Modern Processor Performance is sufficient for Internet Services: Place your $ elsewhere.

7th March 2003

1 Introduction

"Computer technology has made incredible progress in roughly 55 years since the first general-purpose electronic computer was created. Today, less than a thousand dollars will purchase a personal computer that has more performance, more main memory, and more disk storage than a computer bought in 1980 for one million dollars."[1]On the other hand, the Internet has grown exponentially in the past decade, and today we see a wide variety of services being offered from email to peer-to-peer file sharing, web search to streaming media. Over the past two decades, the number of web portals and nodes on the Internet has grown almost exponentially. In this paper, we try to address the issue of whether or not it is worthwhile to invest in faster processors for Internet services.

The key challenges that Internet services face are high availability, incremental scalability, adaptability to frequent dynamic changes and cost effectiveness[2, 6, 5, 7, 8]. Most of these services use a cluster-based architecture to better meet these challenges. Processor performance of nodes has increased by almost 50% per year over the past 4 years. However, improvements in memory latency cannot keep up with that in processors, and the difference between the two keeps growing faster and faster, every year. Network bandwidth, on the other hand has grown from the 100 Mb to 1 Gb in just five years, and even that is not enough for today's Internet services.

Through this paper, I have made an effort to support the argument that for Internet services having a cluster-based architecture, it does make more sense to invest in improving the memory, I/O, network bandwidth, availability and overall throughput of the system rather than the processor performance of individual nodes. Some of the reasons listed in the paper are based on factual evidence, whereas others may be based on mere speculations.

2 Internet Services

2.1 Introduction

The number and variety of services offered over the Internet has grown astronomically over the past few years.The services include email services, remote storage services, e-commerce B2C and B2B services, online auctions, peer-to-peer file sharing systems, e-governance systems, instant messengers, Voice over IP services and web search, just to mention a few. Some examples are Yahoo, Google, AOL, EBay, Kazaa, MSN and Delta Three.

2.2 Requirements

Internet services have a slightly different set of requirements than traditional computer systems. The significant ones are

- Throughput: number of requests satisfied per second
- Availability: fraction of time that correct service is offered
- Scalability: ability to scale the system as the number of requests increases'
- Maintainability: ease with which changes can be made to the system

Throughput used to be *the* requirement for computer systems in the past. However, with the advent of Internet Services, availability of systems has also become a major requirement. Most services ideally like to achieve 99.999% availability. Also, as an Internet service grows older (and more famous) the number of requests grows largely with time. For eg., Google's website traffic has grown over 20% per month since the company started in 1998[12]. The service should be able to scale to meet these requirements. Last, these systems need to be changed dynamically while being available.

2.3 Architecture

Many Internet services are based on the cluster-based architecture. There is a cluster of work-stations that is used to service requests. Such a system is easily scalable, offers redundancy and therefore higher availability and easy online maintenance. In addition, these systems are cost-effective as well, since they use a cluster of inexpensive nodes for meeting their needs.

In this paper, we will discuss the argument with respect to such cluster-based systems.

2.4 Cost

In [2] , Brewer argues that hardware cost and performance are not really issues for giant-scale services. Hardware cost for these systems is typically dwarfed by bandwidth and operational costs. David Patterson et. al also mention that the total cost of ownership is typically 3.6 to 18.5 times the price of the system[7]. This indicates that the cost to operate, maintain and repair the systems is much higher than the price of these systems.

3 Bottlenecks

Brewer argues that a system's overall capacity tends to have a particular physical bottleneck, such as total I/O bandwidth, or total seeks/second, which is tied to data movement. The DQ value of the system is thus bounded by the underlying physical limitation. At the high-utilization level of these giant-scale Internet services, the DQ value approches this limitation[2].

The primary bottlenecks for most Internet services are memory and disk latency, disk bandwidth and network bandwidth. the speed of processors has been following Moore's Law over the years. However, memory performance is not able to keep up with the CPU speeds, and the difference between the two is just getting larger and larger. For eg., on the third generation of alpha processors, it takes 180 ns to service a cache miss. The clock period is 1.7 ns, and the processor can issue 6 instructions per cycle. So, effectively a single cache miss corresponds to 648 instructions! We can imagine the number of cycles that will be wasted while bringing in a page from the disk to memory for servicing a page fault!

Network bandwidth has also increased considerably in recent years. However, as the number of requests for Internet services keeps growing, even the maximum network bandwidth is not able to keep up.

4 Should one invest in better processor performance?

There are several reasons why one might think that investing in faster processors is worthwhile. We have made an attempt at giving counter-arguments to each reason below.

4.1 Faster processors mean potentially more throughput per processor, and thus an overall increase in throughput.

It is true. Faster processors might be able to pump more CPU operations per second, and hence reduce the overall time needed to service a request. How-

ever, as we know I/O (Disk) and Memory are much slower than processors. The difference between processor, memory and disk is in orders of magnitude. d!

Unless the program you are trying to run is largely CPU-bound, memory and IO are always going to be the bottlenecks. Keeping in mind Amdahl's law, increasing the processor performance by a huge factor will not influence the effective throughput significantly. On the other hand, one would expect that reducing memory and disk latency by a reasonable amount will have a significant impact on the overall performance of the system.

Network bandwidth is another bottleneck that is to be kept in mind, when one is considering Internet Services.

4.2 Many Internet services use databases, which are becoming bottlenecks. However, people are trying to come up with databases that fit in memory. For such systems, increasing processor performance should imply significant improvement in throughput. (suggested by Prof. Thu)

Commercial in-memory databases have been developed now, and have significant advantages for Internet Services. In-memory databases provide far better data access rates than traditional disk-based databases by doing away with a lot of disk operations and performing them in the RAM. For eg., these databases would be helpful for addressing the needs rapid, high-volume data delivery to run everything from shopping carts and personal preferences to targeted banner ads[11]. If such databases indeed become popular with many Internet services, then the bottleneck is likely to shift away from disk IO, if not totally towards the CPU. In that case, increasing processor performance will in fact, increase throughput proportionally.

However, the above argument is not completely valid. Firstly, this would affect scalability since bigger systems would mean larger databases that have to be stored in RAM, effectively needed larger RAMs. Since the primary purpose of using in-memory databases is to reduce the time to perform database operations, one would assume that reasonably fast (and expensive) RAMs are needed for such systems.There is a cost penalty, since no matter how cheap memory gets, disks are almost always going to be cheaper.

As stated in 2.1, the difference between memory latency and cpu performance is so high that reducing memory latency will effect in better throughtput improvement than increasing processor performance.

The databases that are used by many Internet services are very huge. For eg., Google indexes 1 billion webpages - I cannot conceive how feasible it is to move such massive databases to memory. The in-memory databases would only be useful for a very small subset of databases used by Internet services.

Thus majority of the databases will continue to be the traditional disk-based ones, which means that I/O will still continue to be a bottleneck.

On a side note, most of the vendors developing commercial in-memory databases are new and small, so companies using these databases are not guaranteed long-term support. Also, most of these products have few extra features for easy integration and management, making it more difficult for companies to migrate from the traditional databases. Due to these reasons, in-memory databases have yet to make much progress selling to Internet companies. In fact, only a handful of vendors, including Angara E-Commerce Services, Polyhedra, and TimesTen, sell in-memory databases to Web companies.

4.3 For companies that can afford the best memories, hard disks and network bandwidth, there isn't much more they can do, other than buying faster processors.

This would have been the case if you were considering a standalone PC or a non-cluster based Internet Service. However, we need to remember that throughput is not the only requirement for large-scale Internet services, availability is essential too. Many Internet Services target the ambitious five nines of availability. For cluster-based services, the total throughput and availability might be improved by adding additional nodes in the system. The assumption for improved availability here is that addition of nodes do not introduce complex dependencies in the system that reduce the availability. Let's consider a simple example:

An Internet Service has a budget of $100,000 for buying nodes. The company is considering two choices: either go for processors running at 2.0 GHz or processors running at 1.5 GHz. Both machines have the same amount of cache and memory. A typical 2.0 GHz Compaq Presario desktop cost around $2100, whereas the same configuration for a 1.5 GHz processor cost $1800 in August 2001. Remove the cost of the monitor and other peripherals from these, and the costs would be around $1800 and $1500 for the two. Thus, we now have a choice of buying either 55 machines running at 2.0 GHz, or 66 machines running at 1.5 GHz. The additional 11 nodes would mean that even though the throughput of each individual node is slightly lower, the overall throughput of the system is expected to be better than the first configuration that uses 55 2 GHz machines. At the same time, the availability of the system increases by addition of nodes, given the assumption in the previous paragraph holds.

Given the large number of transactions and the high monetary value of each transaction on some services (online stock-brokers, for eg.), every minute of downtime or unavailabilty could cost the company dearly. In that case then, wouldn't it be worthwhile to sacrifice a little performance for higher availability? There is some amount of speculation in the argument proposed here, and I have not taken into account the increased complexity of the system that has

more nodes. The exact numbers and math would depend on the particular system architecture and design, but we think that the argument is not very unreasonable.

5 If not processor performance, where should the money go?

We list a few possible places where it might be worthwhile to invest your money for Internet Services

5.1 I/O improvements

5.1.1 Faster memories

SRAMs are the slowest memories, and have been taken over by other technologies since a long time. The following table lists the costs for SDRAM, DDRAM and RDRAM (RamBus) chips of various sizes[14].

Memory Type	Size	Cost
SDRAM	256 MB	~$30
SDRAM	512 MB	~$70
SDRAM	1 GB	~$250
DDRAM	256 MB	~$40
DDRAM	512 MB	~$120
RDRAM	256 MB	$100
RDRAM	512 MB	$350
RDRAM	1 GB	~$800

SDRAM is the slowest of the three, and RDRAM is supposedly the fastest. The difference in the speeds is pretty high, and hence it might be a very good idea to use a RAMBUS on machines instead of SDRAM.

Companies are coming up with faster system interconnects to improve the cache and memory latency in processors. For eg., IBM's PowerPC970 uses an Elastic I/O interconnect that has a peak speed of 6.4 GB/s[10].

Companies that have some databases stored in memory can definitely give up some performance in terms of CPU speed for better and bigger memory systems. For eg., instead of using 2.2 GHz machines with 512 MB SDRAM chips, one can use 1.7 GHz machines with 512 MB DDRAM chips.

5.1.2 Faster and more fault-tolerant disks

Since many Internet services have databases at the back-end, faster disks would increase the overall throughput of the system.

In order to achieve higher availability, one can go for a higher RAID configuration. For eg., Instead of using RAID 5 that protects the system against one disk-failure, one can use an additional redundant disk for RAID 6 and guard against two disk failures and operator errors.

5.2 Increasing Availability

Three nines of availability corresponds to 8 hours of downtime per year, and 2 nines corresponds to 80 hours of downtime per year. Such high downtimes could cost $200,000 per hour for an Internet service like Amazon to $ 6,000,000 per hour for a stock brokerage firm[9]. This shows that high availability is very important for Internet services. In [5], the authors show that majority of the failures caused in Internet services are due to operator errors. A lot of failures are also due to hardware failures and software failures.

5.2.1 Hardware

More effort in terms of development time and cost should go into rigorous testing of each component of the system. As mentioned in 5.2, having better RAID configurations for hard disks is one way of achieving higher availability. Also, hardware developers should develop hardware for these systems keeping in mind the need for high availability. For eg., while developing the Power4 processor designed for the high cost, continuous availability market IBM traded off performance for near-absolute reliability guarantees using thicker silicon oxides.

5.2.2 Operator Training

The advent of Internet Services and their need for high availability demands that operators be treated as first class citizens. Better tools should be developed for operators to help prevent, detect and recover from failures quickly. This effectively reduces MTTR and increases availability. The ROC group at Berkeley is trying to build systems with these characteristics. There is a design methodology in place for hardware and for software. Operators for Internet Services should also be trained in a well-designed, systematic and rigorous manner so that they can perform their jobs better.

5.2.3 More nodes

As mentioned in section 4.3, one can trade off performance for larger number of nodes in the cluster. This will increase the overall peak throughput of the system. In addition, if each node is highly independent of any other node, the availability can also increase.

5.3 Cost/Performance/Availability Analysis of various subsystems

For a small Internet Service with a restricted budget, it becomes very important to analyse the cost, performance and availability of each subsytem in order to find the combination that best meets the requirements of the Service. When we started this study, we intended to quantify various combinations of CPU speeds, memories and disks in terms of both performance and availability. We realised that the analysis is beyond the scope of the paper since we weren't able to model the service architecture well and come up with a reasonable method to actually define and calculate the performability of these Internet services.

6 Conclusion

To conclude, we think that the introduction and growth of Internet services has given rise to a growing need for higher availability of these systems, and in turn higher availability for each subsytem. In order to increase the overall throughput of the system, improvements to the I/O, memory and network bottlenecks are in order. Availability is a key factor in the success of an Internet Service. More efforts in terms of time and money should go into making Internet Services more available. Once we achieve these two goals, we can think about investing in faster processors.

References

[1] J. Hennessy, D. Patterson, "Computer Architecture: A Quantitative Approach", 3rd Edition.

[2] E. Brewer, "Lessons from Giant-Scale Services", IEEE Internet Computing, July/Aug 2001.

[3] J. Gray, "Why do Computers Stop, and what can be done about it?", Proceedings Fifth Symposium on Reliability in Distributed Software and Database Systems, Jan. 1986.

[4] A. Avizienis, J. Laprie, B. Randell, "Fundamental concepts in dependability", Proceedings of the Third Information Survivability Workshop, October 2000.

[5] D. Oppenheimer, A. Ganapathi, D. Patterson, "Why Do Internet Services Fail, and What Can Be Done About It?", USITS 2003

[6] D. Oppenheimer, D. Patterson, "Architecture and Dependability of Large-Scale Internet Services", IEEE Internet Computing, Sep/Oct 2002.

[7] D. Patterson, A. Brown, P. Broadwell, G. Candea, M. Chen, J. Cutler, P. Enriquez, A. Fox, E. Kiciman, M. Merzbacher, D.Oppenheimer, N. Sastry, W. Tetzlaff, J. Traupman, N. Treuhaft, "Recovery-Oriented Computing (ROC): Motivation,Definition, Techniques, and Case Studies", UC Berkeley CS Tech Report UCB//CSD-02-1175, March 15, 2002.

[8] A. Fox, S. D. Gribble, Y. Chawathe, E. A. Brewer, Paul Gauthier, "Cluster-Based Scalable Network Services", SOSP 1997.

[9] Kembel R., "FibreChannel: A Comprehensive Introduction", p8.2000

[10] http://www.realworldtech.com, "Announcement of the IBM PowerPC 970 Processor"

[11] http://www.informationweek.com, "In-Memory Databases Aid Web Customization"

[12] http://www.linuxgazette.com/issue59/korrea.html, "Interview with Google's Seargey Brin"

[13] http://www.zdnet.com, "2 GHz is nice, but saving money is even nicer"

[14] http://www.price.com