Characterizing the Scalability of Decision-Support Workloads on Clusters and SMP Systems

Yanyong Zhang[1], Anand Sivasubramaniam[1], Jianyong Zhang[1]
Shailabh Nagar[2], Hubertus Franke[2]

[1] Dept. of Computer Science & Engg., The Pennsylvania State University
University Park, PA 16802, USA
{yyzhang,anand,jzhang}@cse.psu.edu
[2] IBM Thomas J. Watson Research Center,
Yorktown Heights NY 10598, USA
{nagar, frankeh}@us.ibm.com

Abstract. Using a public domain version of a commercial clustered database server and TPC-H like[3] decision support queries, this paper studies the performance and scalability issues of a Pentium/Linux cluster and an 8-way Linux SMP. The execution profile demonstrates the dominance of the I/O subsystem in the execution, and the importance of the communication subsystem for cluster scalability. In addition to quantifying their importance, this paper provides further details on how these subsystems are exercised by the database engine.

1 Introduction

Commercial workloads have long been used to benchmark the performance of server systems. These workloads have influenced the design of all aspects of computer systems from hardware to operating systems, middleware and applications. The TPC series of benchmarks is an important set of representative commercial workloads that can be used to test the performance of computer systems.

The goal of this paper is to study the scalability characteristics of the TPC-H decision support benchmark. Specifically, it examines the impact of scaling important system resources such as CPU, memory, disks and network on the performance of a midsized TPC-H benchmark implemented on a standard database engine. The study uses the DB2 database engine from International Business Machine Corp. (IBM) on two commonly used server platforms : a cluster of 2-way symmetric multiprocessors (SMPs) and an 8-way SMP, both running the Linux operating system (OS).

In our study, the DB2 database engine, the Linux operating system and the hardware characteristics, such as the speeds of the processor, memory and

[3] These results have not been audited by the Transaction Processing Performance Council and should be denoted as "TPC-H like" workload.

disk, are all considered to be part of the environment. As such, we have done only a reasonable amount of optimization of DB2 and the Linux kernel. The focus of the study is on understanding the impact of varying the amount of hardware resources on the performance of the TPC-H benchmark. Our work neither attempts to maximize the performance of the benchmark nor does it try to evaluate the database engine or the operating system. The same reporting standards that serve to make TPC-H an important workload also prevent us from reporting absolute performance numbers. Since the study focusses on scalability rather than performance, this does not limit the value of the reported results.

Studying the impact of varying amounts of hardware resources on TPC-H performance offers two benefits. It aids system administrators in capacity planning and tuning. Further, middleware and OS designers can gain insights into scalability bottlenecks which helps them make design tradeoffs.

The methodology followed by our study is as follows. We first characterize the various queries in the TPC-H benchmark on a cluster with respect to their usage of CPU, memory and I/O bandwidth. We then run the benchmark on a cluster of 2-way SMPs and study the performance impact of adding nodes to the cluster. Each added node increases the CPU, memory and disk resources available for the workload and incurs a potential penalty of increased network costs. We vary the number of CPUs and memory in the cluster to try to identify which of the resources affects performance the most. A similar exercise is done on an 8-way SMP platform. Besides eliminating the effects of networking, the SMP platform allows a greater flexibility in varying the number of CPUs and memory. Finally, we analyze these three sets of results to glean characteristics of the workload.

The results of the experiments broadly show that I/O is by far the most important scalability bottleneck on both the cluster and the SMP platform. Individual queries demonstrate different attributes which can be correlated to their characteristics. The importance of I/O bandwidth suggests that a more aggressive overlap of computation and I/O would be desirable while designing the next generation databases and operating systems. It also suggests that it is better to explicitly increase the I/O parallelism in a cluster rather than rely on an implicit increase through node addition. Similarly in an SMP environment, more than 4 CPUs and 1.5 GB memory does not increase performance for even a 10GB dataset size. It is more useful to add disks and increase the I/O bandwidth.

The second important conclusion of this study is that the networking overheads of a cluster are not a significant scalability inhibitor. The benefits of the added disks, CPU and memory outweigh the additional networking cost when a cluster is scaled by adding a node. Such scaling is seen to be a viable option even with the current levels of clustering support in hardware, database and OS.

2 Experimental Setup

The TPC-H benchmark is best described by TPC's own website as follows: "TPC Benchmark H (TPC-H) is a decision support benchmark. It consists of a

suite of oriented ad-hoc queries and data modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance. This benchmark illustrates decision systems that examine large volumes of data, execute queries with a high degree of complexity, and give answers to critical business questions".

In the interest of space, we are not including all the details about the distribution of TPC-H tables across the cluster nodes or the implementation of the queries. This workload contains a sequence of 22 queries (Q1 to Q22), that are fired one after another to the database engine. Queries 21 and 22 take too long to run on clusters and hence are omitted from the results shown. There are several measures that are used to determine their performance as specified in [2]. We choose query completion time as our performance metric.

TPC-H workloads use standard dataset sizes ranging from 1 to 3000 GB. We have chosen to run TPC-H with a 10GB dataset size, keeping in mind our resource and time constraints. This is a representative dataset size for a small business. TPC-H workload was run on two platforms : a cluster of 2-way SMPs and an 8-way SMP. Henceforth we shall refer to the former as the cluster and the latter as the SMP.

The cluster consists of eight nodes with each node having two Pentium II CPUs, 256 MB RAM and one disk of 9 GB capacity. The nodes are connected by both switched Myrinet [3] and Ethernet. Unless stated otherwise, TCP over Myrinet is used for network communication. Ethernet is used in one set of results to show that the network bandwidth is not a scalability bottleneck. The cluster nodes run Redhat Linux 7.2 with kernel version 2.4.8. This kernel has been instrumented in detail to gather different statistics, and also modified to provide insight on the database engine execution.

During the runs, a client machine (not part of the cluster), sends the TPC-H queries to a database coordinator node on the cluster, which then distributes the work and gives back the results to the client. A freely available version of DB2 Extended Enterprise Edition (EEE) version 7.2 [1] was used. The EEE version of DB2 is specifically written to take advantage of cluster hardware. The database was run in partitioned mode with one partition per cluster node. Hardware resources were scaled mainly through the addition of nodes. In some experiments, individual node configurations were modified to use only one CPU or lesser RAM through the `maxcpus` and `mem` Linux boot parameters.

The SMP experiments were conducted on an 8-way IBM Netfinity 8500R server with PIII processors, 2MB L2 cache and 2.5GB of main memory. The operating system was Red Hat Linux 7.2 running the 2.4.17 kernel. The default OS installation was modified to increase kernel resource limits such as the number of open files and semaphores. A scalable timer patch was also applied to the base kernel to take care of known timer issues. No TPC-H specific tuning was done to the OS. The freely available version of DB2 Enterprise Edition (EE) version 7.2 [1] was used on the SMP. The number of CPU's and physical memory configuration was varied using `maxcpus` and `mem` boot parameters.

query	user CPU (%)	system CPU (%)	page faults per jiffy	blocks read per jiffy	blocks written per jiffy	packets sent per jiffy	packets received per jiffy	CPU utilization during IO (%)
Q1	53.80	17.71	1.50	51.01	23.1151	0.0012	0.0015	26.87
Q2	70.78	15.67	0.39	21.97	1.7718	1.9077	1.9248	53.73
Q3	69.93	17.19	0.99	55.47	4.9591	0.9778	0.9938	55.40
Q4	53.38	17.10	0.00	22.97	1.0478	0.3517	0.3652	68.41
Q5	60.92	17.09	0.05	16.73	1.1369	2.0779	2.0849	42.81
Q6	41.41	25.44	1.73	90.72	0.0020	0.0012	0.0012	33.49
Q7	68.54	14.76	0.00	16.89	1.9467	2.3880	2.3521	31.04
Q8	28.52	23.22	0.01	27.63	4.8078	0.0261	0.0228	12.91
Q9	63.42	15.51	0.00	6.69	1.9276	0.0133	0.0136	23.21
Q10	39.99	20.40	0.17	41.99	1.8880	0.8774	0.8859	18.71
Q11	79.63	13.16	0.49	18.87	0.0020	2.3794	2.4011	66.46
Q12	16.07	21.96	0.25	40.79	0.0197	0.0102	0.0101	4.8
Q13	49.79	22.31	1.62	45.86	0.0034	2.0966	2.0935	32.71
Q14	36.91	23.27	1.01	84.78	0.0025	0.1156	0.1175	29.75
Q15	55.37	18.69	0.60	75.26	0.0033	0.6260	0.7703	51.68
Q16	51.84	15.30	2.32	15.36	0.8454	2.4836	2.4993	46.24
Q17	33.02	20.49	0.00	32.76	5.2532	0.0036	0.0037	13.94
Q18	65.77	20.15	0.04	17.16	0.6261	0.3890	0.3803	35.11
Q19	38.11	22.29	0.29	78.76	0.0032	0.3343	0.3329	23.38
Q20	29.16	19.35	0.00	15.74	0.1601	0.3456	0.2973	47.36

Table 1. OS profile for 8-node cluster (statistics are collected from node 1)

3 Operating System Profile

Before we study how the queries scale when we increase different hardware resources, it is important for us to first characterize the executions of these queries. These statistics will give us indications what resources are the main limiting factors to the workloads. There are four main hardware components which are being exercised by TPC-H, namely, CPU, memory, disks and network (the last is only applicable in the clustered version). In Linux, one can obtain resource usage statistics through the proc file system which are updated every jiffy (10 milliseconds). We sample these statistics roughly every 500 milliseconds to minimize the perturbation due to the sampling. By sampling the numbers given by files (`stat`, `net/dev`, `process/stat`), we obtain a number of interesting statistics for each query. These are shown in Table 1 for the cluster and in Table 2 for the SMP environment.

We observe that the bulk of the query execution time is spent in waiting for disk I/O. High number of disk I/Os result in not only poor CPU utilization, but also high system call overheads. Amongst the I/O operations, reads are much more common than writes. Individual queries have specific characteristics which are referred to later.

4 Scalability Issues in Clustered Database Engine

In the first set of experiments, we study how the average job response time scales with the number of nodes in the cluster. When we increase the number of nodes, the total amount of memory and the number of disks will also increase

id	user CPU (%)	system CPU (%)	page faults per unit time	blocks read per unit time	blocks written per unit time
1	44.70	39.35	0.00	0.11	0.1109
3	27.86	31.26	0.02	0.05	0.1069
4	16.11	29.29	0.00	0.08	0.1886
5	24.39	24.23	0.02	0.15	0.1695
6	14.90	19.30	0.01	0.71	0.5748
7	27.92	13.17	0.01	0.06	0.0822
8	15.97	9.98	0.01	0.02	0.0151
9	17.56	9.40	0.06	0.04	0.1620
10	31.23	35.54	0.63	0.02	0.2277
11	5.01	7.12	0.35	0.01	0.0118
12	4.93	5.49	0.04	0.02	0.0081
13	45.11	9.83	1.47	0.02	0.0729
14	9.39	16.80	1.47	0.21	0.5112
15	19.93	25.27	1.26	0.02	0.0698
16	25.15	13.23	7.08	0.12	0.0107
17	16.00	9.26	4.20	0.15	0.2397
18	41.44	18.89	0.03	0.05	0.1501
19	39.78	41.32	0.05	0.25	0.0955
20	6.07	7.98	0.28	0.19	0.0078

Table 2. OS profile for 8-way SMP

proportionally. Fig.1 shows the results as the nodes in the cluster are increased from 1 to 8. It may be noted that for the one node configuration, a software RAID using two 9 GB disks was utilized to accomodate the 10GB dataset. Six of the bars shown are truncated in the interest of clarity; their values are: 8.85, 2.4, 1.38, 1.47, 57.64 and 4.07 respectively.

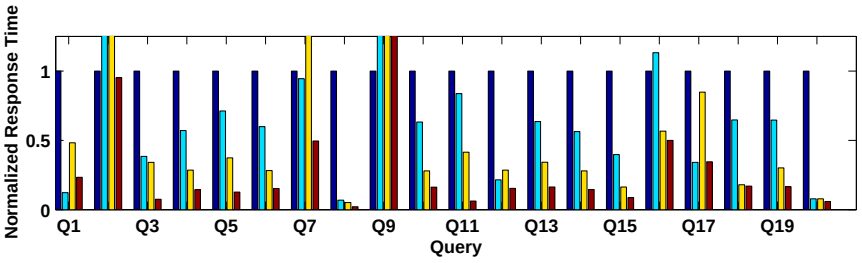

Fig. 1. For each query, we show (left to right) the query response times with a configuration using (respectively): (i) one 2-way SMP node; (ii) two 2-way SMP nodes, myrinet; (iii) four 2-way SMP nodes, myrinet, and (iv) eight 2-way SMP nodes, myrinet. Execution times are normalized with respect to configuration (i).

We have the following observations from the results :

– Increasing the number of nodes can increase the disk parallelism, the total amount of memory which can be used as the bufferpool, and CPU processing power. On the other hand, it will also incur/increase the network overhead. In

general, we find that the benefits of having more nodes offset the drawbacks. We can decrease the response time when we have more nodes.

- Queries Q4, Q6, Q10, Q13, Q14, Q15 and Q19 have a more or less linear decrease in response times. This is because these queries either have high number of disk accesses like in Q6, Q13, Q14, and Q19 (blocks read per jiffy column in table 1), large scope to hide IO cost with useful computation as in Q4 (the last column in. table 1), a mix of these two factors as in Q10 and Q15. In these cases, more nodes can bring in disk parallelism, and they can hide disk overhead with computation.
- Queries Q2, Q3, Q5, Q11 have a super-linear decrease in response times when we increase the number of nodes. These queries also have the characteristics of the above queries. Besides, their CPU utilization is very high. As a result, they can benefit not only from more disk parallelism, but also from more CPU parallelism by more nodes.
- Queries Q7, Q8, Q9, Q16, Q18, and Q20 do not seem to scale from 2/4 nodes to 8 nodes. It can be explained as follows. Both of these queries have low disk accesses, so they cannot benefit much from the disk parallelism. Also, they incur quite high network overhead (packets sent per jiffy column in table 1 when we have more nodes in the system.
- Queries Q1, Q12 and Q17 do not benefit much from adding more nodes in general. It can be explained by the fact that all three have low overall CPU utilization which make them not able to benefit much from the parallelism.

From this study, we can see that performance is improved when we increase number of nodes in the system. When the number of nodes increases, CPU, memory and disk all increase linearly, which all contribute to the performance improvement. We now examine which component is playing the most important role. First, we keep the total amount of memory constant when we increase nodes. Then we keep the number of CPUs in the system constant while increasing the nodes.

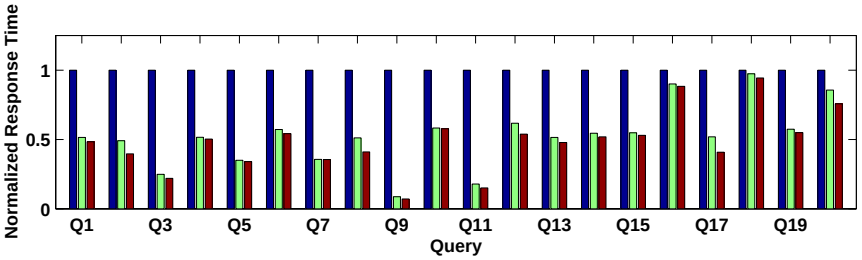

Fig. 2. For each query, we show (left to right) the query response times with a configuration using (respectively): (i) four 2-way SMP nodes, myrinet, with 256M RAM on each node; (ii) eight 2-way SMP nodes, myrinet, with 128M RAM on each node; and (iii) eight 2-way SMP nodes, myrinet, with 256M RAM on each node. Execution times are normalized with respect to configuration (i).

To help us understand the impact of available physical memory on workload performance, we use three system configurations : 4 nodes with 256MB RAM, 8 nodes with 128MB RAM and 8 nodes with 256M RAM, henceforth referred by (4,256), (8,128) and (8,256). In Fig. 2, we see great performance improvement for most of the queries as we move from (4,256) to (8,256). On the other hand, going from (8,128) to (8,256) shows only marginal performance gains. This shows that memory is not a major contributor to the performance, though it can have some impact. In TPC-H, most of the queries involve sequential scans through the data leading to working sets which are much larger than the available memory. Furthermore, the data reuse rate is very low. In general, in the presence of both these characteristics i.e. large working set size and low data locality, adding memory only provides marginal benefits. This is further confirmed by the relatively large performance improvements seen for queries 17 and 20. An examination of their data access patterns shows that they both have relatively smaller working sets.

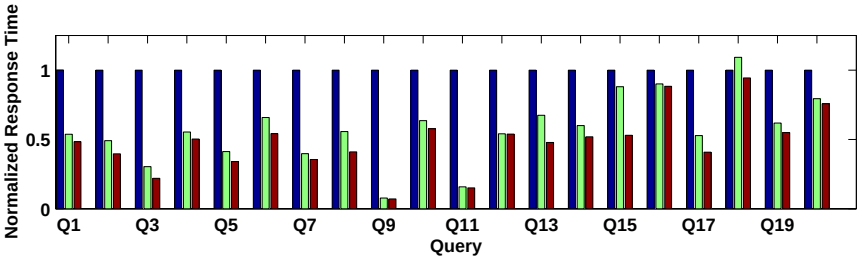

Fig. 3. For each query, we show (left to right) the query response times with a configuration using (respectively): (i) four 2-way SMP nodes ; (ii) eight uniprocessor nodes and (iii) eight 2-way SMP nodes; Execution times are normalized with respect to configuration (i).

Next we want to see if CPU is the limiting factor for queries. We compare the performance of 4 dual cpu nodes, 8 single cpu nodes and 8 dual cpu nodes, henceforth called (4,2), (8,1) and (8,2). Fig. 3 shows that going from (8,1) to (8,2) does not help most of the queries except Q15 (which has a high CPU utilization as well as a good computation overlap with I/O as is seen in Table 1. But we see a substantial gain while going from (4,2) to (8,1). From this we can conclude that the major benefits of increased nodes do not come from the addition of CPUs.

Having eliminated the CPU and memory as overall performance boosters, we next look at the major performance inhibitor in a cluster environment, namely the network. In Fig. 4 we compare the performance under two different network subsystem configurations. In the first setup which is our default network configuration, we run TCP/IP over Myrinet. The Myrinet link bandwidth is 1.06 Gbps which is substantially higher than that of 100 Mbps Ethernet which is the

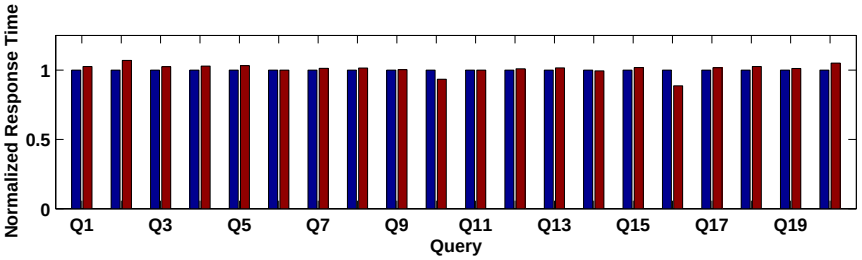

Fig. 4. For each query, we show (left to right) the query response times with a configuration using (respectively): (i) eight 2-way SMP nodes, myrinet, and (ii) eight 2-way SMP nodes, 100Mbps Ethernet. Execution times are normalized with respect to configuration (i).

second setup. While one would expect the faster network hardware to show some benefit, we see less than 5% improvement. Q10 and Q15 even show a degradation with Myrinet. Overall we can conclude that the network is not a bottleneck in this configuration for a 10GB dataset.

Of the four main components of a clustered configuration, namely CPU, memory, network and disk, we have examined the performance impact of the first three. Through individual scalability analysis of each of these components, we have shown that none of them is a significant contributor to scalability. But since we do see an overall improvement in performance as the number of nodes is increased, we conclude that that the increased I/O bandwidth is the major contributor the performance gains.

5 Scalability on SMP

In this section, we analyze the scalability of TPC-H on the 8-way SMP platform. We do not intend to compare the scalability of a cluster with that of an SMP as the underlying hardware and middleware is significantly different. For an SMP server, there are mainly three components which are exercised by the workloads: CPU, memory and disks. We investigate the impact of each of these components individually.

Fig. 5 shows the performance impact when system memory is increased from 1024 MB to 2560 MB. For 8 of the 20 queries shown, we see significant performance gains as memory increases from 1024 to 1536 MB. After that and for all queries, increasing memory barely improves response time. This indicates that memory is not a limiting resource for these queries. Three bars are truncated in the interests of clarity; their values are: 1.54, 1.70 and 1.71 respectively.

Next, we vary the number of available CPUs to investigate its impact on the response time for each query. Though adding CPUs might increase overall parallelism, on an SMP it might also lead to increased contention in the middleware and OS. Fig. 6 shows the response times for each query as the number of CPUs is varied from 1 to 8.

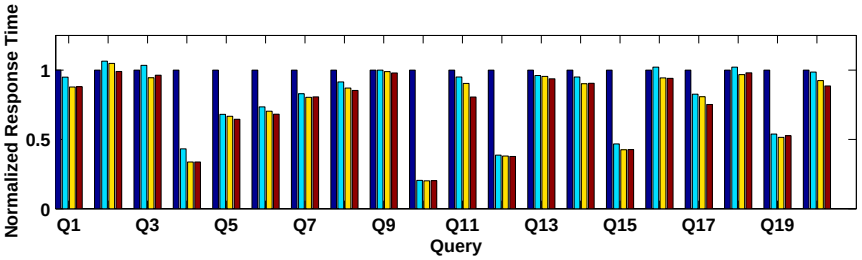

Fig. 5. For each query, we show query response times with a configuration using respectively (left to right): (i) 8-way SMP, 1024M RAM; (ii) 8-way SMP, 1536M RAM; (iii) 8-way SMP, 2048M RAM; and (iv) 8-way SMP, 2560M RAM; Execution times are normalized with respect to configuration (i).

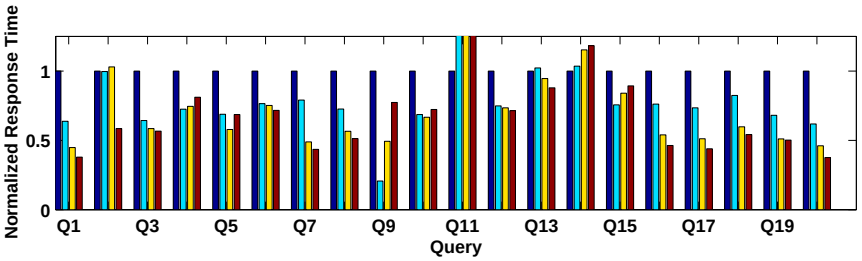

Fig. 6. For each query, we show query response times with a configuration using respectively (left to right): (i) 1-way SMP (ii) 2-way SMP; (iii) 4-way SMP; and (iv) 8-way SMP. Execution times are normalized with respect to configuration (i).

Most queries benefit when we increase number of CPUs. We see consistent response time decrease when we move from 1-way to 8-way SMP for queries Q1, Q3, Q7, Q8, Q12, Q13, Q16, Q17, Q18, Q19, and Q20. The maximum gains from increasing CPUs count are seen when we go from 1 to 4 cpus. The performance gain from 4-way to 8-way are, in general, marginal. The performance for queries Q11 and Q14 consistently degrades with number of CPUs. This is probably due to the aforementioned increase in contention. In some cases like queries Q2, Q4, Q5, Q9, Q10, and Q15, we see an optimal CPU count. Performance improves with cpu count upto the optimal point and degrades thereafter. The most common optimal point is 2 CPUs.

Finally, we vary the number of disks that are used by the database. In our setup all disks are managed by a single controller and share a single I/O bus. Hence, increasing the number of disks potentially increases the contention for these shared elements. Fig. 7 shows the benefits of I/O parallelism as the number of disks increases from 2 to 10. Most queries are seen to benefit significantly, with the exception of Q13 and Q18. More significantly, response time are lowest with the maximum number of disks.

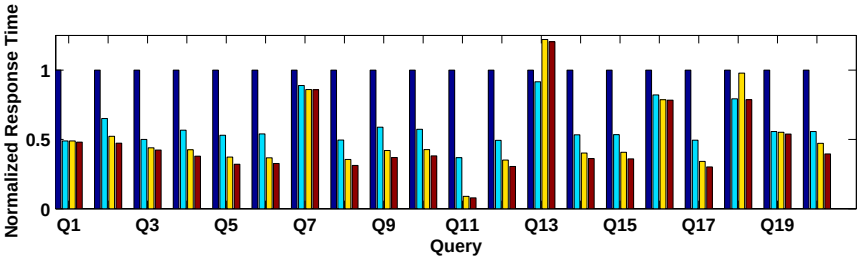

Fig. 7. For each query, we show (left to right) the query response (execution) time with a configuration using respectively: (i) 2 disks (ii) 5 disks; (iii) 8 disks; and (iv) 10 disks. Execution times are normalized with respect to configuration (i).

As in the clustered case, we have selectively examined the impact of increasing the CPUs, memory and disks that are available to the decision-support workload. We see that CPU and memory only improves performance upto a point, but disks continue to increase performance throughout the range studied. Scalability is limited beyond (and often before) 4 CPUs and 1536 MB of main memory. Hence we conclude that the workloads are fundamentally limited by the available I/O bandwidth.

6 Conclusions and future work

In this study we set out to examine the scalability of the TPC-H decision support benchmark on cluster and SMP environments. Our main focus was not on performance but on gaining an insight into the impact of various hardware parameters such as CPU, memory, disk and network. Consequently, we did not attempt to optimize the middleware or OS used in this paper.

Our results show that for both the cluster and the SMP environments, the CPU and memory resources are not major contributors to performance beyond a point. For a cluster, adding nodes improves performance even though it increases network overhead. This leads us to conclude that the decision support workload is most sensitive to an increase in I/O parallelism. The hypothesis is confirmed by the SMP results in which query response times consistently improve with an increase in the number of disks.

There is a considerable amount of work that remains to be done as part of this scalability study. We would like to expand the scope of our investigation and look into the Linux operating system to identify potential scalability inhibitors.

References

1. DB2 Universal Database. http://www-3.ibm.com/software/data/db2/udb/.
2. TPC-H Benchmark. `http://www.tpc.org/tpch/default.asp/`.
3. N. J. Boden et al. Myrinet: A Gigabit-per-second Local Area Network. *IEEE Micro*, 15(1):29–36, February 1995.