Characterization of an IMAP Server on a Shared-Memory Multiprocessor

Pranay Koka, Mikko H. Lipasti

Dept of Electrical and Computer Engineering
University of Wisconsin-Madison, WI 53706
{pkoka,mikko}@ece.wisc.edu

Abstract

Modern commercial server systems are employed for a diverse set of applications. Prior architectural characterization research has focused primarily on transaction processing, decision support, web serving and application serving workloads, and has identified three common workload characteristics: large instruction working sets, high cache miss rates, and a large fraction of dirty misses in multiprocessor systems. This paper studies an additional class of commercial workload—email serving—that is widely deployed yet poorly understood. In addition to studying the standard SPECmail2001 POP3-based workload, we define and characterize a new IMAP-based transactional workload that models interactive e-mail users in various configurations under two diverse transaction mixes. On an 18 processor IBM S80 shared-memory multi-processor, we find that these workloads, though extremely CPU intensive, differ dramatically from other commercial workloads due to their smaller instruction working sets, reduced cache miss rates, and less frequent dirty misses in multiprocessor systems.

1. Introduction

Traditionally, server systems were evaluated using scientific and engineering workloads like SPECint95 and Splash [20]. In recent years an increasing number of server systems were deployed to process commercial applications. Prior research shows contrasting characteristics between commercial and scientific workloads [11]. Commercial applications are mostly multi-programmed and exhibit random I/O behavior. Due to large amounts of I/O, commercial workloads have high process switching rates. Unlike technical applications, which are more iteration based with tight loops, commercial applications have very few loops and hence fewer loop branches. Due to these differences, commercial applications place an entirely new set of requirements on the server systems, driving the need to study these workloads. Because of the diverse nature of these applications, development of a set of benchmarks representing the entire commercial application domain has become impossible. Database workloads form a major portion of the server market. Multi-tier business applications and online transaction processing systems are another class of applications that dominate the server market. Considerable effort has been put in by academia and industry consortia to develop and characterize commercial workloads. The Transaction Processing Performance Council (TPC) is one such consortium that has developed a range of data base workloads [18]. TPC-A, TPC-B, TPC-C, TPC-D, TPC-H, and TPC-R are some of the standard benchmarks developed by TPC that represent transaction processing and decision support systems. TPC-W is another benchmark that represents an e-commerce workload modeling an online bookstore. Prior research work has characterized and studied these workloads on a variety of processor models from simple in-order cores to more aggressive out of order processors [6,14]. Due to the large data set sizes and irregular data access patterns, a significant body of research work has studied the memory system performance of these workloads [1,17]. Some of the memory characteristics studied are spatial/temporal locality and sharing patterns. Research has established that commercial workloads display poor cache performance leading to long stalls while accessing memory. Simultaneous multi-threaded processors, which are known to hide memory latencies [19], are well-suited for these applications [10]. With the emergence of Java as an important application development platform, e-commerce systems developed in Java have been characterized [2,7]. Some of the published work has also evaluated existing server machines using commercial workloads [4,5,9]

Systems Performance Evaluation Cooperative (SPEC), another consortium, has developed a set of commercial workloads that includes the, Java business benchmark (SPECjbb2000), Static and dynamic web content delivery benchmark (SPECweb99) and the Java application server (SPECjAppServer2002) [16].

Another emerging class of application is the electronic mail service. The worldwide email volume which is around 31 billion messages a day is expected to double by 2006. Substantial numbers of server systems are being deployed to cater to email needs. The increased email usage in conjunction with authentication, encryption, and spam filtering has placed high performance requirements on commercial

Transaction Type	Description	Browsing	E-mgmt
CRTE	Create a mailbox	**0.5%**	**10%**
DELT	Delete a mailbox	**0.5%**	**6%**
RENM	Rename a mailbox	**0.5%**	**6%**
SRCH	Search a message by subject	**4.5%**	**17%**
SELT	Select a mailbox and fetch headers of all emails	**25%**	**5.5%**
COPY	Copy a message from current mailbox to a random mailbox	**3.5%**	**5.5%**
MOVE	Move a message from current mailbox to a random mailbox	**6.5%**	**3%**
VIEW	Fetch entire body of a message	**32%**	**10%**
SEND	Send a message to a random user	**21%**	**12%**
MDEL	Mark delete flag on a message and expunge the message	**6%**	**25%**

Table 1. Description of Mail server transactions

	CRTE	DELT	RENM	SRCH	SELT	COPY	MOVE	VIEW	SEND	MDEL
CRTE	1/1	1/1	1/7	7/21	39/9	5/15	5/25	10/5	25/2	6/14
DELT	1/1	1/1	1/1	5/15	21/1	5/15	5/25	50/0	11/6	0/35
RENM	1/1	1/1	1/1	5/5	21/1	5/25	5/25	10/10	45/5	6/26
SRCH	0.5/0.5	0.5/0.5	0.5/0.5	5/15	43/3	2/32	5/25	28/2	0/6	15.5/15.5
SELT	0.5/7.5	1 / 4	0.5/0.5	4/24	4/4	4/23	3/23	39/0	35/13	10/1
COPY	0.5/0.5	0.5/5.5	0.5/2	4/10	42/30	1/3	1/17	19/2	21.5/8	10/20
MOVE	0.5/2.5	0.5/9.5	0.5/4.5	4/1.5	42/9.5	2.5/4.5	3.5/41.5	25/5	16/4	5.5/17.5
VIEW	0.5/0.5	0.5/0.5	0.5/0.5	6/6	25/25	5/5	2/2	35/35	25.5/5.5	0/20
SEND	0.5/0.5	0.5/0.5	0.5/0.5	1/1	15/15	2/2	25/25	15/15	25/25	15.5/15.5
MDEL	0.5/21	0.5/10	0.5/5.5	5/12	10/1	0.5/17.5	7/17	50/0	16/2	10/14

Table 2. State transition probabilities in percentages for browsing and e-mgmt mixes.

server machines. SPEC has taken the first step towards characterizing mail server workloads. SPECmail2001 is a benchmark that models a mail server system with the Post Office Protocol 3 (POP3) [12] as the mail access protocol. A Java based client load generator is used to emulate POP3 clients. The benchmark also emulates a real world scenario of mail exchange between local and remote servers using SMTP servers and SMTP sinks. POP3 has established itself as the father of all standardized mail access protocols. It allows only the basic functionality of message retrieval from the server using POP3 clients. All mail operations are performed by the mail client. Such a simple message retrieval systems restricts the users to a single client machine.This has led to the development of Internet Message Access Protocol (IMAP) [13]. IMAP shifts the onus of managing emails to the server side, allows users to start multiple simultaneous connections and also permits mailbox sharing. It supports various operations like searching and moving messages on the server side. Owing to its inherent advantages over POP, IMAP is gaining popularity in the email service community. This transition to processing emails on the server side places new demands on server machines.

As of now, no IMAP benchmark has been proposed or has been used to evaluate a server machine. In this paper we propose and characterize an IMAP workload on an 18 processor IBM RS/6000 S80 SMP system [8]. In section 2 we describe our workload and its parameters. In section 3 we present the characterization results and finally conclude in section 4.

2 Workload description

There are a number of IMAP implementations available in the market. Some of the popular ones are UW-IMAP server, Cyrus IMAP server, IBM Lotus Notes and Microsoft Exchange Server. The University of Washington IMAP server is an open source reference implementation of IMAP written by Mark Crispin, the inventor of IMAP. It is popular for its ease of administration, flexibility and compatibility with existing mailbox formats. For the above reasons we chose to use UW-IMAP server for this workload. IMAP supports three modes of connectivity – Online, Offline and Disconnected. Online mode provides an interactive session to the user to perform email operations. The offline mode has the same functionality as that of POP. In the disconnected mode the IMAP client connects to the server, synchronizes the mailboxes and then disconnects. We have developed a load generator that emulates multiple IMAP clients with a single online session per client.

The setup used for this characterization includes mail exchange with remote mail servers. The workload load generator simulates the external mail server for incoming messages. All messages sent by the load generator are destined to the local domain. We define 10 different types of transactions that are performed by the emulated clients on messages and mailboxes. Some of the transactions, as listed in Table 1 have a one-to-one mapping to the IMAP commands. We classify the mail server interactions as two mixes based on the type of usage. Each mix varies from the other in the frequencies of different kinds of transactions performed. The transaction frequencies for each mix shown in Table 1 are assigned based on the type of transactions that dominate a mix. The purpose of this classification is to study the dependence of the workload characteristics on the kind of mailbox usage. The browsing mix emulates a scenario in which 76% of the transactions involve selecting mailboxes, reading messages and responding to emails. The mail management mix (e-mgmt) emulates management of mailboxes and messages with 60% of the interactions comprising of message deletion, moving and some mailbox operations.

The load generator is driven by a state transition table presented in Table 2. Each element *b/e* shows the probability of transition from state X to state Y for each of browsing and e-mgmt mixes respectively. The initial mailbox sizes were in the order of 4MB, with an average of 6 mailboxes per user. We use the Unix mailbox format for the mail folders. The number of emails in a mailbox and the number of mailboxes are allowed to vary between preset thresholds.

3. IMAP Workload Characterization

3.1 Throughput and Response Time

In this section we present the evaluation of the workload using an appropriate performance metric and subsequently present the characterization results collected by running the workload on an 18-processor IBM RS/6000 S80 SMP system in the following sections. Detailed system parameters are tabulated in Table 3. We concentrate more on the memory system characteristics and also present an instruction profile of the workload. We also compare the characterization results of the IMAP workload with that of other commercial benchmarks. The performance metric we use to evaluate the workload is the number of mail transactions performed per second. We have measured the throughput of this workload for both the types of mixes with a varying number of emulated clients. Think time for the clients was set to zero for the throughput measurements. In the figures that follow and in the relevant discussion, we refer to the browsing mix as *imapb*, and the e-mgmt mix as imap*e*.

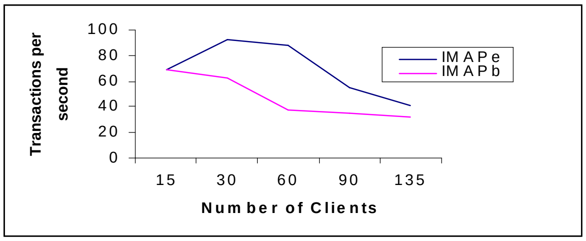

Figure 1. Throughput in transactions per second

Figure 1 shows the measured throughput for the two mixes. IMAPe has a higher throughput than the browsing mix. Browsing mix consists mostly of view and select transactions. A select transaction is comprised of an IMAP select command followed by a series of header fetch commands for all the existing messages in the mailbox. The high frequency of fetch commands in this mix results in a large volume of data to be transferred from the disk reducing the throughput. The E-mgmt mix exhibits higher throughput, than the browsing mix but the throughput drops steeply beyond 90 clients. With the help of system monitoring tools we found that beyond 90 clients, the disks were flooded with a large number of small transactions. This kept the disks busy seeking for most of the time, reducing the throughput.

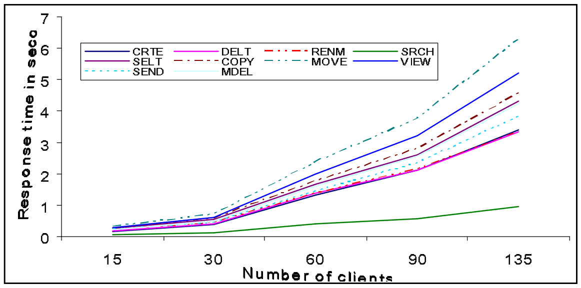

Figure 2. Response times of message transactions for IMAPb Mix

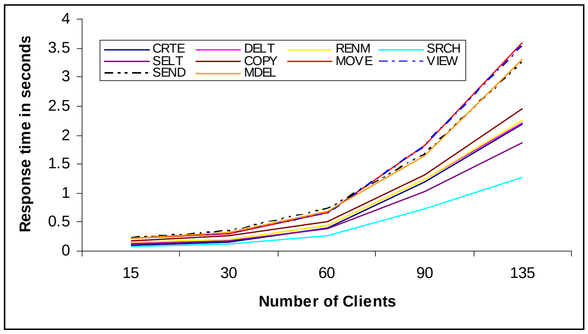

Figure 3. Response time of message transactions for IMAPe mix

Figures 2 and 3 show the response times for different types of transactions. In both the mixes, the move transaction is the slowest. The primary contributor to the transaction latency is the disk I/O. Hence, the search transaction which is more CPU bound is the fastest. The response time of the browsing mix is higher than that of the management mix.

3.1 Workloads

We present the native execution results of three other commercial applications - SPECjbb2000, SPECweb99ssl and SPECmail2001 for a comparison with the results of the IMAP workload.

SPECjbb2000 is a three tier business application written in java. It concentrates more on the middle tier which comprises the business logic. The backend database is emulated using binary trees. It is based on the TPC-C specification and models a wholesale company with many warehouses. All the three tiers run under the same java virtual machine. For the results presented, we use 32 warehouses.

SPECweb99_ssl is a web content delivery benchmark for evaluating the performance of HTTP 1.0/1.1 web servers over the secure sockets layer. For our measurements, we use Apache-SSL with 512 simultaneous connections. We have also increased the percentage of dynamic content to 80% of the pages served in-order to increase the CPU utilization.

SPECmail2001 is a mail server benchmark to evaluate mail server systems based on SMTP and POP3. It emulates a realistic mail server scenario with peak hour simulation, modem simulation and arrival rates for messages. For our measurements we use 5000 users with 40 POP checks per day per user.

3.2 Methodology

For characterizing the workload we use an IBM RS/6000 S80 multiprocessor system, with 18 RS-64 III (Pulsar) processors [3,15]. Pulsar is a 5-stage 4-issue in-order superscalar processor. The other system parameters are tabulated in Table 3. We collected performance data using the on-chip performance counters in the RS 64 pulsar processor. Pulsar has a rich set of eight performance counters which can measure 275 unique events.

Hardware	
Processors	18-way 450 MHz RS-64 III (Pulsar
Memory	18GB
L2 Cache	8MB unified cache per processor
L1 Cache	128 KB I-Cache, 128KB D-Cache
Disk configuration	All disks are 9.1GB, mix of SCSI, SSA. Mail folders are manually striped across multiple disks. Mailbox Lock files are placed in a RAMfs.
Software	
IBM JDK 1.1.8, AIX 4.3.3	Apache_1.3.27+ssl_1.48
UW IMAP, POP3 server	Compiler: gcc 2.95.2

Table 3. System parameters

We used the process tree counting mode provided by the performance monitoring software. Since the UW-IMAP server runs behind the internet super daemon, the same set of counters measure the characteristics of all the IMAP children forked by the super server.

To characterize the workload in tune with the current email service scenario, we use SSL encryption for IMAP client server communication. We run the load generator on the local machine in order to reduce the network latency and achieve maximum CPU utilization on the machine. We believe that the noise introduced in our measurements by the load generator is negligible as the load generator uses a maximum of 2% of all 18 processors. We used 200 emulated clients for both the mixes with a think time of four seconds. All the execution runs consisted of a five minute warm-up phase followed by a 20 minute measurement phase.

3.3 CPU Utilization and Speedup

We split the utilization into user and kernel components. For both the mixes, 15% of the time is spent in the kernel. We achieved 95% overall utilization on 18 processors for both the mixes. In the graphs that follow, the browsing and e-mgmt IMAP mixes are referred to as *imapb* and *imape*, SPECweb99_ssl as *web*, SPECjbb as *jbb* and SPECmail as *mail*. Figure 5 shows the scaleup curve of the workload on a multiprocessor system. Speedup is measured over five processor configurations with a fixed load of 90 clients. For measurements involving more than 18 processors, the system was scaled down by virtually shutting out processors using spin loops. There is negligible speedup from 8 to 16 processors because the smaller configurations were overwhelmed by the load. This was evident from the unrealistic response times, which were in the order of 15 to 20 seconds per transaction.

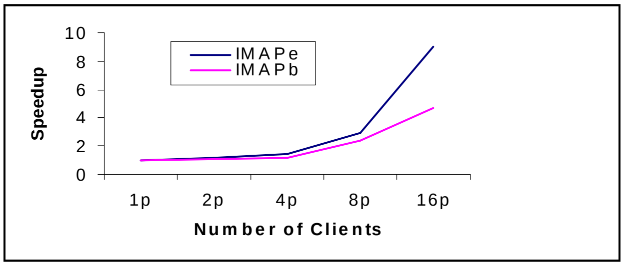

Figure 5. Scaleup curve

3.4 Memory System Characterization

In Figure 6 we present the contribution of the L2 stalls towards the CPI. All the workloads studied exhibit an infinite cache CPI of nearly one. With a L2 miss rate per instruction of only 0.9%, SPECmail2001 experiences the least amount of stalls due to L2 misses. The kernel component of SPECmail experiences a miss rate of 24% and hence an increase in CPI by 0.73 due to L2 stalls. Since SPECmail spends only 4.4% of the CPU time in the kernel, the higher kernel CPI has almost no influence on the overall CPI. SPECjbb and SPECweb have a L2 miss rate of 40% and 11%, respectively. The L2 stalls in these two benchmarks make significant contribution towards CPI. Both the IMAP mixes exhibit more or less the same CPI for both user and kernel components. The user component runs nearly at infinite cache CPI. The kernel transfers the message data to the user address space and due to the large data set size experiences more L2 misses. The IMAP mixes spend 15% of the time in the kernel and hence the kernel does have some influence on the over all CPI. The greater fraction of kernel time in the IMAP workload as compared to that of POP is due to the rich set of functionality implemented by IMAP that causes more system call activity.

We also analyzed the time spent by the IMAP workload by gathering function call traces using *tprof*. 84% of the user component time is spent in the function *unix_mbxline()*. This function reads a line from the mailbox and in the process performs string manipulation and character-at-a-time scan of the mailbox lines. This string parsing activity is mainly due to the text format of the mailbox. Other mailbox formats like a database (used by Lotus Notes and MS-Exchange server) could help save the time spent on string parsing. Only 2% of the user component is spent in SSL routines.

In Figure 7 we show the breakdown of L2 stall cycles into stalls due to data load and store misses, instruction misses and PTEG (Page table entry group) misses. PTEG misses are the translation misses in L2 when the hardware page table handler of PowerPC loads the page table entry groups into L2 during a page table walk. All the workloads experience more store miss stalls than those due to loads, with SPECmail experiencing the maximum. In all IMAP mixes 20% of the stalls are translation stalls. Instruction stalls are comparatively lesser and contribute to an average of 10-15% across all the workloads.

The S80 system uses the MOESI cache coherence protocol. Hence on a cache miss, the miss can be serviced from either memory, or other caches having the block in Modified, Owned, Exclusive, or Shared state. In Figure 8 we show the breakdown of L2 cache misses based on from where the miss was serviced. Cache-to-cache transfers are infrequent in the IMAP mix with 75% to 85% of the misses being serviced from memory. SPECmail has contrasting characteristics in this respect. 85% of the misses for SPECmail are serviced from other caches in M, O or E-state. But this is not an indicator of significant data sharing as the miss rate experienced by SPECmail is as low as 0.9%. The sharing seen in Figure 8 is primarily from kernel data structures and filesystem metadata. Approximately 5% of the misses for SPECmail and SPECjbb are store misses serviced from other caches in the modified state. This component is almost absent in other workloads.

3.5 Impact of the Exclusive state

The importance of the Exclusive state has been a long standing debate in the research community. In the MSI protocol, a transition of the cache block from the shared state to the modified state causes a bus upgrade transaction. The Introduction of the E-state saves a bus upgrade transaction on every E->M transition. Hence an effective E-state would mean saving bus bandwidth with fewer bus upgrades. Recent research work [4,9] has evaluated the importance of the E-state by determining the percentage of stores that find the block in L2 in the exclusive state. Authors in [7] argue the limitations of this approach and propose a better method for the same. They evaluate the importance of the E-state by finding the percentage of blocks that come into the L2 cache in the E-state and transition into M state. As shown in Figure 9, 13% to 84% of the cache blocks

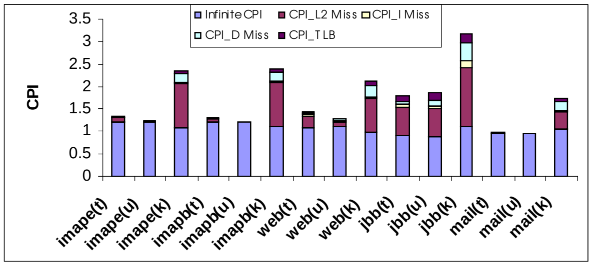

Figure 6 CPI breakdown – shows the memory system contribution to the CPI

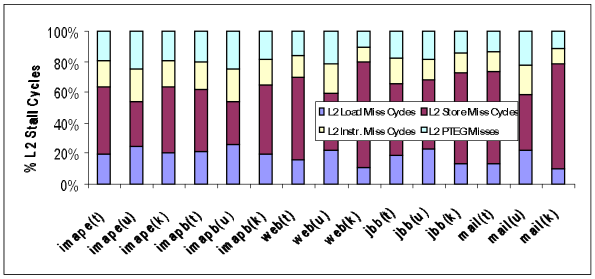

Figure 7. Breakdown of stall cycles due to L2 cache miss

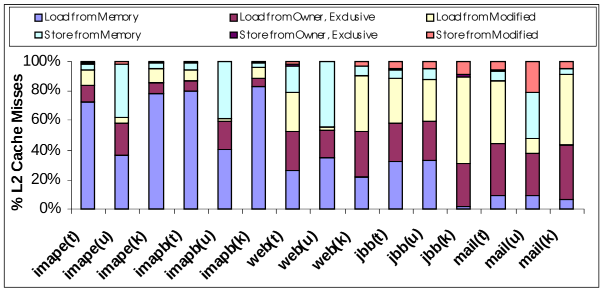

Figure 8. Breakdown of L2 cache misses based on the source servicing the mix.

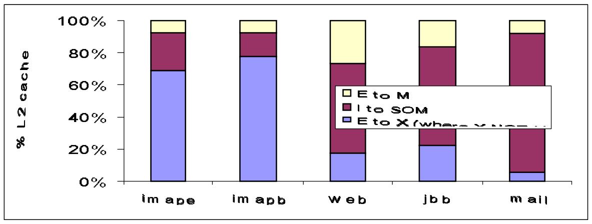

Figure 9. Impact of E-state

enter in the E-state and around 10% to 61% of these transit to the M-state. IMAP has the minimum number of blocks transiting from the E to the M state. The E-state is most useful in SPECweb where 44% enter the E-state and 61% of these transit to the M-state.

3.6 Instruction profile

Figure 10 shows the instruction distribution for all the workloads. We classify the instructions into load/store, arithmetic and logic, unconditional branches, conditional taken and not taken branches. In the IMAP mixes, 60% of the instructions are loads and stores. SPECjbb and SPECweb execute more ALU iinstructions. SPECmail has a contrasting instruction mix compared to IMAP. Branch instructions form 40% of the instruction mix of SPECmail and only 11% of that of IMAP.

4. Conclusion and Future Work

We present and characterize the IMAP server workload. We show that IMAP workload has a smaller instruction working set when compared to other commercial benchmarks like SPECjbb. Data cache misses are also less when compared to other commercial workloads. We show that IMAP also differs in terms of the effectiveness of the Exclusive state, which is less effective in the case of IMAP. We saw that IMAP characteristics differ from that of POP in terms of the instruction mix. Also similar results for both the mixes show that the type of mailbox usage does not significantly change the workload characteristic. From our characterization results we conclude that IMAP server workload dramatically differs from other commercial workloads.

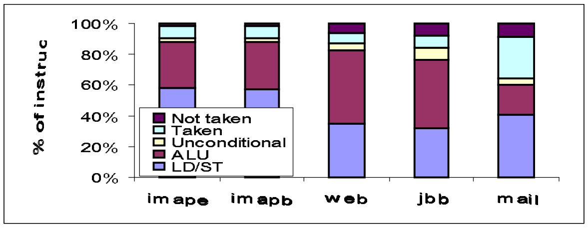

Figure 10. Instruction Profile

A comparison of UW-IMAP with other IMAP implementations that use different mailbox formats would bring out interesting results, especially in the work performed in parsing mail text. We propose to perform a full system simulation study of this workload for a more complete characterization. Using an optimizing compiler would be another area we propose to investigate. After a more detailed characterization we plan to release our implementation to the public domain in the near future.

References

[1] L. Barroso, K. Gharacharloo, and E. Bugnion, Memory System Characterization of Commercial Workloads. In *Proceedings of the 26th Annual International Symposium on Computer Architecture*, May 1999.

[2] T. Bezenek, H.Cain, R.Dickson, T.Heil, M.Martin, C.McCurdy, R.Rajwar, E. Weglarz, C.Zilles, and M. Lipasti. Characterizing a Java Implementation of TPC-W. Third Workshop On Computer Architecture Evaluation Using Commercial Workloads, January 2000.

[3] J.Borkenhagen, and S. Storino. 5th Generation 64-bit PowerPC-Compatible Commercial Processor Design. IBM whitepaper, http://www.rs6000.ibm.com, 1999.

[4] Q. Cao, P. Trancoso, J,L. Larriba-Pey, J. Torrellas, R. Knighten, and Y. Won. Detailed characterization of a quad Pentium Pro server running TPC-D. In *Proceedings of the Third International Symposium on High Performance Computer Architecture,* February 1997

[5] Z.Cventanovic and D. Bhandarkar. Performance characterization of the Alpha 21164 microprocessor using TP and SPEC-workloads. In *proc' of the 21st Annual Symposium on Computer Architecture,* pages 60-70, Apr 1994.

[6] R. J. Eickemeyer, R.E Johnson, S.R. Kunkel, M.S. Squillante, and S. Lie. Evaluation of Multithreaded Uniprocessors for Commercial Application Environments. In *Proc' of the 23rd Annual Symposium on Computer Architecture,* May 1996

[7] Harold W. Cain, Ravi Rajwar, Morris Marden and Mikko H. Lipasti. An Architectural Evaluation of Java TPC-W. In *The Seventh International Symposium on High-Performance Computer Architecture,* January 2001.

[8] IBM Corporation. The RS/6000 Enterprise Server Model S80 Technology and Architecture. IBM Whitepaper available from http://www.rs6000.ibm.com

[9] K.Keeton, D.Patterson, Y.He, R.Raphael, and W.Baker. Performance Characterization of a Quad Pentium Pro SMP using OLTP Workloads. In *Proc' of the 25th Annual International Symposium on Computer Architecture,* June 1998

[10] J.Lo,L.A.Barroso, S.Eggers, K.Gharacharloo, H. Levy, and S. Parekh. An Analysis of Database Workload Performance on Simultaneous Multithreaded Processors. In *Proceedings of the 25th Annual International Symposium on Computer Architecture*, June 1998

[11] A.Maynard, C. Donnelly, and B. Olszewski. Contrasting characteristics and cache performance of technical and multi-user commercial workloads. In *proc' ASPLOS,* pages 145-156, October 1994.

[12] Network Working Group RFC 1939 – Post Office Protocol

[13] Network Working Group RFC 3501 – Internet Message Access Protocol – Version 4rev1

[14] P. Ranganathan, K.Gharacharloo, S. Adve and L.Barraoso. Performance of database workloads on shared-memory systems with out-of-order processors. In *Proc'of ASPLOS-VIII*, October 1998.

[15] S. Storino, A. Aipperspach, J. Borkenhagen, R. Eickmeyer, S. Kunkel, S. Levenstein, and G. Uhlmann. A Commercial Multi-threaded RISC Processor. In *International Solid-State Circuits Conference*, 1998.

[16] Systems Performance Evaluation Cooperative. SPEC Benchmarks. http://www.spec.org.

[17] P. Trancoso, J.-L. larriba-Pey, Z. Zhang and J. Torrellas. The memory performance of DSS commercial workloads in shared memory multiprocessors. In *Third International Symposium on High-Performance ComputerArchitecuture*,

[18] Transaction Processing Performance Council. TPC Benchmarks. http://www.tpc.org.

[19] D.M. Tullsen, S.J.Eggers, and H.M. Levy. Simultaneous Multithreading: Maximizing OnChip Parallelism. In *Proceedings of the 22nd Annual International Symposium On Computer Architecture*, pages 392-403, June 1995.

[20] S.C. Woo, M. Ohara, E. Torrie, J.P. Singh, and A. Gupta. The SPLASH-2 programs: Characterization and methodological considerations. In *Proceedings of the 22nd International Symposium On Computer Architecture*, pages 24-36, June 1995.