Available on CMS information server **CMS IN 2005/012**

CMS Internal Note

The content of this note is intended for CMS internal use and distribution only

September 22, 2005

Benchmarking AMD Opteron (AMD64) and Intel (EM64T) systems

H. Wenzel

Fermi National Accelerator Laboratory, Batavia, USA

Abstract

This report first gives a short introduction into the AMD64 and EM64T architectures. Then results from measuring the performance and the behavior of such two way systems in various Particle Physics applications as compared to the classical 32-Bit systems are presented. The investigations cover analysis tools like ROOT, the MonteCarlo generator Pythia, the GEANT 4 based CMS detector simulation program (OSCAR) and the ORCA based event reconstruction programs. It is shown that for the considered applications the available 64-bit commodity computers from AMD and Intel are a viable alternative to comparable 32-Bit systems. Currently CMS application can be run with a 32 Bit OS or with a 64 Bit OS in compatibility mode. A significant gain in performance might be achieved porting the CMS software to the new processor platform in 64 Bit mode.

1 Introduction

A nice introduction to the AMD64, EM64T as well as various benchmark results can be found in [1]. There are now three 64-bit implementations in the Intel compatible processor marketplace:

- Intel IA64, as implemented on the Itanium 2 processor. Since it doesn't provide a compatibility mode with the current IA32 platform we will not be cover this architecture in this note.
- Intel EM64T, as implemented on the Xeon DP Nocona and future Xeon MP processors
- AMD AMD64, as implemented on the Opteron processor.

In this note we present benchmark results obtained with EM64T and AMD64. As operating system we used Scientific Linux 3.03 which is available in 32 Bit and 64 Bit.

There are three distinct operation modes of interest available in AMD64 and EM64T:

- **32-bit legacy mode (32Bit/32Bit)**

  In this mode, both AMD64 and EM64T processors will act just like any other IA32 compatible processor. You can install your 32-bit OS on such a system and run 32-bit applications, however, you will not be able to make use of the new features such as the flat memory addressing above 4 GB or the additional General Purpose Registers (GPRs). 32-bit applications will run just as fast as they would on any current 32-bit processor.

  Most of the time IA32 applications will run even faster since there are numerous other improvements that boost performance regardless of the maximum address size. For applications that share large amounts of data, there might be performance impacts related to the NUMA-like architecture of multi-processor Opteron configurations, since remote memory access might slow your application down.

- **Compatibility mode (64Bit/32Bit)**

  The second mode supported by the AMD64 and EM64T is compatibility mode, which is an intermediate mode of the full 64-bit mode described below. In order to run in compatibility mode, you will need to install a 64-bit operating system and 64-bit drivers. If a 64-bit OS and drivers are installed, both Opteron and Xeon processors will be enabled to support a 64-bit operating system with both 32-bit applications or 64-bit applications.

  Compatibility mode gives you the ability to run a 64-bit operating system while still being able to run unmodified 32-bit applications. Each 32-bit application will still be limited to a maximum of 4 GB of physical memory. However, the 4 GB limit is now imposed on a per-process level, not at a system-wide level. This means that every 32-bit process on this system gets its very own 4 GB of physical memory space (assuming sufficient physical memory is installed). This is already a huge improvement compared to IA32, where the operating system kernel and the application have to share 4 GB of physical memory.

- **Full 64-bit mode (64Bit/64Bit)**

  The final mode is the full 64-bit mode. AMD refers to this as long mode and Intel refer to it as IA-32e mode. This mode is when a 64-bit operating system and 64-bit application are used. In the full 64-bit operating mode, an application can have a virtual address space of up to 40-bits (that equates to 1 TB of addressable memory).

  Applications that run in full 64-bit mode will get access to the full physical memory range, and will also get access to the new GPRs as well as to the expanded GPRs. However it is important to understand that this mode of operation requires not only a 64-bit operating system (and of course 64-bit drivers), but also requires a 64-bit application that has been recompiled to take full advantage of the various enhancements of the 64-bit addressing architecture.

All available AMD64 operating modes are summarized in Figure 1. Figure 2 shows the additional registers available on the AMD64 processor architecture.

Operating Mode		OS Required	Application Recompile Required	Defaults: Address Size (bits)	Defaults: Operand Size (bits)	Register Extensions	Typical GPR Width
Long Mode	64-bit Mode	New 64-bit OS	yes	64	32	yes	64
	Compatibility Mode		no	32		no	32
				16	16		16
Legacy Mode	Protected Mode	Legacy 32-bit OS	no	32	32	no	32
				16	16		
	Virtual-8086 Mode			16	16		16
	Real Mode	Legacy 16-bit OS					

Figure 1: *AMD64 operating modes (from [6])*

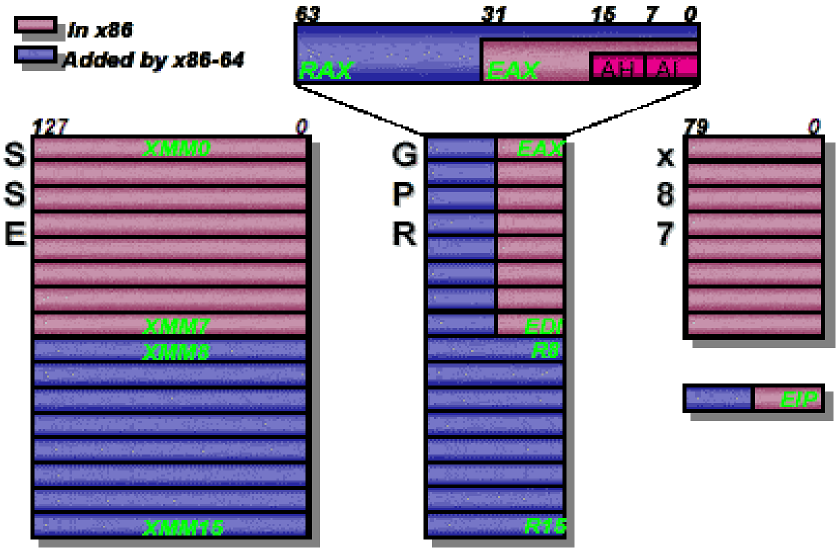

Figure 2: *Additional registers added in AMD64 [6])*

2 Benchmark Applications

We wanted to run benchmarks related to CMS physics. Porting and compiling all of the CMS OO (namely OSCAR and ORCA) ([3]) software for the new platform to run in **(64Bit/64Bit)** mode requires a lot of work and the recompilation of all the underlying libraries (e.g. POOL, SEAL....). Therefore we benchmarked this applications only in **(32Bit/32Bit)** and **(64Bit/32Bit)** mode. We were specifically interested in the compatibility **(64Bit/32Bit)** option which allows to run a 64 Bit Operating system while still being able run all CMS software. The CMS OO applications were installed using **DAR** ([**?**])which provides a 'nearly' complete runtime environment for various CMS applications but doesn't allow for recompilation.

To demonstrate the gain in speed that one gets making full use of the new architecture, we also included several ROOT ([4]) based tests and also the Monte Carlo generator Pythia ([5]) in the test suite. CMS code and ROOT are dynamically linked C++ programs while PYTHIA is a statically linked FORTRAN program. This allowed us to run this application in all three operating modes.

The Applications needed to be compiled in 64 Bit. For the recompilation we used the g++ (ROOT) and g77 (Pythia) compilers based on gcc version 3.2.3 that is provided with Scientific Linux 3.03. We are using the 4.02/00 production version of ROOT and configured it for 64 Bit using the command:

configure linuxx8664gcc

- **ROOT:**

  ROOT provides various benchmarking routines testing various aspects with its distribution:

  - **bench:** this test program compares the I/O performance obtained with all STL collections of objects or pointers to objects and also Root collection class TClonesArray.
  - **stressLinear:** The suite of programs tests many elements of the vector, matrix and matrix decomposition classes (matrix size 100x100).
  - **stressgeom:** tests the ROOT geometry classes (TGeo).
  - **stress:** The suite of programs tests the essential parts of Root. In particular, there is an extensive test of the I/O system and Trees.

- **Pythia:** For the benchmark we run 100.000 super symmetry events at $\sqrt{s}$ =14 TeV (the Pythia main65.f example). The code is compiled with g77 with the O2 option:
  g77 -O2 -o main65 main65.f pythia6227.o

- **OSCAR_3_7_0:** The GEANT 4 based CMS detector simulation program. Here we simulate 300 single pion events of 50 GeV/c P_t.

- **ORCA_8_7_1:** The OSCAR output events are then digitized (writeAllDigis) and reconstructed (writeDST) with the corresponding ORCA applications:

  - **writeAllDigis**
  - **writeDST**

Table 2 shows the matrix of the different applications that we used to benchmark the performance and the different operation modes that we tested. Of course on the XEON 32 Bit machines that we used for comparison only the 32 Bit mode is available.

3 Hardware

We had different vendors available giving similar results when equipped with the same processor. Therefore the results here are listed as a function of processor type. All machines tested here are dual processor machines. The 246HE denotes the AMD Opteron 246HE processor which is a low power version of the Opteron which can be found in various Opteron based Blade servers. We were interested in how this processor performs compared to the "normal" Opteron. The 2 IA32 machines represent our current worker nodes and are listed here for comparison.

Application	**Operating mode:**		
	32-bit legacy (32Bit/32Bit)	Compatibility (64Bit/32Bit)	64-bit (64Bit/64Bit)
ROOT: stress	Yes	Yes	Yes
ROOT: stress $\times$ 2	Yes	Yes	Yes
ROOT: stressgeom	Yes	Yes	Yes
ROOT: stressLinear	Yes	Yes	Yes
ROOT: bench	Yes	Yes	Yes
Pythia	Yes	Yes	Yes
Pythia $\times$ 2	Yes	Yes	Yes
OSCAR	Yes	Yes	N/A
OSCAR $\times$ 2	Yes	Yes	N/A
ORCA Digitization	Yes	Yes	N/A
ORCA Digitization $\times$ 2	Yes	Yes	N/A
ORCA Reconstruction	Yes	Yes	N/A
ORCA Reconstruction $\times$ 2	Yes	Yes	N/A

Table 1: *Matrix of various benchmarks run in the different operating modes. The $\times 2$ indicates that both processors were running the application simultaneously.*

Processor	Speed	Architecture
Opteron 244	1.8 Ghz	AMD-64
Opteron 246	2.0 Ghz	AMD-64
Opteron 246HE	2.0 Ghz	AMD-64
Opteron 248	2.2 Ghz	AMD-64
Opteron 250	2.4 Ghz	AMD-64
Xeon	3.4 Ghz	EM64T
Xeon	3.6 Ghz	EM64T
Xeon	2.4 Ghz	IA32
Xeon	2.8 Ghz	IA32

Table 2: *Different processors tested.*

4 Benchmarking results

Pythia Benchmarking results						
	Operating mode:					
	32-bit legacy (32Bit/32Bit)		Compatibility (64Bit/32Bit)		64-bit (64Bit/64Bit)	
CPU	single (Events/sec)	× 2 (average) (Events/sec)	single (Events/sec)	× 2 (average) (Events/sec)	single (Events/sec)	× 2 (average) (Events/sec)
244	91.0	91.0	90.3	90.4	110.1	110.0
246	101.3	101.4	100.9	101.1	121.2	121.5
246HE	101.1	101.2	100.5	100.6	122.4	122.5
248	113.0	112.9	112.3	112.3	136.4	136.1
250	121.3	121.3	120.8	120.7	146.9	146.8
XEON 3.4	88.5	88.7	88.1	88.3	108.6	108.6
XEON 3.6	93.5	94.21	93.4	93.4	115.6	115.4
XEON 2.4	65.5	65.5	-	-	-	-
XEON 2.8	75.9	75.9	-	-	-	-

Table 3: *Summary of Pythia Benchmarking results.*

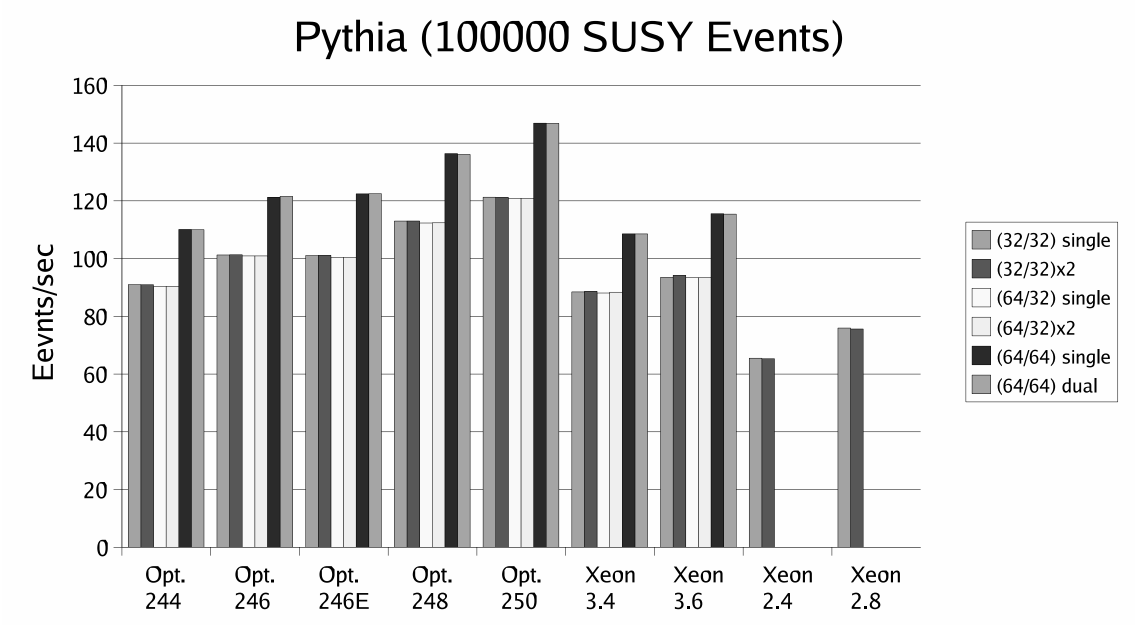

Figure 3: *Summary of Pythia Benchmarking results.*

ROOT Stress						
	Operating mode:					
	32-bit legacy (32Bit/32Bit)		Compatibility (64Bit/32Bit)		64-bit (64Bit/64Bit)	
CPU	single ROOTMarks	× 2 (average) ROOTMarks	single ROOTMarks	× 2 (average) ROOTMarks	single ROOTMarks	× 2 (average) ROOTMarks
244	723.6	715.35	672.5	673.3	1059.9	1050.75
246	797.1	793.45	751.6	748.25	1176.1	1165.65
246HE	801	805.05	752.2	749.1	1183.9	1170.35
248	894.2	888.4	834.7	835.75	1320.8	1305.95
250	957.6	956.7	898.3	891.35	1402.6	1395.55
XEON 3.4	964.6	950.05	909.3	903.8	1145.9	1135.8
XEON 3.6	1034.8	1020.6	969.7	958.7	1208.6	1194.4
XEON 2.4	590.7	583.75	-	-	-	-
XEON 2.8	731	720.05	-	-	-	-

Table 4: *Summary of Root Stress Benchmarking results. We observe that the application runs about 6% slower in compatibility mode compared to (32/32). Also in this case the Opteron runs more than 40 % faster in 64Bit mode compared to (32/32) while the 64 Bit xeons only gain about 20% .*

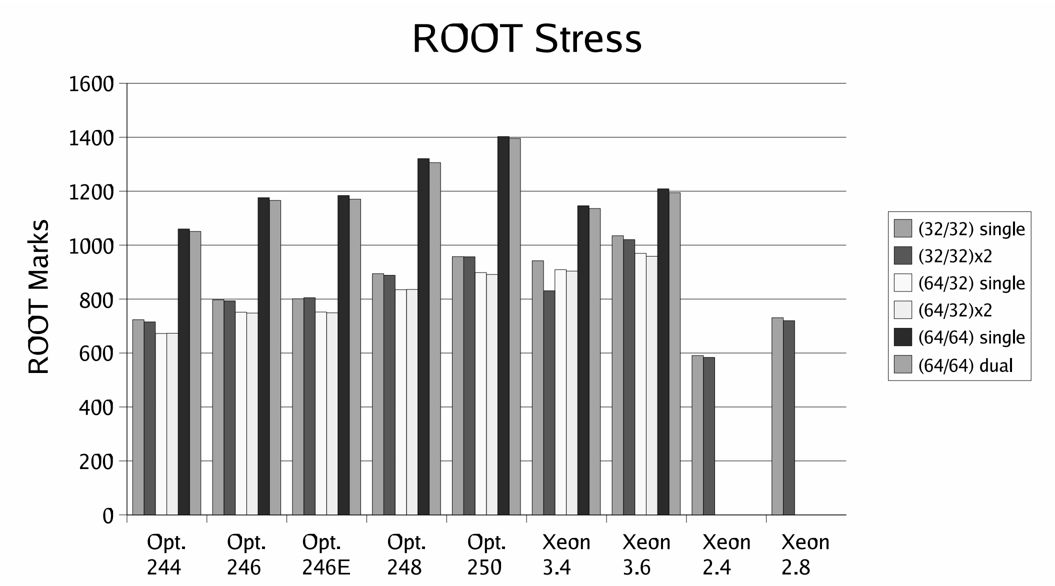

Figure 4: *Summary of Root Stress Benchmarking results. We observe that the application runs about 6% slower in compatibility mode compared to (32/32) mode. Also in this case the Opteron runs more than 40 % faster in 64Bit mode compared to (32/32) while the 64 Bit xeons only gain about 20% . Note the Opteron 250 in (64/64) Bit mode runs about 2.4 times faster than our current worker nodes (2.4 GHz xeons in (32/32)).*

ROOT Benchmarks									
	Operating mode:								
	32-bit legacy (32Bit/32Bit)			Compatibility (64Bit/32Bit)			64-bit (64Bit/64Bit)		
CPU	geom	linear	bench	geom	linear	bench	geom	linear	bench
244	673	626.1	798.59	657.4	589.9	617.15	1210.9	1017.2	1035.76
246	739.1	694.7	877.73	731	653.7	684.86	1323.6	1121.4	1160.47
246HE	737.8	697.4	883.65	735.1	653.5	681.15	1350.5	1129.6	1147.12
248	826.6	766.9	978.33	814.7	726	758.83	1497.7	1250.7	1277.26
250	893.3	833	1049.86	875.6	783.2	815.84	1632.8	1353.1	1374.24
XEON 3.4	784.3	858.6	1080.14	746.1	807.3	917.49	1293.5	1202.4	1190.89
XEON 3.6	796.8	905.7	1151.87	798.4	863.4	975.32	1369.1	1274.4	1265.8
XEON 2.4	575.7	599.3	603.79						
XEON 2.8	692.9	746.1	841.28						

Table 5: *Summary of various Root Benchmarking results.*

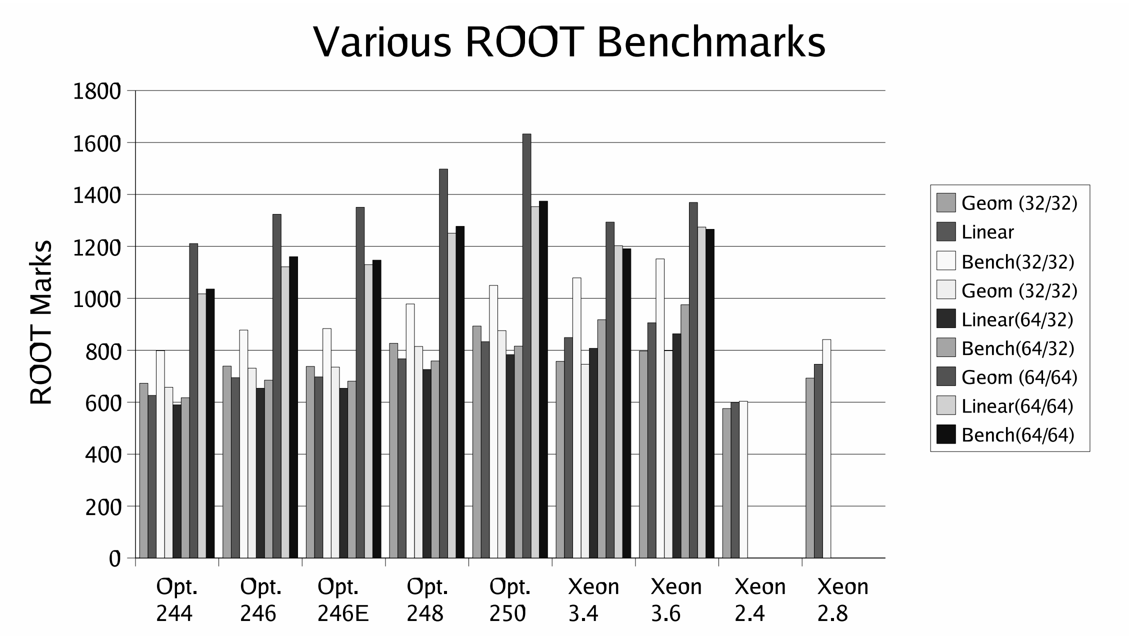

Figure 5: *Summary of various ROOT Benchmarking results.*

CMS OO Benchmarks										
	Operating mode:									
	32-bit legacy (32Bit/32Bit)			Compatibility (64Bit/32Bit)						
CPU	OSCAR single (Evt./sec)	OSCAR ×2 (av.) (Evt./sec)	DIGIS single (Evt./sec)	OSCAR single (Evt./sec)	OSCAR ×2 (av.) (Evt./sec)	DIGIS single (Evt./sec)	DIGIS ×2 (av.) (Evt./sec)	DST single (Evt./sec)	DST ×2 (av.) (Evt./sec)
244	0.0915	0.0923	0.1171	0.0866	0.0867	0.0989	0.0980	0.809	0.806
246	0.1021	0.1028	0.1293	0.0956	0.0959	0.1092	0.108	0.901	0.895
246HE	0.1037	0.1026	0.1325	0.0967	0.0966	0.1103	0.1104	0.912	0.898
248	0.1147	0.1139	0.1430	0.1067	0.1074	0.1210	0.1196	0.974	0.988
250	0.1186	0.1221	0.1562	0.1154	0.1156	0.1300	0.1296	1.039	1.035
XEON 3.4	0.1067	0.1060	0.1403	0.0938	0.0929	0.1251	0.1238	0.929	0.890
XEON 3.6	0.1066	0.1057	0.1427	0.1012	0.1012	0.1332	0.1310	1.026	1.019
XEON 2.4	0.0603	0.0604	0.0868						
XEON 2.8	0.0715	0.0702	0.0996						

Table 6: *Summary of CMS Benchmarking results.*

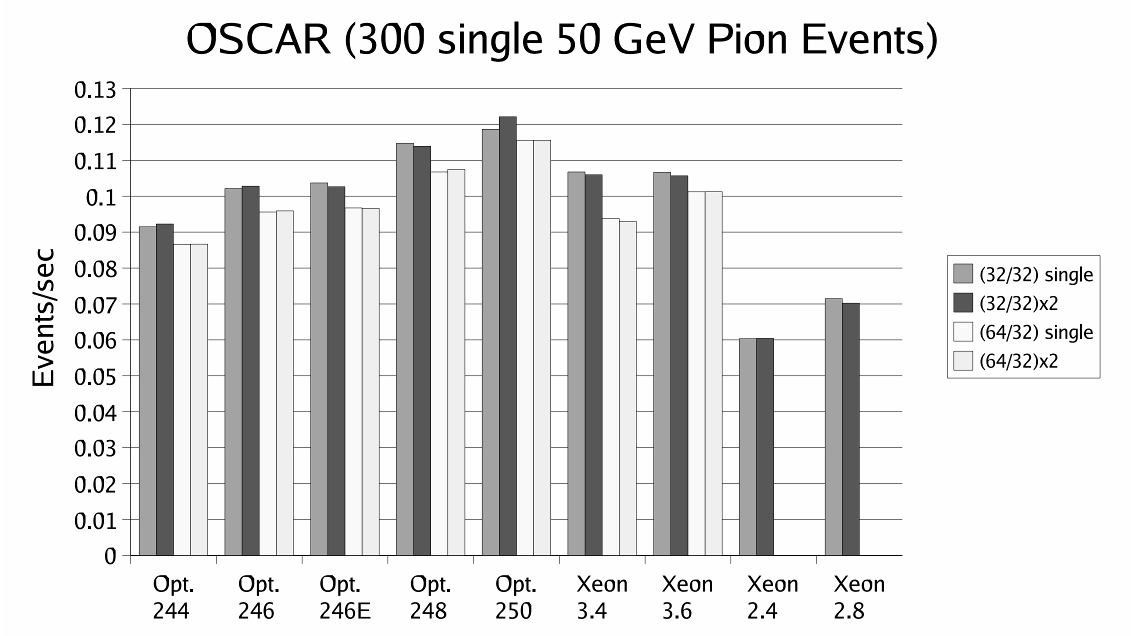

Figure 6: *Summary of OSCAR Benchmarking results. Again as for the Root stress results we observe that the application runs about 5 % slower in compatibility mode.*

5 Administration

In principle the new platform behaves like any PC. Linux distributions exist and installation and booting the system just works as we are used to. We also successfully used the Rocks Cluster Distribution to install the OS.

With the new platform System management becomes a little bit more complex. When running with a 64 Bit OS in addition to the 64 Bit libraries also the 32 Bit Environment must be provided and maintained to allow applications to run in compatibility mode. Some of the tools (e.g. APT) are not multi platform aware and for other tools (YUM, RPM) the support for the new environment could be improved.

6 Conclusion

CMS software runs in 64 Bit compatibility mode and 32 Bit native mode without any modifications to the code. We observe OSCAR and ROOT applications to run around 5% slower in compatibility mode than in legacy mode. The effect is even more pronounced for the digitization step where the loss in performance is around 15 %. When running in 64 Bit native mode we observe speed increases ranging from 21 % (Pythia) to 49 % (SGeom) compared to 32 Bit native mode. So it will be interesting to see how CMS code would perform after being ported to the new platform. Currently we definitely can't use all the performance provided by the new platform. The Opteron 250 that we tested in (64/64) Bit mode runs the ROOT stress test about 2.4 times faster than our current worker nodes (2.4 GHz xeons in (32/32)).

When porting the code to 64 Bit cernlib probably would have to be dropped since there are no plans to port it to 64 bit and the code assumes that the data types int, long and pointer have the same size which is not the case for AMD64/EM64T.

The low power version of the Opteron performs just as well as the "normal" one with the same speed.

Dual core CPU's are announced for 2005. They promise to nearly double the computing power while not increasing the demands for space, cooling and electricity significantly. It will be very interesting to test this CPU's once they become available. In general with the applications studied in this report we observe no drop in efficiency when running 2 processes simultaneously on the 2 CPU's. This is a good indication that multi-core chips will just scale with the number of processors for our applications.

References

[1] **Proceedings of CHEP 2004 Interlaken, Switzerland**, P. Wegner, S.Wiesand, *"64-Bit Opteron systems in High Energy and Astroparticle Physics"*.

[2] http://cmsdoc.cern.ch/cmsoo/cmsoo.html.

[3] *DAR was developed by Natalia Ratnikova and stands for:* **D***istribution* **A***fter* **R***elease*, http://home.fnal.gov/~natasha/DAR/dar.html.

[4] http://root.cern.ch.

[5] T. Sjöstrand, P. Edén, C. Friberg, L. Lönnblad, G. Miu, S. Mrenna and E. Norrbin, Computer Phys. Commun. 135 (2001) 238 (LU TP 00-30, hep-ph/0010017).

[6] *"What's So Great for Developers About the AMD64?"* David Kaplowitz June 23, 2003 http://www.devx.com/amd/Article/16018.

[7] *"64-bit Computing with Intel EM64T and AMD AMD64"* http://www.redbooks.ibm.com/abstracts/tips0475.html.