64-bit Linux on AMD64/EM64T

Stephan Wiesand
DESY -DV -

November 23rd , 2004

Outline

- introduction to AMD64 & EM64T
  - more than just an extended address space
- performance comparisons for physics applications
  - ROOT, Sieglinde, Pythia, FORM
  - on Opteron, Nocona, Prescott & 32-bit systems
- managing and using linux on these systems
  - 64bit distributions
  - 32bit compatibility
  - problems
- status & strategy for DESY computing

Terminology

- AMD started with x86-64
  - (to sound vendor neutral ?)
  - then renamed to AMD64
    - (around this time, intel claimed nobody wants or needs this)
- intel started out with IA32E
  - back then, IPF was still called IA64
  - then renamed it to EM64T
    - Extended Memory 64 Technology
- rpm architecture suffix is x86_64 or ia32e
- I'll use **AMD64 as the generic term**
  - credit where credit's due

Another 64bit Platform ?

- linux has been running on 64bit platforms for a while
  - Alpha, Sparc, PPC, PA-RISC, IPF (formerly known as IA64)
  - all are RISC, and none can execute i386 instructions
  - software emulation exists for Alpha and IPF
    - slow
- AMD64 is an extension of the i386 CISC architecture
  - executes i386 instructions in hardware
  - can run a 32bit OS
  - supports running 32bit applications under 64bit OS
  - 64bit mode needed an extended instruction set
    - allowed for additional registers and addressing modes

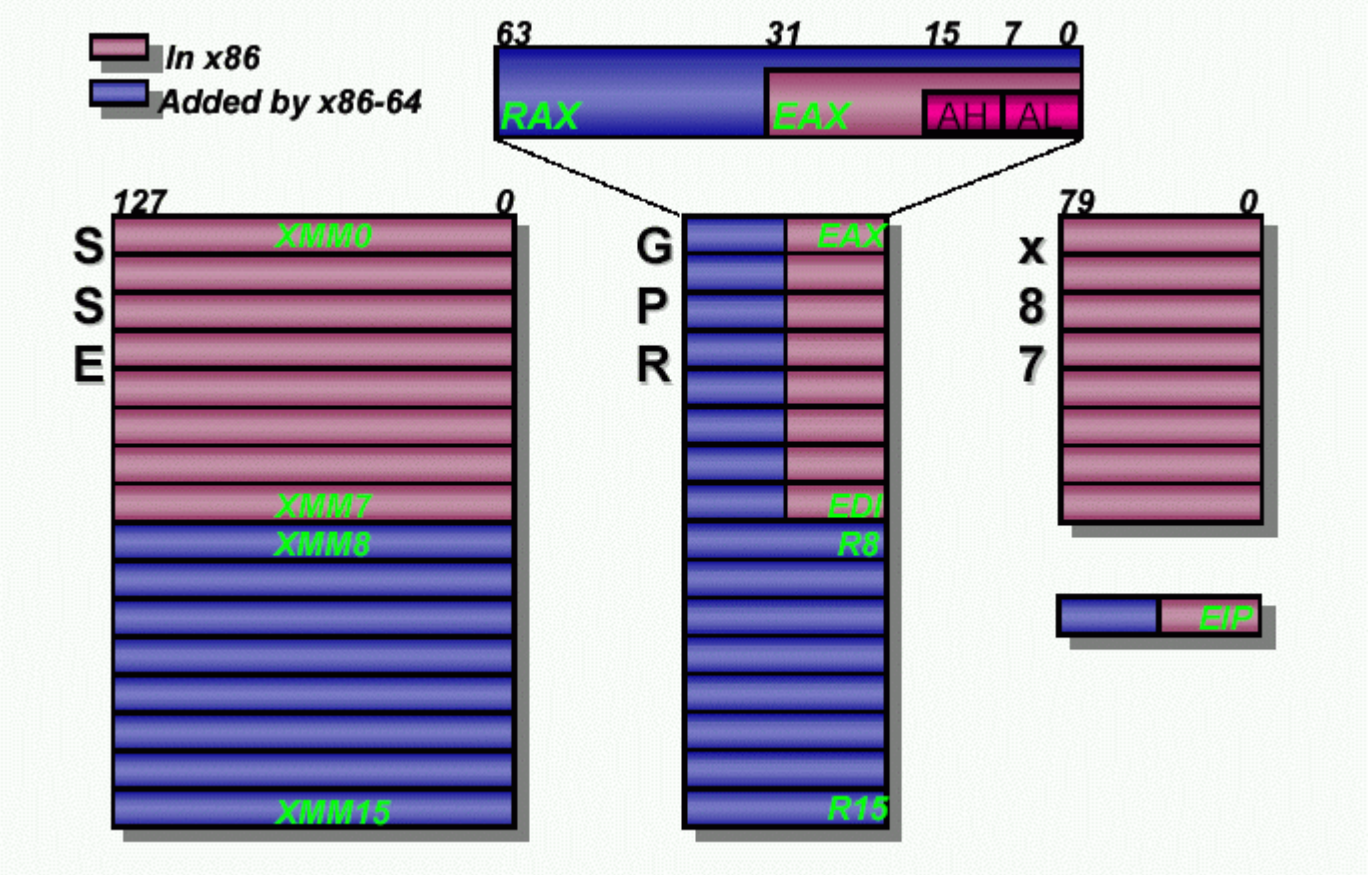
Why 64bit, anyway ?

- 64bit ≠ twice the performance of 32bit
  - often rather slower than faster
    - obvious exception: 64bit integer arithmetics
      - but often 32bit does fine
  - higher memory consumption (pointers, longs, long doubles)
  - AMD64 ist fast for other reasons
- but it breaks the 4GB limit we're approaching rapidly
  - max virtual address space for 32bit
  - actual limit is 3 GB (3.5 at best) per process
    - even if RedHat claim their kernel can do 4GB/4GB split
    - i386 allows up to 64 GB memory for OS (via PAE)
      - eats cycles, clumsy (remember "DOS extended memory" ?)
      - mappings need (low) memory themselves

64 bits ?

- not quite: AMD64 supports
  - 40 bits (1TB) of physical memory
  - 48 bits (256 TB) of virtual memory
- current chipsets may support less
  - 915X/925X: 4GB of physical memory...
- ABI imposed limits for executables in 64bit mode:
  - "small" code model: 2 GB code + data
  - "medium model": 2 GB code (w/ performance penalty)
- 32bit apps under 64bit OS have full 4GB address space
  - 3GB is the limit under 32bit kernels (3.5 at best)

AMD64 register set



- general purpose registers and instruction pointer are 64 bits wide, twice the number of GPRs
  - all addressable as 8,16,32, or 64 bits as needed
- twice the number of SSE (formerly MMX) registers
  - still 128 bits wide

AMD64 Operating Modes

Operating Mode		OS Required	Application Recompile Required	Defaults		Register Extensions	Typical GPR Width
				Address Size (bits)	Operand Size (bits)		
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					

- CPU enters Long Mode or Legacy Mode during boot, no way back
- rumour: extended register set could be accessed in 32bit mode as well ("REX32")
  - would still need modified OS and compilers

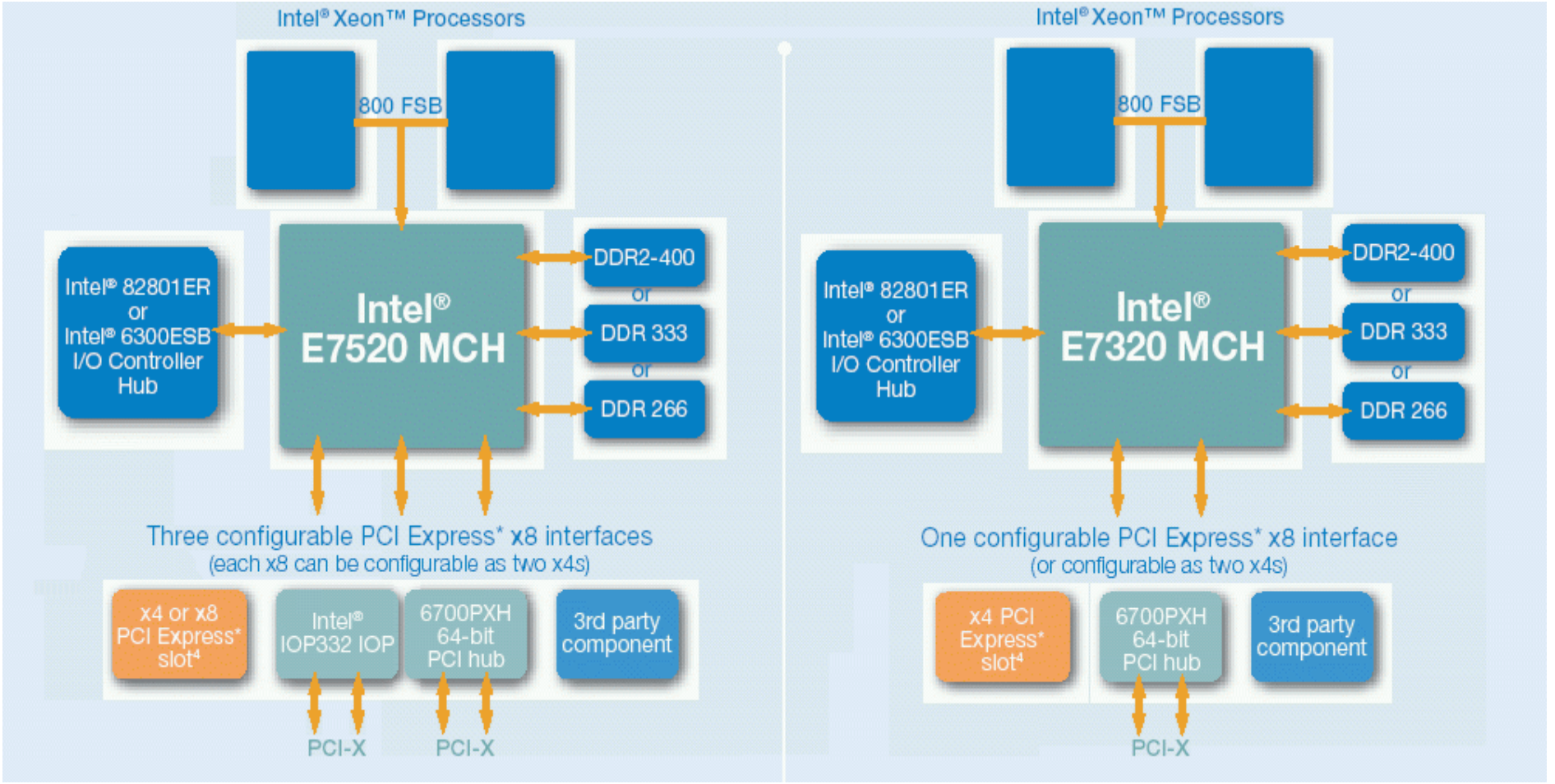
AMD64 Instruction Set Changes

- besides 64bit specifics:
- effective protection of memory against execution
  - "NX" bit
  - available in 32bit mode as well
- generally usable instruction pointer relative addressing
  - reduced performance penalty for position independent code
    - -> shared libs
    - from 20% to 8%
- 64bit apps must not use x87 instructions
  - x87 stack not preserved across context switches

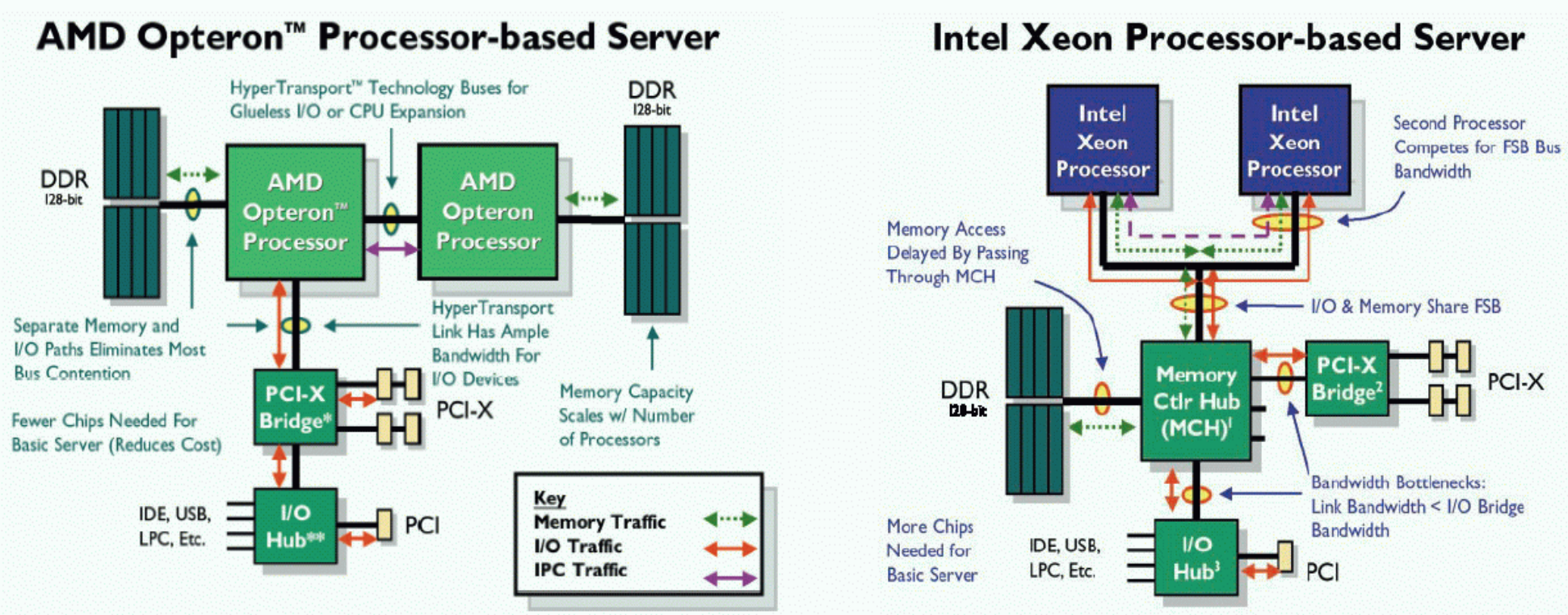
AMD64/EM64T Differences

- most visible: SSE instructions
  - both implement SSE2
  - only AMD64 implements 3dNow!
  - only EM64T implements SSE3
- a few more subtle differences in instruction sets
  - should only matter for kernel, glibc, compilers
  - should not affect ordinary application programmes
  - everything we compiled with pre-EM64T gcc releases worked on EM64T systems

Where's the Bottleneck ?

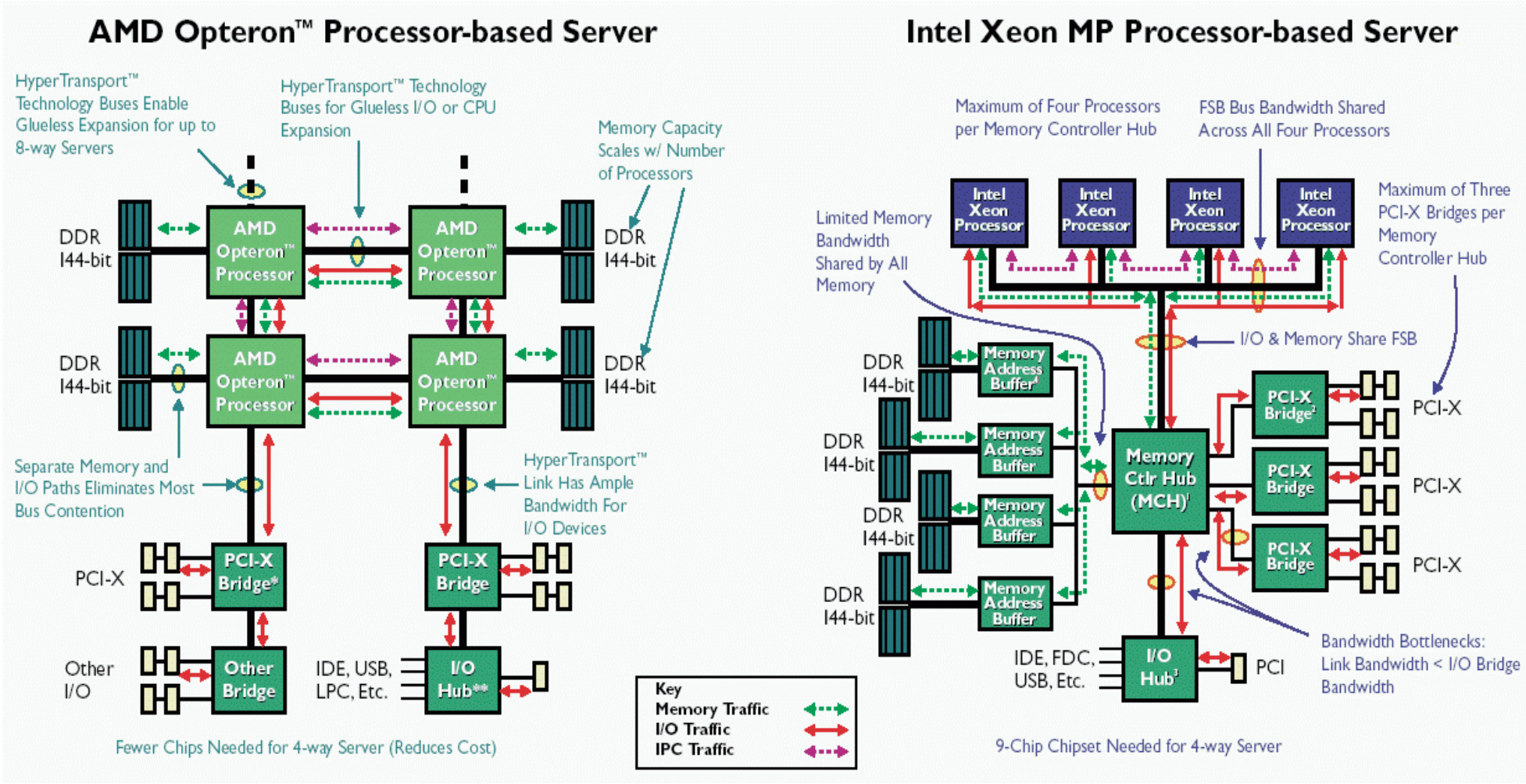


AMD's Additional Step



- currently all 6.4 GB/s:
  - memory interfaces
  - front side bus (FSB 800; FSB 1066 is in the pipeline)
  - HyperTransport links

4-way systems



- we may finally see 4-way systems that make (more) sense
  - and become affordable because there's a sizable market

NUMA: Non Uniform Memory Access

- memory now may be more or less close to CPU
  - cache coherent access to remote memory at full bandwidth
  - but bandwidth has to be shared and latencies increase
  - requires kernel with NUMA support to be most efficient
    - memory should be allocated close to requesting process/thread
    - processes/threads should be scheduled close to their memory
- alternatively, BIOS may also present all RAM to the OS as single uniform block, node memory interleaved by page
  - no OS support required
- whenever using shared memory, allocate it from the process or thread that uses it most

Opteron vs. Xeon w/ EM64T

	Opteron	Xeon
Process	130 nm (90: 2H/04)	90 nm
L1 Cache	64KB+ 64KB	12KB + 8KB
L2 Cache	1MB	1MB
Memory Controller	on chip	northbridge
FSB Speed	chip clock	800 MHz
Memory Bandwidth	6.4 GB/s/CPU	6.4 GB/s
Hyperthreading	no	yes
SSE2	yes	yes
SSE3	no	yes
1-way	Opteron 1xx	Pentium 4
2-way	Opteron 2xx	Xeon
4/8-way	Opteron 8xx	Xeon MP

- xx: 46 = 2 GHz, 48 = 2.2 GHz, 50 = 2.4 GHz, ...

Other Differences in Architectures

- DMA to memory above 4 GB
  - Opterons have an I/O MMU to make this possible
  - Intel's chips do not
    => have to use bounce buffers

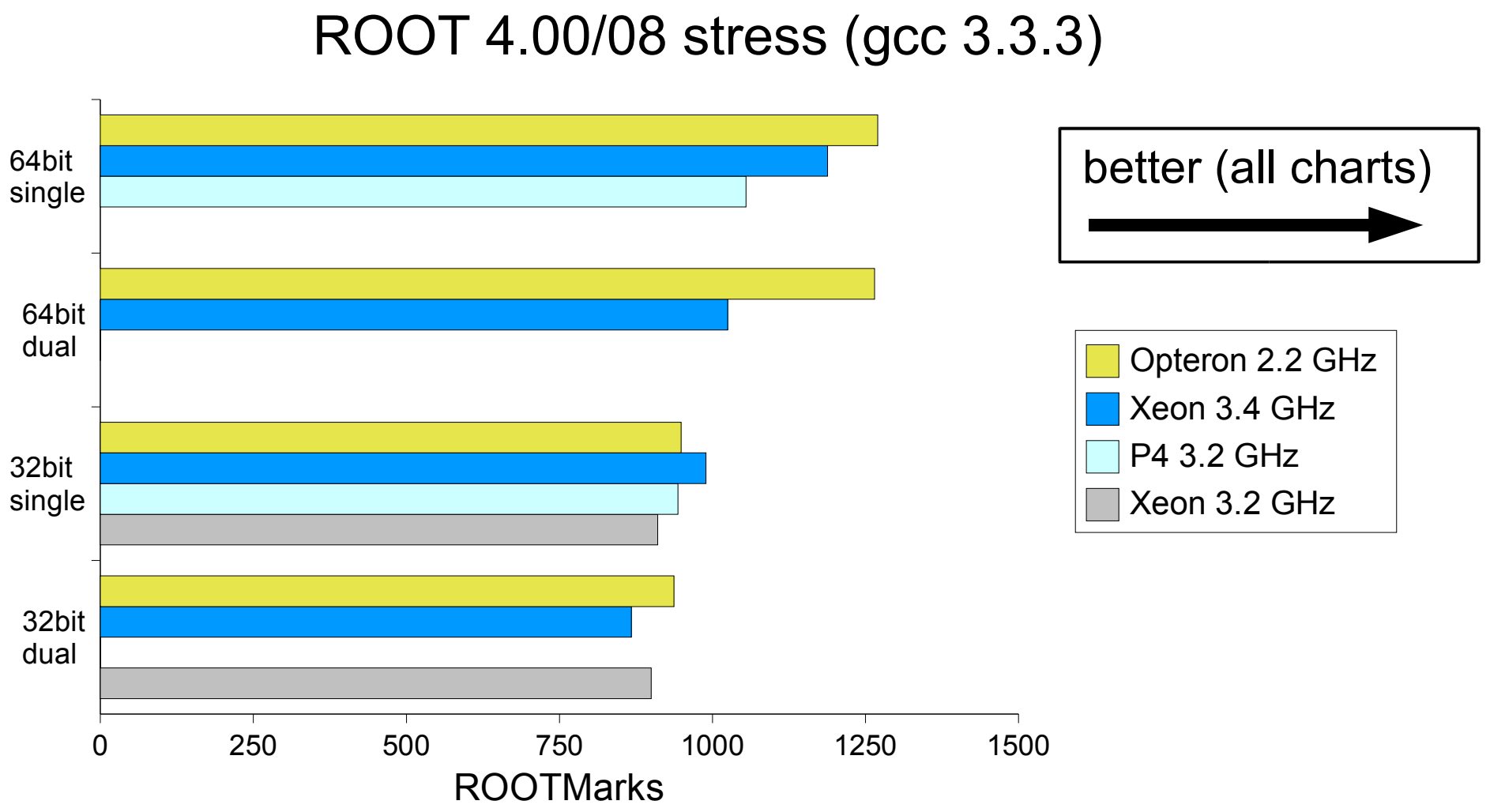
Hardware for Performance Comparisons

- All equipped with 2 CPUs and SCSI disk:
  - Opteron 2.0 GHz: IBM eServer 325, 4 GB
    - SuSE 9.0 professional, kernel 2.4.21-215-smp
  - Opteron 2.2 GHz: Sun Fire V20z, 4GB
    - SuSE 9.0 professional, kernel 2.4.21-231-smp
  - Xeon 3.4 GHz: Supermicro 7044H-X8R, 4GB
    - SuSE 9.1 professional, kernel 2.6.4-52-smp
  - Xeon 3.2 GHz: Sun Fire V65x, 2 GB
    - SuSE 8.2 professional, kernel 2.4.26
  - Tualatin 1.266 GHz: Supermicro 6013H, 1GB
    - SuSE 8.2 professional, kernel 2.4.25

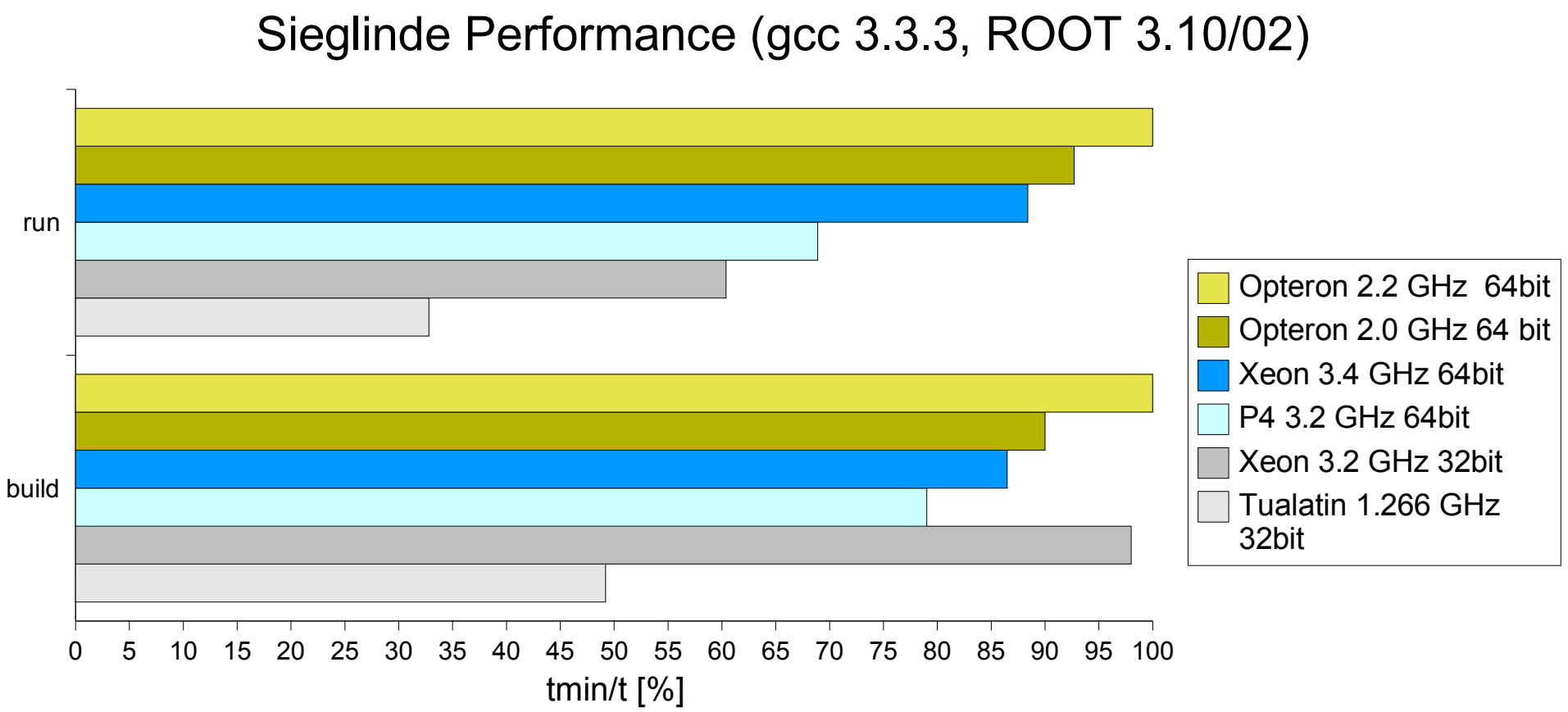
Latest Addition

- EM64T hitting the desktop:
  - single P4 3.2 GHz
    - Dell Precision 370
    - 512 MB
    - SATA disk (80 GB WD)
    - SL 3.0.3, kernel 2.4.20-21.EL
    - 925X chipset

ROOT Performance

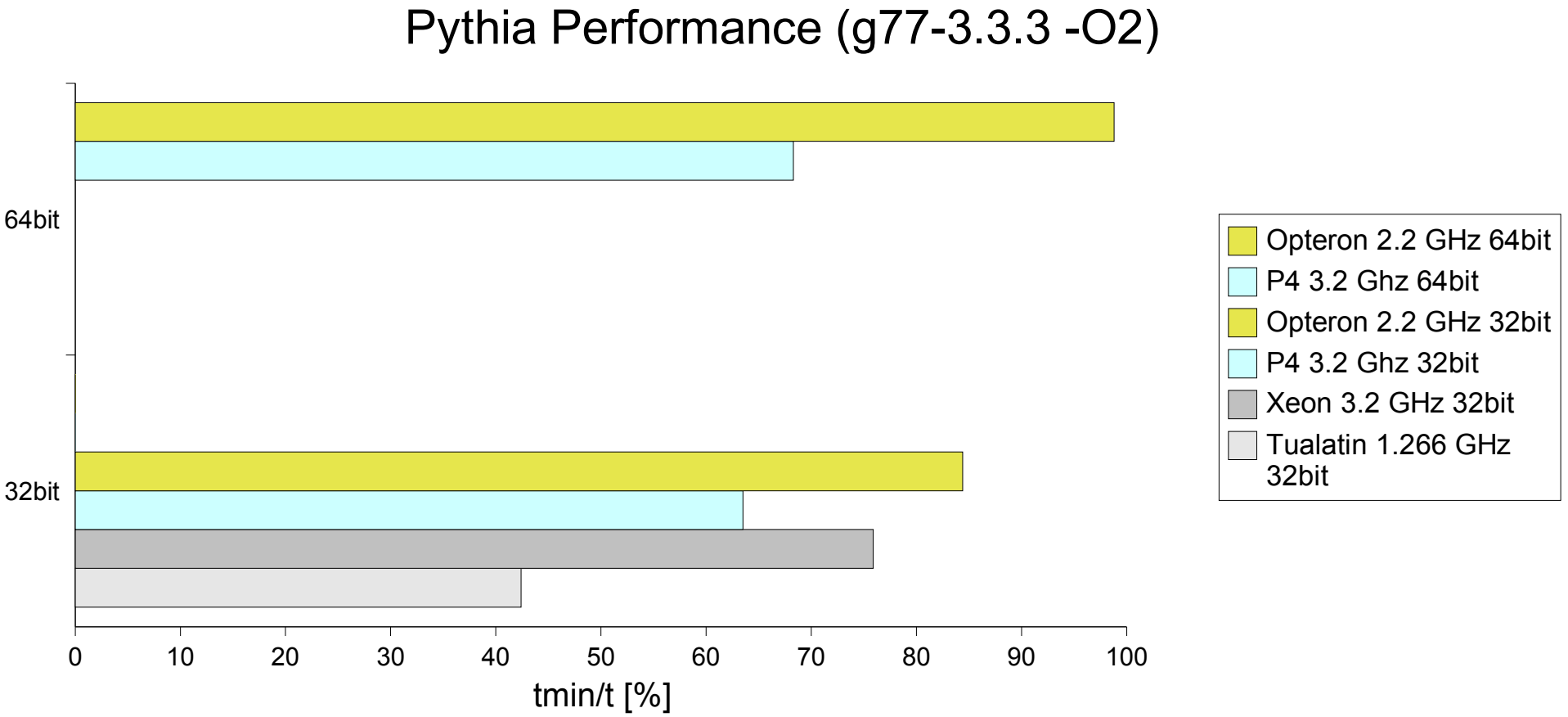


Sieglinde Benchmark



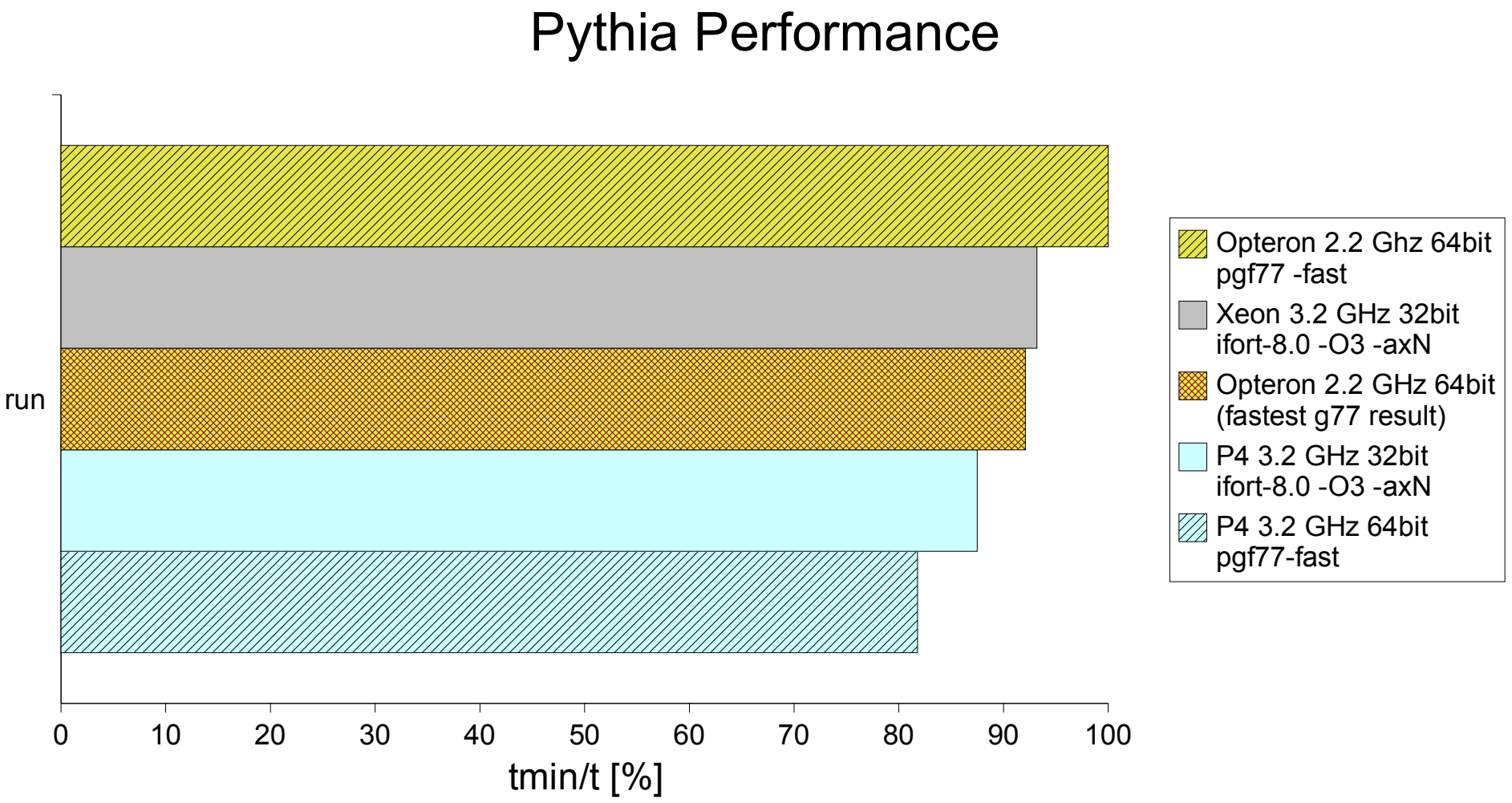
- Amanda experiment's neutrino reconstruction / filtering software
- single process, but uses a MySQL server on same host
- software made available by Peter Nießen, Univ. of Delaware

Pythia 6.2 (g77)

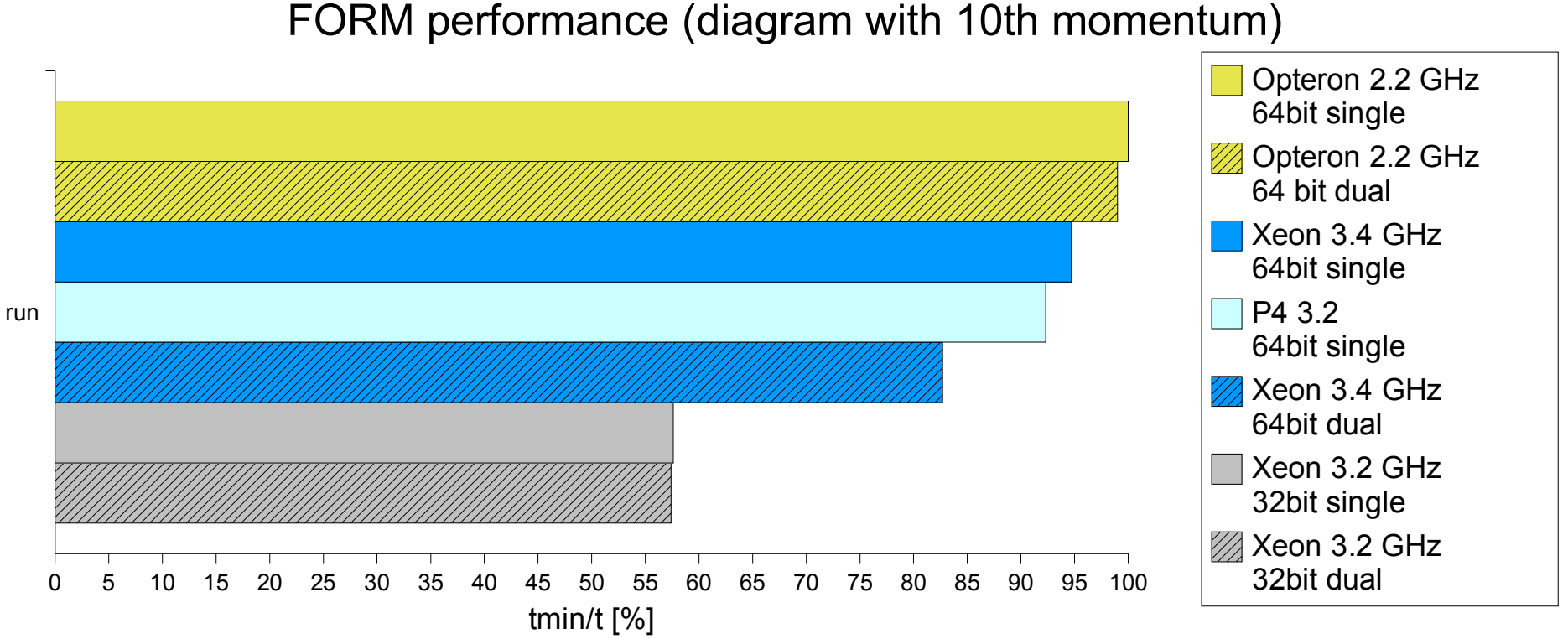


- Pythia 6.2 example 4
  "study of W mass shift by colour rearrangement at LEP 2"

Pythia 6.2 (Commercial Compilers)

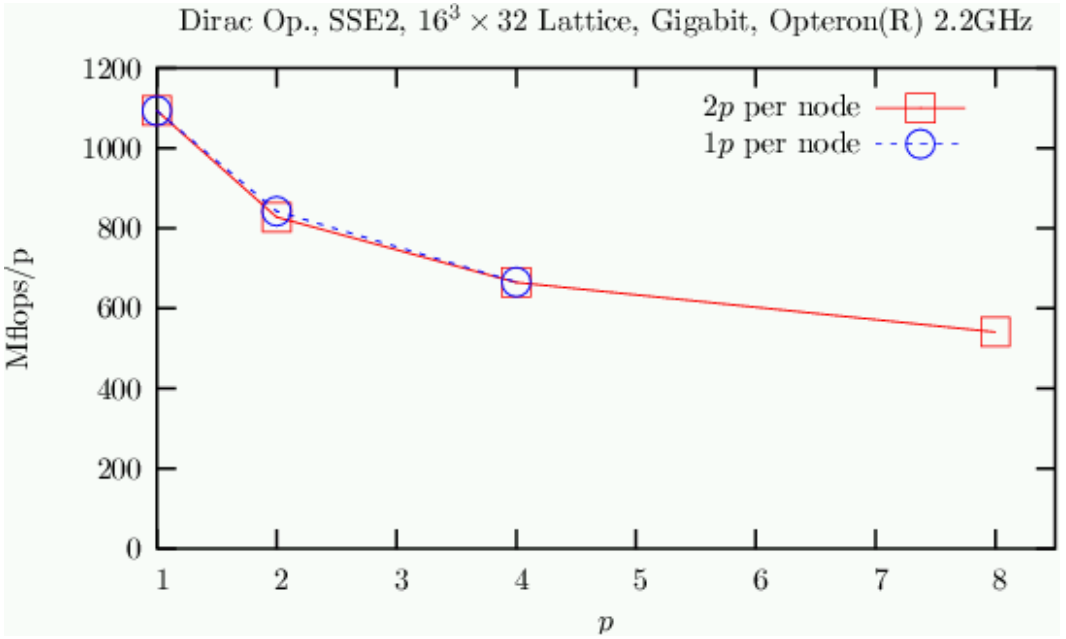


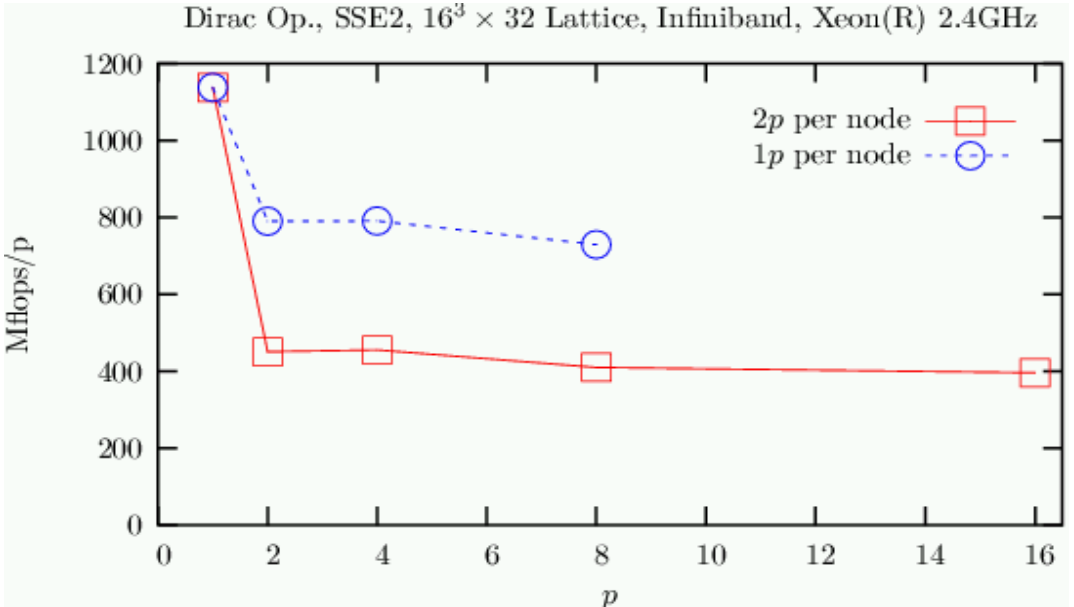
FORM 3.1



- symbolic formula manipulation, C, huge data sets
- implements own "paging" of data to disk
- 64bit executable built by author J. Vermaseren on DESY test system
- 32bit executable built with icc (www.nikhef.nl/~form)

Dual/Single CPU performance in Clusters





- measurements by C. Urbach, FU Berlin
- 32bit Lattice QCD, MPI
- performed on clusters with Gigabit Ethernet and Infiniband interconnects (FZK)
  - not our test systems
- p = number of processes

Performance Comparisons: Summary

- AMD64/EM64T systems are fast, even in 32bit mode
  - they're significantly faster in 64bit mode
  - missing: repeat 32bit runs under 32bit OS on same hardware
- Opteron systems make very efficient use of a 2nd CPU
  - and of additional MHz
- one gets more out of both with commercial compilers
- 64bit comes at a cost:
  - increased footprints in memory & on disk
    - typically 25%
  - additional platform to support

64bit Linux on AMD64/EM64T systems

- good news: system looks, feels and behaves like "a linux PC"
  - BIOS (press F2 during boot...)
  - boot loader (grub, lilo)
  - OS installation (Red Hat, SL, SuSE)
- problems:
  - porting physics applications to 64bit
  - providing 32bit compatibility environments
  - residual bugs (features?) in 64bit ports of system software

Porting Issues

- potential problems:
  - assumption that sizeof(int) = sizeof(long) = sizeof(void*)
  - inline assembly must not use x87 instructions
  - x87 registers were 80bit wide
    - intermediate results kept in registers with this precision
      - was a problem when we moved from RISC to Linux/x86
    - intermediate results of FP arithemetics in SSE registers are 64bit again (standard IEEE precision)
- can't mix 32/64-bit in same application
  - all libraries needed must be available as 64-bit
  - cernlib isn't

Data Type Sizes

type	x86	x86-64
char	8	8
short	16	16
int	32	32
long	32	64
long long	64	64
float	32	32
double	64	64
long double	96	128
void*	32	64

- no alignment constraints (like on i386)
  - but "natural" alignment is much faster

About CERNLIB

- CERN discontinued support end of 2003
  - announced well before
  - no hope for a change of policy
- it will not be ported to any new platform
  - no known sustainable effort on 64bit port
- if you're using it in your current software
  - you have a problem
  - cut off from performance gains due to platform enhancements
- if you're using it in software for future projects
  - get rid of it now, or you lock yourself into the past

Now Shipping for AMD64/EM64T

- Oracle DB
- compilers, libraries:
  - Intel
  - PGI
  - NAG
  - Pathscale
- SUNs Java SDK 1.5 (5 ?)
- MySQL DB
- Mathematica 5, Matlab
- ...

32bit Compatibility: Runtime

- 64bit linux allows running 32bit applications transparently
  - provided all shared libs are available
    - 64bit libraries go into .../lib64
    - 32bit libraries go into .../lib as before
    - mandated by Linux Standards Base
    - not all ISVs comply
      - Oracle uses $ORACLE_HOME/lib and $ORACLE_HOME/lib32
  - some applications must be persuaded by using the "linux32" prefix command (see `setarch(1)`):
    - **`uname -m`** returns x86_64
    - **`linux32 uname -m`** returns i686
    - **`linux32 math`** (only app found to need this yet)

32bit Compatibility: Development

- 64bit Linux also allows building 32bit software
- gcc >= 3.2 creates 64bit objects by default on x86-64
  - `-m32` switch makes it create 32bit objects
  - and link against 32bit libraries
  - gcc3 on 32bit accepts the -m32 switch as well (noop there)
- reality is more complex
  - a decent Makefile uses commands like **`root-config --libs`**
- 32bit development best done in pure 32bit environment
  - may be chroot (or CHOS) environment on a 64bit system

32bit Compatibility: Distributions

- Red Hat and SuSE (at least) provide 32bit packages
- the SuSE way:
  - RPM "xyz-32bit" with 32bit specific content
    - installed alongside the "xyz" 64bit package, no clashes
- the Red Hat way:
  - first install xyz.i386.rpm, then xyz.x86_64.rpm
  - limited support by RPM/YUM, not yet by APT (SPMA/rpmt ?)
    - **`rpm -ql glibc.i686`**
    - **`yum install openssl.i686; yum remove openssl.i686`**
      - careful:
        this will remove any files shared with openssl.x86_64
      - order matters

Other Problems Encountered

- Kerberos 5/AFS problem (SL incl. 3.0.3)
  - login yields K4/K5 Tickets, but no AFS token
  - aklog segfaults, afslog fails (used in pam_krb5afs.so)
  - krb5 code defines KRB4_32 to be 64 bits on any 64bit platform except alpha

```
#ifndef __alpha
#define KRB4_32 long
#else
#define KRB4_32 int
#endif
```

  - SRPM has a Patch37 fixing these issues
    - disabled after discussion on krb5 development list
  - => workaround: rebuild krb5 with Patch37 enabled
    - afslog (and the pam module) now work
    - aklog still segfaults

Strategy for DESY Computing

- 64-bit is the way to go for physics computing
  - the 4 GB limit is lurking
  - top 20% of performance potential of current hardware is accessible to 64bit applications only
  - x86 CPUs are approaching the ceiling
    - performance gains due to multicore or different platforms only
- probably no more 32-bit-only farm nodes
  - we prefer Opterons over Xeons for the time being
- SL3 will be available in 32-bit or 64-bit from the start
  - next generation desktops most likely 64-bit capable
    - 64-bit support for desktops & interactive work may lag a bit

Status in Zeuthen

- 1 single P4 system
  - SL3/64 bit development
- 14 dual Opteron systems
  - 9 x HPC Infiniband cluster, being set up (SL3/RHEL3)
  - 2 x restricted use (special theory installation) (still SuSE 9.0)
  - 1 x test for cluster, to become farm node (SL 3.0.3)
  - 1 x farm node (still SuSE 9.0/amd64)
    - access with `-l sys=amd64`
  - 1 x public interactive login (still SuSE 9.0/amd64)
    - lx64.ifh.de
- 10 more dual Opteron farm nodes being tendered

Summary

- AMD64 is becoming mainstream
  - ecosystem is in place
  - distributions are usable
  - hardware is affordable
  - fast even for legacy codes
- x86 is close to (or past) its "best before" date
- Itanium price/performance better than Xeon in 2007
  - according to intel
  - for native 64-bit programmes only
- time to get ready

Sources/Reading

- [1] Porting to AMD64 Frequently asked questions
  - www.amd.com/us-en/assets/content_type/DownloadableAssets/dwamd_AMD64_Porting_FAQ.pdf
- [2] The AMD64 ISA value proposition
  - www.amd.com/us-en/assets/content_type/DownloadableAssets/dwamd_Value_of_AMD64_White_Paper.pdf
- [3] Intel E7520/E7320 Product Brief
  - ftp://download.intel.com/design/chipsets/E7520_E7320/303033.pdf
- [4] Opteron 2P Server Comparison Reference
  - www.amd.com/us-en/assets/content_type/DownloadableAssets/30291C_brief_p1.pdf
- [5] Opteron 4P Server Comparison Reference
  - www.amd.com/us-en/assets/content_type/DownloadableAssets/30291C_brief_p1.pdf
- [6] Jan Hubička: Porting GCC to the AMD64 architecture
  - www.ucw.cz/~hubicka/papers/amd64.pdf