over to Atom nodes requires substantially more network infrastructure. For example, if we assume that the wimpy node clusters will require 4X more nodes than the traditional cluster, then given a 16 node Xeon cluster that requires one 48 port switch, a wimpy 64 node cluster will require two 48 port switches. High performance network hardware cannot be sacrificed in wimpy node clusters and anecdotally we found this to be true in real deployments of Atom clusters. Optimistically assuming enterprise class 48 port 10 Gigabit switches cost $10,000 each, the additional amortized switch cost per wimpy node is $313 which doubles the cost of the spindle-based Atom node to $700 and increases the SSD-based Atom cost to $1150!

Our results suggest that there may be an interesting middle ground between wimpy Atom node clusters and traditional Xeon node clusters that lies with hybrid cluster deployments [7]. Such clusters made up of both wimpy and traditional nodes may be the most effective deployment. The architectural make-up of such a hybrid cluster and its scaleup behavior is an interesting avenue of future research.

4. RELATED WORK

Recent studies on the energy consumption of large clusters [4, 10, 1] have shone a light on existing [22, 21, 18, 5] and future problems faced by datacenter operators. These studies discussed the increasing size of datacenter server clusters as well as their dramatic energy inefficiency due to the absence of energy proportionality (proportional energy consumption with hardware utilization).

The database community has begun to seriously consider the energy costs of database management systems [12, 13, 17, 24, 20]. In [24, 25], a well-defined sort metric for the energy efficiency of a hardware platform was given as well as hints as to the nature of an energy-efficient system configuration. Furthermore, it was shown that over the past decade, published TPC-C results used systems that have increased their power draw six-fold [20], and studies began revealing the true energy consumption costs of running database workloads [13, 17]. While these studies have examined the energy consumption profiles of single nodes running a database, a 'local' approach to energy efficiency, another direction is to treat a cluster as a holistic entity for energy optimization.

This 'global' approach to energy management offers many different directions for research. Both the systems and database communities have produced a number of studies on reducing the energy consumption of data processing clusters [23, 3, 29, 11]. As mentioned in this paper, some of these have focused on rethinking cluster node architecture and using new low-power, low-cost hardware [23, 3]. Approaches such as service consolidation through virtual machine migration have been explored [29, 8]. Finally, custom energy-efficient hardware has been presented which targets the needs of datacenter operators [11].

Powering down cluster nodes has been one method in which these studies have looked at achieving energy proportionality [14, 15, 16]. In [14], a study on powering down a replicated parallel database cluster with an eye on load balancing was presented. Powering down MapReduce clusters was discussed in [15]. A new server architecture that eliminates server idle power draw by rapid transitioning of component power states is presented in [16].

5. CONCLUSIONS

Our study presents evidence that for complex data processing workloads, a scale-out solution of a low-power low-end CPU-based

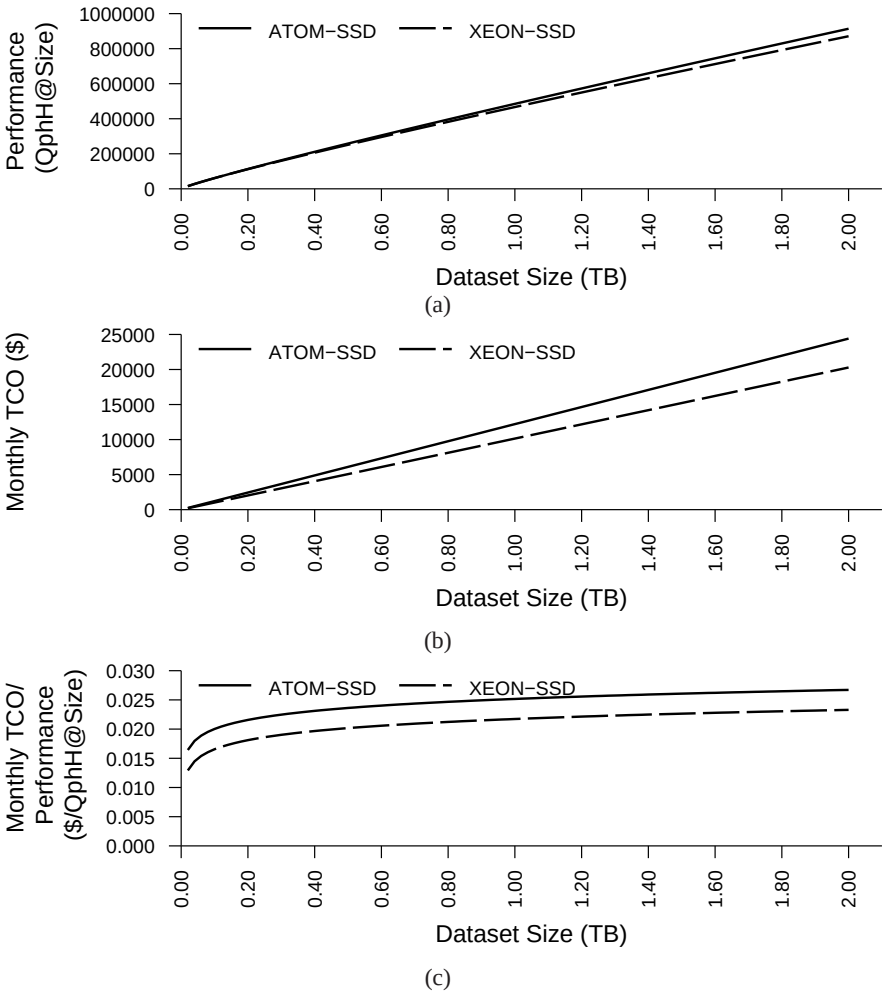


Figure 8: Analysis of TPC-H workload: (a) Performance, (b) Price, and (c) Price/Performance for Atom-SSD and Xeon-SSD based clusters using the Join scaleup model (Figure 2(c)). Atom-SSD nodes have 2GB partitions and Xeon-SSD nodes have 10GB partitions.

cluster may not be as cost-effective (or produce equivalent performance) as a smaller scale-out cluster of traditional high-end server nodes. We have shown that depending on the scaleup characteristics of the query workload and the software system, poor scaleup behavior can occur when increasing the cluster size. Poor scaleup degrades the price/performance of a larger cluster. Thus, the parallel scaleup characteristics of the environment largely determines the feasibility of so-called "wimpy" node configuration for building clusters for such complex data processing workloads.

While our results suggest that wimpy node clusters are not suited for complex database workloads, it does open up the area of hybrid (heterogeneous) cluster deployment. Hybrid cluster deployment strategies, job scheduling, and scaleup analysis are interesting avenues of future research.

6. ACKNOWLEDGEMENTS

We would like to thank the reviewers of this paper for their constructive comments. This research was supported in part by a grant from the Microsoft Jim Gray Systems Lab, Madison, WI.

7. REFERENCES

[1] Report To Congress on Server and Data Center Energy Efficiency. In *U.S. EPA Technical Report*, 2007.

[2] A Comparison of SSD, ioDrives, and SAS rotational drives using TPC-H Benchmark. Technical Report White Paper, HP Development Company, 2010.

[3] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan. FAWN: A Fast Array of Wimpy Nodes. In *SOSP*, 2009.

[4] L. A. Barroso and U. Hölzle. The Case for Energy-Proportional Computing. *IEEE Computer*, 40(12), 2007.

[5] C. Belady. In the Data Center, Power and Cooling Costs More than the IT Equipment it Supports. *Electronics Cooling*, 23(1), 2007.

[6] K. G. Brill. Data Center Energy Efficiency and Productivity. In *The Uptime Institute - White Paper*, 2007.

[7] B.-G. Chun, G. Iannaccone, G. Iannaconne, R. Katz, G. Lee, and L. Niccolini. An Energy Case for Hybrid Datacenters. In *HotPower*, 2009.

[8] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live Migration of Virtual Machines. In *NSDI*, 2005.

[9] D. DeWitt and J. Gray. Parallel Database Systems: The Future of High Performance Database Processing. In *CACM*, 1992.

[10] X. Fan, W.-D. Weber, and L. A. Barroso. Power Provisioning for a Warehouse-sized Computer. In *ISCA*, 2007.

[11] J. Hamilton. Cooperative Expendable Micro-slice Servers (CEMS): Low Cost, Low Power Servers for Internet-Scale Services. In *CIDR*, 2009.

[12] S. Harizopoulos, M. A. Shah, J. Meza, and P. Ranganathan. Energy Efficiency: The New Holy Grail of Database Management Systems Research. In *CIDR*, 2009.

[13] W. Lang and J. M. Patel. Towards Eco-friendly Database Management Systems. In *CIDR*, 2009.

[14] W. Lang, J. M. Patel, and J. F. Naughton. On Energy Management, Load Balancing and Replication. In *SIGMOD Record*, 2010.

[15] J. Leverich and C. Kozyrakis. On the Energy (In)efficiency of Hadoop Clusters. In *HotPower*, 2009.

[16] D. Meisner, B. T. Gold, and T. F. Wenisch. PowerNap: Eliminating Server Idle Power. In *ASPLOS*, 2009.

[17] J. Meza, M. A. Shah, P. Ranganathan, M. Fitzner, and J. Veazey. Tracking the Power in an Enterprise Decision Support System. In *ISLPED*, 2009.

[18] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making Scheduling 'cool': Temperature-aware Workload Placement in Datacenters. In *USENIX*, 2005.

[19] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker. A Comparison of Approaches to Large-Scale Data Analysis. In *SIGMOD*, 2009.

[20] M. Poess and R. O. Nambiar. Energy Cost, The Key Challenge of Today's Data Centers: A Power Consumption Analysis of TPC-C Results. In *VLDB*, 2008.

[21] K. Rajamani and C. Lefurgy. On Evaluating Request-Distribution Schemes for Saving Energy in Server Clusters. In *Proc. of the IEEE Intl. Symp. on Performance Analysis of Systems and Software*, 2003.

[22] P. Ranganathan, P. Leech, D. Irwin, and J. Chase. Ensemble-level Power Management for Dense Blade Servers. In *ISCA*, 2006.

[23] V. J. Reddi, B. Lee, T. Chilimbi, and K. Vaid. Web Search Using Small Cores: Quantifying the Price of Efficiency. Technical Report MSR-TR-2009-105, Microsoft Research, 2009.

[24] S. Rivoire, M. A. Shah, P. Ranganathan, and C. Kozyrakis. JouleSort: a balanced energy-efficiency benchmark. In *SIGMOD*, 2007.

[25] S. Rivoire, M. A. Shah, P. Ranganathan, C. Kozyrakis, and J. Meza. Models and Metrics to Enable Energy-Efficiency Optimizations. *Computer*, 2007.

[26] M. Russinovich. Windows 7 and Windows Server 2008 R2 Kernel Changes. In *Microsoft Tech Ed*, 2009.

[27] B. Schroeder and G. A. Gibson. Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You. In *USENIX Conference on File and Storage Technologies*, 2007.

[28] B. Schroeder, E. Pinheiro, and W. D. Weber. DRAM Errors in the Wild: a Large-Scale Field Study. In *SIGMETRICS*, 2009.

[29] N. Tolia, Z. Wang, M. Marwah, C. Bash, P. Ranganathan, and X. Zhu. Delivering Energy Proportionality with Non Energy-Proportional Systems - Optimizing the Ensemble. In *HotPower*, 2008.

[30] Transaction Processing Council. `http://www.tpc.org/tpch`.

[31] Transaction Processing Council. `http://www.tpc.org/tpce`.

[32] V. Vasudevan, D. G. Andersen, M. Kaminsky, L. Tan, J. Franklin, and I. Moraru. Energy-efficient cluster computing with FAWN: Workloads and implications. Apr. 2010. (invited paper).

[33] Vertica Systems, Inc. `http://www.vertica.com`.