Paper 286-28

SAS® Performance Optimizations on Intel Architecture

Patrick J. Fay, Intel Corporation, Cary, North Carolina, USA
Tracy Warren Carver, SAS Institute Inc., Cary, North Carolina, USA

ABSTRACT

This paper is intended for intermediate to advanced SAS users who want to know more about performance on Intel architecture-based systems. In early 2001, SAS and Intel jointly established the SAS and Intel Advanced Research Center (SIARC). Learn how SIARC research is used to optimize SAS performance on the latest 32-bit and 64-bit Intel processors. Learn about new Intel technologies that will affect you, such as the Intel® NetBurst™ microarchitecture, the Intel® Itanium® processor family, and Hyper-Threading Technology (HT Technology)[1]. Learn how SAS performance is enhanced by these technologies. Learn about the characteristics that can affect SAS performance in areas ranging from SAS/STAT® and Enterprise Miner™ analytical procedures and traditional SAS/Base procedures.

SAS performance improvements are significant and many optimizations have been included in the SAS System for Windows*, Version 9.0.

INTRODUCTION

Recent developments in Intel architecture have permitted some astonishing leaps forward in performance. New designs present some fascinating opportunities for Independent Software Vendors (ISVs) who want to take advantage of these designs. There are two different major fronts to understand, each with their own performance characteristics.

INTEL® NETBURST™ MICROARCHITECTURE

The Intel NetBurst microarchitecture is an internal redesign of the IA-32 processor, based on the x86 instruction set. It is fully compatible with programs written for previous 32-bit Intel processors. This microarchitecture is used for the Intel® Xeon™ processor as well as the Pentium® 4 processor.

INTEL® ITANIUM® PROCESSOR FAMILY

Intel Itanium processors are 64-bit processors that use Explicitly Parallel Instruction Computing (EPIC), allowing simultaneous execution of multiple instructions. Predication and speculation techniques are used heavily to optimize performance.

HISTORY

Each of these technologies have presented interesting opportunities for SAS Research and Development. Intel and SAS have worked together closely for several years in order to move SAS forward on the Itanium Processor Family. This was especially challenging to Windows-based developers at SAS because the Itanium has a different instruction set from the traditional “x86” instruction set used on 32-bit Windows. At the same time, Intel has continued to enhance its flagship 32-bit offerings, moving from the Pentium® Pro processor to the Pentium® II processor and then the Pentium® III processor. However, recognizing limitations in the “P6” series, Intel opted for a processor redesign for the Intel Xeon processors and the Pentium 4 processor. To make matters even more interesting, SAS Project Mercury was in full swing, allowing MVA SAS to take full advantage of threads for the first time. Coincidentally, Intel was adding HT Technology to the Xeon processor, allowing a single physical processor to run 2 threads truly concurrently.

Recognizing that the new designs required thorough and organized work to understand their performance characteristics, SAS and Intel teamed up to form the SAS and Intel Advanced Research Center. The SIARC group studies the performance characteristics of the new processors, specifically in the context of the SAS System on Version 9. Findings from SIARC are fed into SAS R&D and Intel R&D to improve products from both companies.

This paper discusses the most important factors of each processor that affects SAS performance, and highlights SIARC solutions to these issues. First the Intel NetBurst microarchitecture is discussed in some detail, and next the Intel® Itanium processor family.

THE NEW ARCHITECTURES DEFINED

Here we discuss the two new processor designs in just enough detail so you can understand their basic definitions, and their most important characteristics with regards to SAS performance.

INTEL® NETBURST™ MICROARCHITECTURE IN DETAIL

IA32 processors are based on the "x86" instruction set. Most of the instructions on an Intel 32-bit processor can be traced to the original Intel 8086/8088 microprocessor designed in the late 1970s. Features of the IA32 processors, most often used under Windows, include a set of about 5 main registers, a floating point stack, and numerous assembly instructions. The IA32 processors are historically considered to be CISC processors (Complex Instruction Set Computing), though the definition is debatable.

The microarchitecture of a processor is the internal implementation “in silicon”. Processors manufactured in different years are completely compatible, but the later processors will typically have an enhanced microarchitecture as well as a somewhat increased MHz rating. There may occasionally be some new assembly instructions added for performance reasons, but these will not be backwards compatible. SAS only uses new instructions if they can be used in such a way as to provide the same output as a previous implementation.

The most important fact about the Intel NetBurst microarchitecture is that it is a redesign of the internals of the IA32 processors. Although there are some new assembly instructions added, the most interesting effect is that the Instructions Per Clock (IPC) are not the same as with previous processors for a given set of binaries. In some cases the IPC decreases. However, the redesign permits dramatic increases in the MHz ratings which supersede differences in IPC. Each new release of a Pentium 4 or Xeon processor typically offers a 100 to 200 MHz increase. In contrast, new “P6” (Pentium II and Pentium III processors) chips typically increased by only 30 to 40 MHz.

The most important characteristics of the Intel NetBurst microarchitecture are:

- Hyper Pipelined Technology – pipeline over twice as deep as P6 generation. More instructions flow through the processor more quickly. The downside is that the performance penalty for a branch mis-predict is more severe and the IPC rate may be lower. To compensate, Intel greatly improved its branch prediction solution.
- 400 MHz (and above) System Bus – this means a 100 MHz Front Side Bus that is “quad-pumped” to allow a 4x increase in the amount of data flowing through the system. The performance can peak at 3.2 GB per second of data flowing in and out of the micro-processor.
- Advanced Dynamic Execution – the Execution Unit can choose from a large window of instructions in the queue to

execute. This allows for more out-of-order execution and takes advantage of the large number of internal registers (at least 128).

One may think of Intel processors with the Intel NetBurst microarchitecture as chips with a CISC shell, but a RISC inside (Reduced Instruction Set Computing).

An interesting side-effect of the new architecture is that in many cases hand-tuned assembly code is not as desirable nor as necessary as it has been on “P6” systems. In some cases legacy assembly code can impede performance on Intel NetBurst microarchitecture.

INTEL® ITANIUM® PROCESSOR FAMILY IN DETAIL

The Itanium processor family is a series of 64-bit processors supporting a flat address space using 64-bit pointers. For the largest data processing jobs, there is no viable alternative to this solution. Although mature 32-bit CISC and RISC systems may offer fine performance for most things, ultimately there is no choice but to support 64-bit processing to solve the largest memory-bound problems.

The Itanium processor family was co-designed with Hewlett Packard. Overtly, many of the instructions in IPF are similar or identical to HP's PA-RISC. EPIC (Explicitly Parallel Instruction Computing) is the core feature of IPF that distinguishes it from RISC. EPIC uses predication, speculation, and explicit parallelism to surpass traditional RISC architectures. EPIC compilers use architectural information (such as latencies to cache and main memory, and the number and size of internal execution units) to implement parallelism within an application.

A key principle of EPIC and RISC is that the compiler should do the hard work of performance optimization. Although hand-tuned assembly code has been quite common for SAS in IA32, in IPF it is definitely not desirable. It is best to let the compiler do the hard work, and understand the compiler options needed to achieve performance. The most important characteristics of IPF are:

- EPIC (Explicitly Parallel Instruction Computing) – the compiler does all the hard work. It schedules groups of instructions that can execute concurrently if the conditions are right.
- Register Stack Engine (RSE) – over 128 integer and floating point registers are exposed at the assembly language level. As functions are called, a rotating register mechanism is used to make the function call without allocating memory on the processor stack in most cases.
- 400 MHz (and above) System Bus – The performance can peak at 6.4 GB per second of data flowing in and out of the Itanium 2 processor.
- 64-Bit Flat Address Space – Memory addressability goes from 4 Gigabytes (32-bit) to a theoretical maximum of 16 Terabytes.

PERFORMANCE OPTIMIZATIONS WITH SAS ON THE NEW ARCHITECTURES

Here we discuss performance characteristics encountered on the new Intel designs, and how SIARC has addressed them. Throughout this discussion, keep in mind certain points:

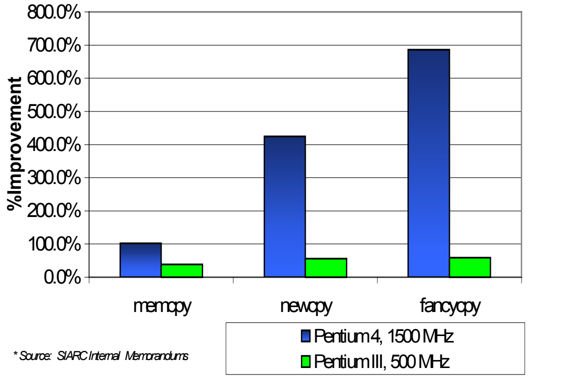
- Remember that the classic standard bottlenecks are CPU, memory, and I/O, and learn about the various aspects of each bottleneck.
- Good performance means making maximum use of the resources of the system. You should strive to understand those resources and understand the documented system limits.
- When a resource becomes exhausted or maximized, you will be at a boundary condition. Additional loads past the boundary condition are likely to show very different performance characteristics of your system.
- When analyzing bottlenecks and tuning, we look to see how well the performance measures up to the documented system capabilities. For instance, in an I/O performance test that reads a large SAS data set, we typically look at the specification sheets for the I/O controllers & disk drives and ensure that we approach the specs.

PERFORMANCE OPTIMIZATIONS ON INTEL® NETBURST™ MICROARCHITECTURE

In order to take advantage of the Intel NetBurst microarchitecture, SAS Version 9 was re-worked in several key areas to achieve the best possible performance. These were:

- Version 8 and Version 6 SAS toggled exception mask bits in the processor Control Word when performing operations such as rounding a double-precision number to an integer. In general these toggles were unnecessary and were found in legacy assembly code. On the Intel NetBurst microarchitecture, they cause a pipeline flush which stalls the processor significantly. This is documented in the Intel document, *Desktop Performance and Optimization for Pentium 4 Processor.* This is very important in SAS DATA steps that call SAS functions or that use DATA step arrays. This was reworked in SAS Version 9.0.
- Memory utility routines were rewritten. These are common C/C++ functions such as `memcpy`, `memset`, `memcmp`, and `memmove`. As originally implemented by the SAS in-house compiler used for the 32-bit SAS System for Windows, these routines frequently used the `REP`-prefix instructions such `as REP MOVSB`, `REP MOVSW`, and `REP CMPSB`. The IPC performance of these instructions is less than “P6” family processors when the memory block is less than 32 bytes. The latest Intel and Microsoft compilers sometimes use alternatives to these instructions. By avoiding these instructions, performance increases significantly. In addition to dropping the `REP`-prefix instructions, the SSE2 instructions are used to get the best performance for blocks over 32 bytes in length. Patrick J. Fay of Intel Corporation, a member of the SIARC team, wrote these new memory utility routines. These routines are very important in traditional Base procedures as well as in many DATA steps. The new routines allow Version 9.0 SAS to take full advantage of the increased memory bandwidth of the Intel NetBurst microarchitecture. Figure 1 shows the performance increases measured with several alternative approaches to the SAS memory utility routines for variable length cases[2]. See the Appendix for machine configuration information.

Figure 1
Performance Increases For 3 Approaches To Memcpy Versus Original SAS Memcpy For Variable Length Blocks *



** Source: SIARC Internal Memorandums*

- String utility routines were rewritten. These were typically originally written in legacy assembly code. The routines were either rewritten in assembly to avoid the `REP`-prefix instructions, or implemented in C. Not all of these were changed in Version 9.0; work continues in Version 9.1 of SAS. This is very important in traditional Base procedures as well as in many DATA steps.
- Matrix algebra subroutines were rewritten. The most important of these had previously been coded in 32-bit assembly language. The assembly was enhanced with unrolled loops, and the performance benefits are best seen when the vectors being processed fit into CPU cache. These routines are very important in analytical applications such as SAS/STAT procedures and Enterprise Miner. An interesting point in the matrix algebra routines is that although unrolling loops may not help out-of-cache situations much, the system bus is 4x faster on a Pentium 4 processor with Intel NetBurst microarchitecture than "P6", so approximately a 4x speedup on memory-intensive analytical procs may be seen[2]. This happens even without any optimizations. Refer to Table 1 in the Appendix for an example.

Performance work in the 32-bit arena in SAS R&D continues, and SIARC helps when it can.

PERFORMANCE OPTIMIZATIONS ON INTEL® ITANIUM® PROCESSOR FAMILY

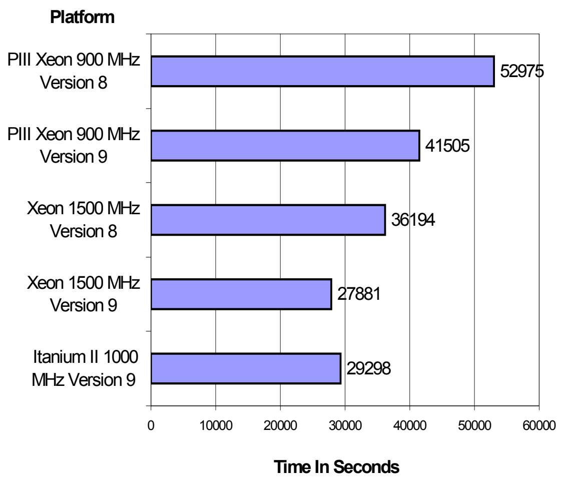
In order to take advantage of the EPIC feature of the Intel Itanium processor family, SIARC focused on certain key areas, and leveraged findings from 32-bit performance studies. These are some of the key areas. Note the similarities to 32-bit in some cases:

- Memory utility routines (`memcpy`, `memset`, `memcmp`, `memmove`) – the Itanium Processor Family is terrific at copying, setting, and comparing larger blocks of memory (above 32 bytes). However, SAS makes extensive usage of memory routines on blocks smaller than 32 bytes. SIARC and Intel are working together to improve these routines on smaller blocks. Again as in 32-bit, this is very important in traditional Base SAS procedures and many DATA steps.
- String utility routines – these are coded in C. The performance leaves something to be desired on smaller blocks of memory. Additional compiler options need to be added, along with checks for "pointer disambiguation", which allows the best parallel processing. Again as in 32-bit, this area is very important in traditional Base SAS procedures.
- Matrix algebra subroutines – these are coded in C (whereas on 32-bit the most used-routines are in assembly). For large vectors, the Intel Math Kernel Library routines are called. Similarly to the string utility routines, "pointer disambiguation" is very important. The performance here does not happen for free and requires some up-front and careful work.
- Compiler performance options – The Intel compiler for IPF is code-named "Electron". It is important to remember how critical the compiler is for IPF performance. A large part of SIARC's work on IPF so far has consisted of qualifying the compiler for SAS with the "/O2" compiler option, which is a minimal requirement for performance. SAS's requirement for robustness and solid exception handling precludes general use of an "/O3" option. Because the IPF assembly has much in common with traditional RISC, we have been able to leverage the experience of SAS developers on the RISC/Unix platforms. SIARC compiler option research continues, especially on Profile Guided Optimization (PGO), and InterProcedural Optimizations (IPO).
- Itanium 2 processor tuning – Version 9.0 of the SAS System for Windows on IPF is "tuned" for the Itanium processor. The Itanium 2 processor has different characteristics than the Itanium processor such as cache size, memory bandwidth, and memory latencies. The Intel compiler used in Release 9.1 of the SAS System for Windows under IPF is Itanium II-aware, and this will hopefully mean a potential 25-30% speedup versus Version 9.0 for some applications.

THE PERFORMANCE IMPACT OF SIARC ON 9.0 SAS

The effects of SIARC performance work on The SAS System for Windows, Version 9.0 can be seen in Figure 2[2]. The data is an aggregation of times from an internal performance test suite. The Version 8 tests were Release 8.2 (TS2M0). The Version 9 tests were from pre-production Version 9.0. The internal performance suite includes examples of Base Procedures such as SORT, SUMMARY, and analytical Procedures such as GLM, LOGISTIC, DMREG, NEURAL, and others. All Version 9 tests were run with the NOTHREADS system option. See the Appendix for machine configuration information.

Figure 2
Internal Performance Tests
User CPU Time Aggregate Performance



SAS PERFORMANCE PROFILES

Here, we show some performance profiles that highlight interesting characteristics of the Intel processors and multi-threading in Version 9.0 SAS. Although this paper is not a sizing guide, you can begin to get an idea of how SAS performance may vary on different types of Intel systems. You will also see how SIARC tracks down performance issues using CPUBUSY, a tool developed by Patrick J. Fay of SIARC to combine Intel® VTune™ counters with Perfmon data, displayed on a timeline. CPUBUSY offers capabilities that are not present in VTune. Intel currently has plans to add new features to VTune based on CPUBUSY.

Example 1 – Small GLMs Compared Between 1 and 2 Threads on 2800 MHz Intel® Pentium® 4 Processor

Here, we start by observing that two relatively short PROC GLM invocations, each using about 2.9 Megabytes of memory, averages about 23 seconds of real-time with 2 threads, but 19 seconds of real-time with 1 thread[2]. This is on a 2800 MHz Pentium 4 processor with HT Technology[1]. See the Appendix for machine configuration information. With CPUBUSY, we plot VTune counters that you would typically examine when thinking about threading. Examining Figures 3, 4, 5, and 6, we make some observations and conjectures about the data:

Observations about the test **when 1 thread** is used:

- The Context Switches Per Second is low (Figure 3).
- The CPU stays nicely saturated, and there is very little Idle time (Figure 4).
- Overall looks very good as far as threading and CPU utilization is concerned.

Observations about the test when **2 threads** are used:

- The Context Switches Per Second soars tremendously (Figure 5), which is very inefficient for threading.
- The two logical processors do not stay simultaneously busy (Figure 6).
- There is much more idle time, which implies more time is being spent in the operating system.

Conjecture (Educated Guesswork):

- The threaded code does not have enough work with this data to ensure each thread consumes its entire time slice.
- Too many processor yields are occurring; these are inefficient.
- Probably the -nothreads option is best for this test because the matrices are too small.

Also:

- If a long-running multi-threaded CPU-intensive program is run concurrently, this test will not finish until the long-running program finishes!
- Influence of HT Technology on this is unknown.
- The performance of this test case is currently being researched at SAS.

Figure 3 - \\NORTHWOOD1X1\System\Context Switches/sec - 1 Threads

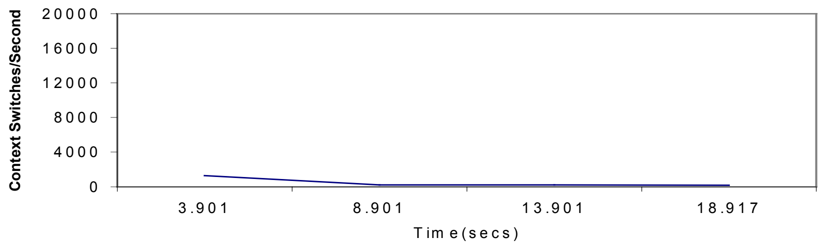


Figure 4 - Busy Over Time - 1 Threads

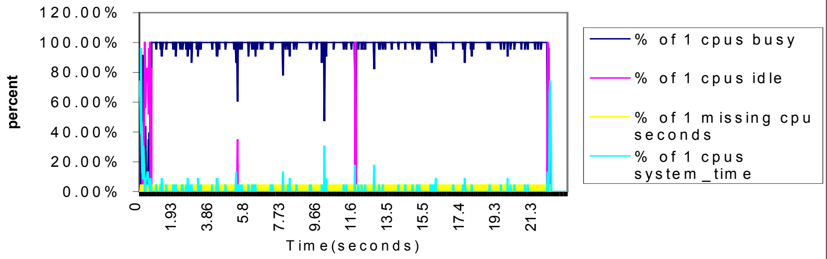


Figure 5 - \\NORTHWOOD1X1\System\Context Switches/sec - 2 Threads

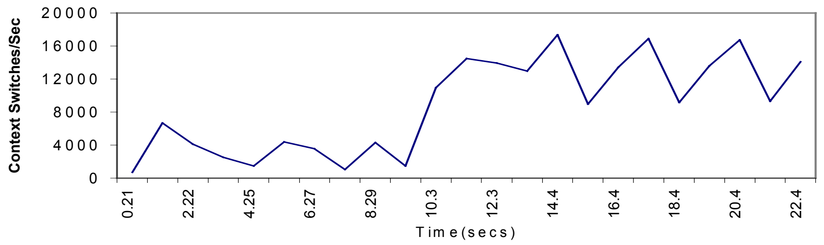
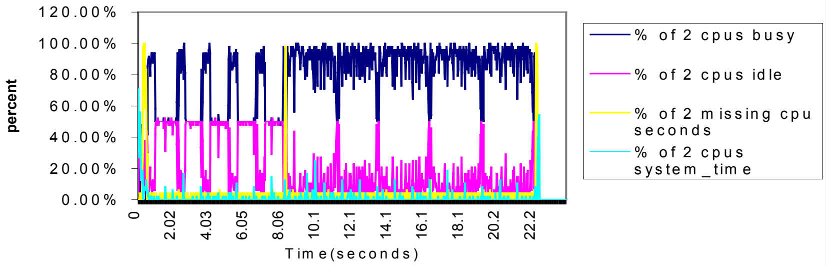


Figure 6 - Busy Over Time - 2 Threads



Example 2 - Large GLM Compared Between 1 and 2 Threads on 2800 MHz Intel® Pentium® 4 Processor

Here, we start by observing that a very large PROC GLM, using about 500 Megabytes of memory, averages over 2450 seconds of real-time with 1 thread, and a little less than 2000 seconds of real-time with 2 threads[2]. This is on a 2800 MHz Pentium 4 processor with HT Technology[1]. See the Appendix for machine configuration information. With CPUBUSY, we find these profiles for bandwidth in Figure 7 and Figure 8:

Observations:

- The VTune counter is MB/Sec of Memory Bandwidth.
- The test is mostly reading/writing from memory, at 1.9GB/sec with 1 thread and 2.2GB/sec with 2 threads. The system is theoretically capable of 3.2 GB/sec.
- The first memory bandwidth dip represents a large file written to disk, which we can deduce by examining CPUBUSY I/O graphs not displayed here.
- The periodic bandwidth dips are from VTune storing data to disk.
- 2.2 GB/sec / 1.9 GB sec = 1.15, which correlates nicely with 2000 sec / 2450 sec = 1.22.
- HT Technology works very well in this example; only 1 physical CPU!

Figure 7 - MEM32_BW_SELF_MBperSec (ave= 1929.58) - 1 Threads

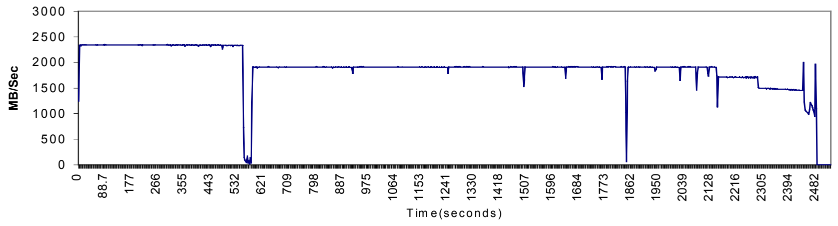
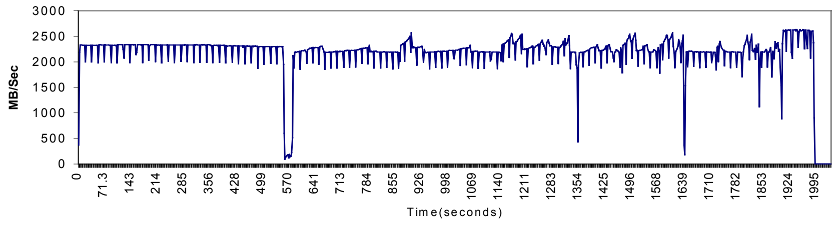


Figure 8 - MEM32_BW_SELF_MBperSec (ave= 2214.02) - 2 Threads



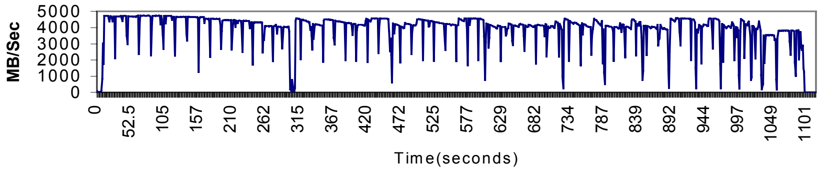
Example 3 - Large GLM at 4 Threads on 1000 MHz Itanium I2 Processor

Here, we are pleased to see that the same GLM as in Example 2, runs in slightly less than 1100 seconds on a system with 4 1000 MHz Itanium II processors, and 4 threads[2]. See the Appendix for machine configuration information. With CPUBUSY, we find:

Observations:

- The VTune counter is MB/Sec of Memory Bandwidth.
- As in Example 2, see the first dip due to file I/O.
- The periodic bandwidth dips are from VTune storing data to disk.
- The average bandwidth is 3.9 GB/sec as seen in Figure 9. The system can theoretically perform at 6.4 GB/sec of memory bandwidth.
- The memory bandwidth of the Itanium II processor is a significant performance differentiator versus Xeon and Pentium 4.

Figure 9 - MEM64_BW_SELF_MBperSec(CL) (ave= 3962.68) - 4 Threads



CONCLUSIONS

Intel's new processor designs are significantly different from their legacy systems, and these new designs have presented interesting opportunities for SAS R&D. By understanding the capabilities of these new designs, the SIARC Team has made significant optimizations that will enhance the performance of your SAS V9 applications in 32-bit or 64-bit Windows. SIARC means continuous performance enhancements and better future products for SAS customers on Intel architecture.

REFERENCES

Intel® Pentium® 4 Processor Optimization Reference Manual
Desktop Performance and Optimization for Intel® Pentium® 4 Processor
Intel® Itanium® 2 Processor Reference Manual for Software Development and Optimization

ACKNOWLEDGEMENTS

Thanks SIARC Team Members: Chuck Craddock, SAS, and Sajjid Reza, Intel, for an incredible amount of great work in 9.0.

REVIEWERS

Christopher T. Simonson, Intel
Jenny Marquez, Intel
Norman White, Intel
Mark Pallone, Intel

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Patrick J. Fay, Ph. D.
Intel Corporation Inc.
SAS Campus Drive
Cary, NC 27513
Work Phone: 919 – 531 – 0228
Fax: 919 – 677 – 4444
Email: patrick.j.fay@intel.com
Web: www.intel.com

Tracy Warren Carver
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Work Phone: 919 – 531 – 6674
Fax: 919 – 677 – 4444
Email: tracy.carver@sas.com
Web: www.sas.com

APPENDIX

TABLES

Table 1 - Large memory-bound GLM using about 500 MB of memory on Xeon™ versus Intel® Pentium® III Xeon™. This test is part of the aggregate performance data in Figure 2. Both numbers from are from the SAS System for Windows Release 8.2 (TS2M0). On both systems the test is bottlenecked by memory bandwidth. The ratio is 4.17, approximately equal to ratio of system bus frequency. See below for machine configuration information.

Processor	User CPU Time In Seconds
Intel® Pentium® III Xeon™ processor, 900 MHz, 100 MHz System Bus	15895
Intel® Xeon™ processor, 1500 MHz, 400 MHz System Bus	3807

MACHINE CONFIGURATION INFORMATION

Figure 1

- Intel® Pentium® III 500 MHz, Unified L2 cache 512K, 32-byte cache line size, 2-way desktop, Windows 2000 Professional.
- Intel® Pentium® 4 1500 MHz, Unified L2 cache 256K, 64/128-byte cache line size, Windows 2000 Professional.

Figure 2

- Intel® Pentium III Xeon™ processor, 900 MHz, Unified L2 cache 2M, 32-byte cache line size, 4-way server, Windows 2000 Advanced Server. The SAS System for Windows, Version 9.0 pre-production. The SAS System for Windows Release 8.2 (TS2M0).
- Intel® Xeon™ processor, 1500 MHz, Unified L2 cache 256K, L3 cache 1 MB, 64/128-byte cache line size, 4-way server (4 physical == 8 logical processors), HT Technology active, Windows .Net Server Enterprise Edition (Beta). The SAS System for Windows, Version 9.0 pre-production. The SAS System for Windows Release 8.2 (TS2M0).
- Intel® Itanium® 2 processor, 1000 MHz, Unified L2 cache 256K, L3 cache 3 MB, 128-byte cache line size, 4-way server, Windows .Net Server Enterprise Edition (Beta). The SAS System for Windows, Version 9.0.

Example 1, 2
Figures 3, 4, 5, 6, 7, 8

- Intel® Pentium® 4 processor, 2800 MHz, Unified L2 cache 256K, L3 cache 1 MB, 64/128-byte cache line size, 1-way desktop, (1 physical == 2 logical processors), HT Technology active, Windows XP Professional.
- The SAS System for Windows, Version 9.0.

Example 3
Figure 9

- Intel® Itanium® 2 processor, 1000 MHz, Unified L2 cache 256K, L3 cache 3 MB, 128-byte cache line size, 4-way server, Windows .Net Server Enterprise Edition (Beta).
- The SAS System for Windows, Version 9.0.

ENDNOTES

[1] Hyper-Threading Technology requires a computer system with an Intel® Pentium® 4 processor at 3.06 GHz or higher, a chipset and BIOS that utilize this technology, and an operating system that includes optimizations for this technology. Performance will vary depending on the specific hardware and software you use. See www.intel.com/info/hyperthreading for information.

[2] Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference www.intel.com/procs/perf/limits.htm or call (U.S.) 1-800-628-8686 or 1-916-356-3104.

[*] Other brand and product names are trademarks of their respective companies.