Massive Tracking on Heterogeneous Platforms

Eric McIntosh
(Attached to AB/ABP CERN)
eric.mcintosh@cern.ch

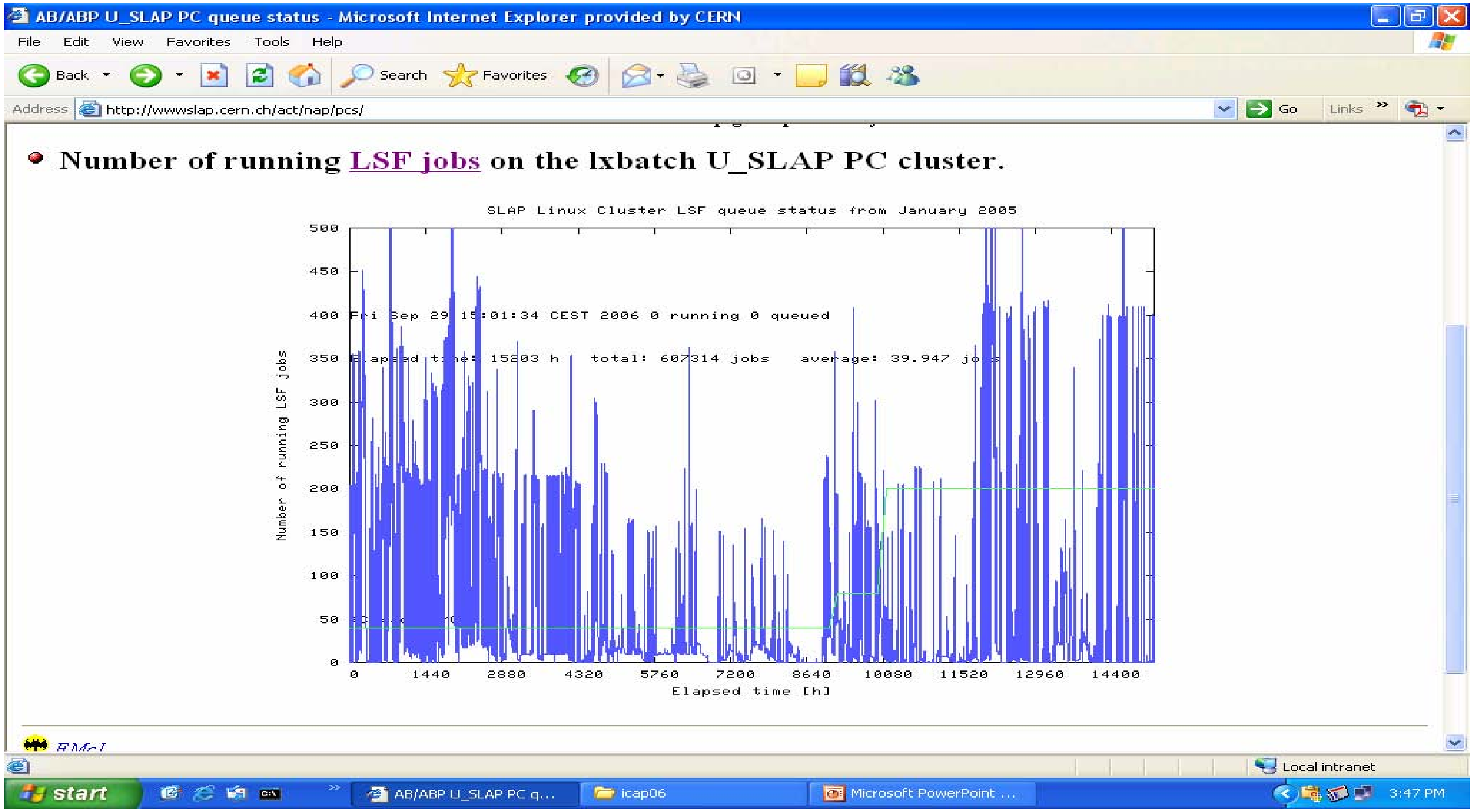
Computing at CERN

- Dominated by the needs of the experiments
- Accelerator design, a small fraction of the various mainframes (1964 – 1998) and the “PARC” IBM workstation cluster
- In 1997 the LHC Machine Advisory Committee recommended more tracking
- The “Numerical Accelerator Project”, NAP

NAP Evolution

- A 10 processor Digital/Compaq Alpha TurboLaser (800 CERN Units)
- Added 10 Workstations (1,300 CUs)
- Overlapped by 20 DUAL 800Mz PIII's (7,200 CUs)
- Today 64 Dual 2.4GHz PCs (51,200 CUs)
- Operated as "Fair Share" of the central Linux LSF Batch system lxbatch

Tracking Studies



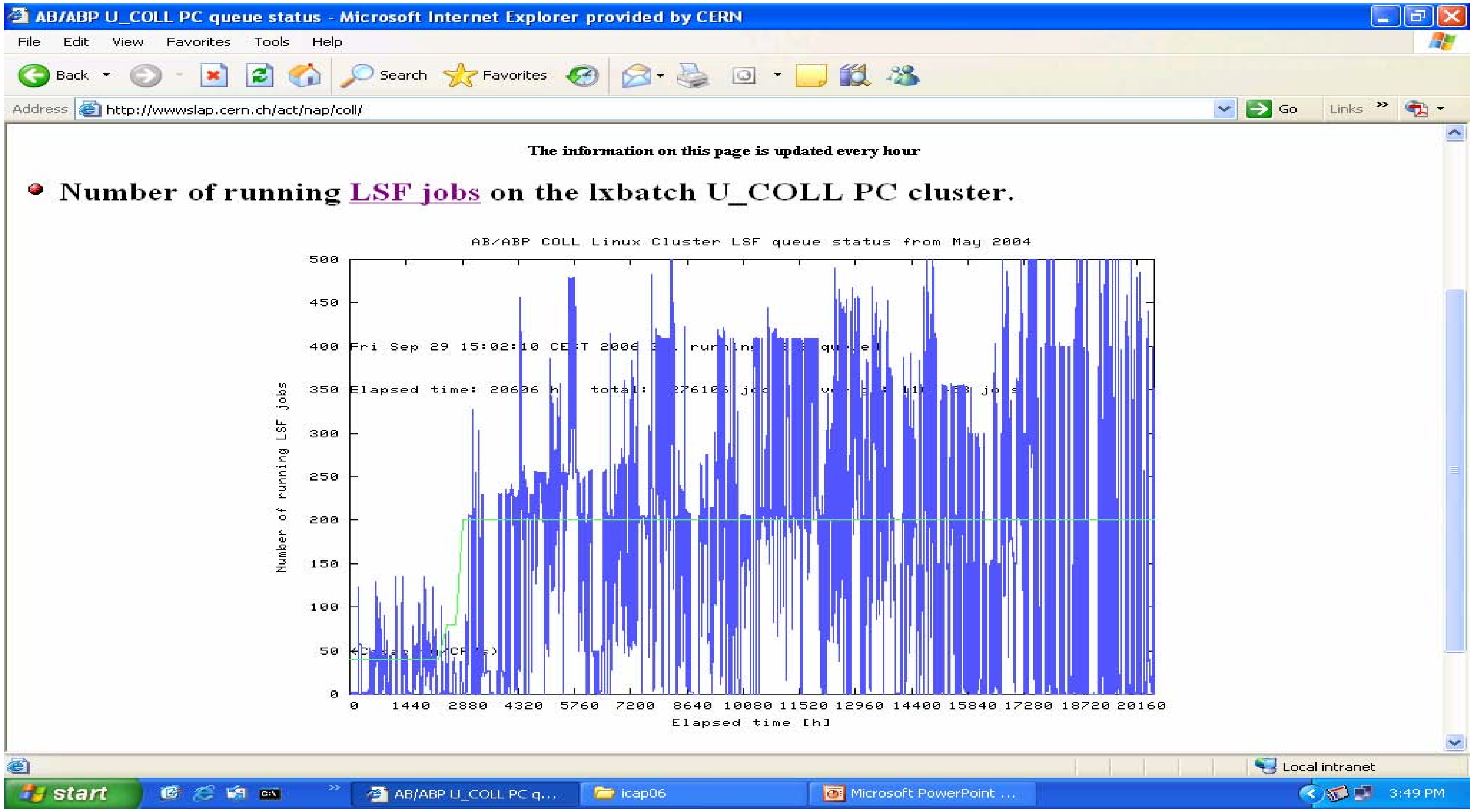
AB/ABP U_SLAP PC queue status - Microsoft Internet Explorer provided by CERN
File Edit View Favorites Tools Help
Back Search Favorites
Address http://wwwslap.cern.ch/act/nap/pcs/
Go Links
Local intranet
start
AB/ABP U_SLAP PC q...
icap06
Microsoft PowerPoint ...
3:47 PM

Number of running LSF jobs on the lxbatch U_SLAP PC cluster.

SLAP Linux Cluster LSF queue status from January 2005

```
Fri Sep 29 15:01:34 CEST 2006 0 running 0 queued
Elapsed time: 15203 h    total: 607314 jobs    average: 39.947 jobs
```

Beam Collimation


AB/ABP U_COLL PC queue status - Microsoft Internet Explorer provided by CERN

Address http://wwwslap.cern.ch/act/nap/coll/

The information on this page is updated every hour

- **Number of running LSF jobs on the lxbatch U_COLL PC cluster.**

The Idea (not original)

- Studies were still typically 1 tune, 60 seeds, up to 8 amplitudes, and 5 angles
- Use ~5000 Windows desktops at CERN to run SixTrack, a highly optimised LHC tracking program
- SixTrack is standard F77 and part of SPECFP2000
- Only 50KB (500KB) IN and < 2MB (6MB) OUT for ~ 1 to 10 hours CPU – ideal for distribution
- At least double the tracking capacity and potentially provide an order of magnitude increase for zero financial investment

Initial Problems

- No compatible WINDOWS graphics – just dummied out the HBOOK calls
- LineFeed in Windows text files – remove them on Linux when retrieving the result
- Lost Particle processing 1000 times slower – check more frequently to avoid NaNs and Infs

CPSS Project

- A. Wagner CERN/IT/WINDOWS provided a screen saver, Web Server and PERL interfaces for job submission and result retrieval
- SixTrack Checkpoint/Restart, vital for rapid release of the PC and long term run efficiency
- Transparent (almost) SixTrack run environment on Linux
- Worked well ………until occasional RESULT differences

First real problem

- 1500 jobs, 60 seeds, 5 amplitudes, 5 angles, (v64lhc.D1-D2-MQonly-inj-no-skew) for 10,000 turns
- The final results, the minimum, average and maximum Dynamic Aperture were within 1% of the lxbatch results
- The average DA was within 3 parts in 1000
- Tried 600 seeds/15,000 jobs as final pre-production
- ……BUT…..

Result Comparison

LSF/Linux Results

```
                                     Min   Ave   Max   Angle
v64lhc.D1-D2-MQonly-inj-no-skew5 1  11.27 12.20 13.17 15.00
v64lhc.D1-D2-MQonly-inj-no-skew5 2  12.18 13.69 15.46 30.00
v64lhc.D1-D2-MQonly-inj-no-skew5 3  13.90 14.83 16.14 45.00
v64lhc.D1-D2-MQonly-inj-no-skew5 4  16.29 17.32 18.08 60.00
v64lhc.D1-D2-MQonly-inj-no-skew5 5  15.50 16.30 17.34 75.00
```

Windows CPSS Results

```
v64lhc.D1-D2-MQonly-inj-no-skew5 1 11.17 12.21  12.97 15.00
v64lhc.D1-D2-MQonly-inj-no-skew5 2 12.18 13.66  15.24 30.00
v64lhc.D1-D2-MQonly-inj-no-skew5 3 13.53 14.80  16.09 45.00
v64lhc.D1-D2-MQonly-inj-no-skew5 4 16.41 17.31 -18.00 60.00
v64lhc.D1-D2-MQonly-inj-no-skew5 5 15.60 16.30  17.15 75.00
```

One bit too many…….

- Careful checking of duplicate results, for one specific seed, identified a difference in the distance in phase space, between a particle pair, when computed on Windows 2000 and on Windows XP.
- Exhaustive analysis identified one number 3.756403155274550e-09 was being input as HEX BE3022357D9B0651 on Windows 2000 as compared to HEX BE3022357D9B0650 on Windows XP (and on Linux)

…..but how often? how important?

- 600 fort.16 input files (Multipole Errors)
- 2364 blocks of 40 double-precision numbers
- 100,000 turns each involving 10,000 steps
- Quickly ran 2 times 600 jobs on W2000/XP
- 505 files affected (95 OK) with from 1 to 7 numbers being one bit too large
- Total of 1115 errors in 60 million numbers

A known problem

- Depends on Compiler/OS
- Could be fixed by (over-)specifying the input values
- Decided to buy the LAHEY-FUJITSU lf95 compiler for WINDOWS (already on Linux) to replace the obsolete COMPAQ compiler
- Surprisingly? Gave “IDENTICAL” results on Windows and Linux

IEEE 754

- Defines unique reproducible result for +, -, *, /, and sqrt – the correctly rounded result being the floating-point number closest to the exact result
- It is incomplete and open to interpretation
- Needs to be combined with the language standard
- Strict compliance conflicts with performance
- Does NOT cover Elementary Functions

Floating-Point issues (Double Precision)

- Extended (internal) 80-bit Precision EP
- (Double) rounding applied arbitrarily
- Fused Multiply Add
- SSE2

- DISABLE EP, in fact the default with lf95
- (“everything” else is disabled anyway)

EP Disabled

- NOT really practicable with libm and probably other libraries
- May introduce new problems in borderline evaluations
- Could affect performance (convergence)
- I contend that these cases are best solved otherwise

Compilation/Linking

- lf95 -o1 --tp -static
- -O/-o1 provide almost all optimisation
- -static is obviously required for portability
- --tp “generates Pentium code” (other options are --tpp Pentium Pro/Pentium II or --tp4 for Pentium 4)
- Success? Almost…………………

The beam-beam case

- While running some 400,000 2 hour jobs covering 1000 angles to prove CPSS
- Tried a study involving beam-beam interactions over a million turns
- Immediately detected a few result differences between INTEL IA32 and ATHLON AMD64 (also INTEL IA64)
- Traced back to an “exp” function - Not easy, but do-able with binary output
- Abandon the goal of reproducibility??? Abandon the whole idea!!!

Investigation

- Verified that IA64 was same as AMD64 (but see later)
- Found the log function similarly afflicted
- WWW search – insulted on a News Group
- Most problems/solutions eliminated because of the simple code generation
- Found several relevant libraries:MPFR, libultim IBM, libmcr SUN, crlibm Ecole Normale Superior LYON

The libraries

- MPFR – arbitrary precision – slow
- libultim – 800 bits – too much/not enough and no longer supported
- libcmr – arbitrary precision – slower
- crlibm – double precision – optimised and portable to any IEE-754 compliant CPU
- Finally adopted CRLIBM from the Ecole Normale Superieur at Lyon

crlibm

- Delivers correctly rounded double precision results for the elementary functions
- Proven to do so
- Performance “comparable” to libm on average (optimised versions available for different platforms)
- REQUIRES EP DISABLED
- Really more than I needed

Crlibm functions

- EXP, LOG, LOG10, SIN, COS, TAN
- ATAN, SINH, COSH
- ASIN, ACOS, ATAN2 are now available
  - I wrote them in terms of ATAN to be portable but NOT necessarily correctly rounded
- Each function has four rounding modes – nearest, up, down, to zero
- E.g. exp_rn, exp_ru, exp_rd and exp_rz

A Solution

- Installed crlibm (portable for Linux and Windows with gcc and Lahey-Fujisu C)
- The numerical differences disappeared
- Performance was at worst 10% slower in the most difficult beam-beam case (but on portable code)
- The only subsequent numerical differences have been traced to failing computers (3 desktops and 1 lxbatch)

Some simple test results

- ULP – One Unit in the Last Place of the mantissa of a floating-point number (one part in roughly 10**16)
  - libm/crlibm IA32: 304 differences of 1ULP
  - libm IA32/IA64: 5 differences of 1ULP
  - libm IA32/AMD64: 7 differences of 1ULP
  - libm IA64/AMD64: 2 differences of 1ULP
  - libm/libm NO EP: 134623 differences of 1ULP
- NO differences with exp_rn
- 1,000,000 exp calls with random arguments between -0.5 and 0.5

…and with lf95

  - lahey/crlibm IA32: 134645 differences of 1ULP
  - lahey IA32/IA64: 7 differences of 1ULP
  - lahey IA32/AMD64: 7 differences of 1ULP
  - lahey IA64/AMD64: 4 differences of 1ULP

- NO differences with exp_rn

crlibm exp performance

Pentium 4 Xeon gcc 3.3			
	Average	Min	Max
libm	365	236	5528
crlibm	432	316	41484
libultim	210	44	3105632
mpfr	23299	14636	204736

When quadruple precision is not enough – The Table Maker’s Dilemma

- Rounding the approximation of f(x) is not always the same as rounding f(x)
- Worst case for exp(x), x=7.5417527749959590085206221e-10
- Binary example with x=1.[52]1*2-53
- exp(x)=1.[51]001[104]1010101…correctly rounded to 1.[51]01 in Double
- quad (112 bit) approximations, 1.[51]010[58]00, 1.[51]001[58]11, 1.[51]001[58]10, are all within 1 Quad ULP, but rounding the last gives an incorrectly rounded Double result.

BOINC

- The Berkeley Open Infrastructure for Network Computing (c.f SETI@home)
- LHC@HOME – up to 60,000 computers
- Running LHC Tracking (intermittently) for more than twelve months so far
- Beam-beam estimated to need 600,000 one million turn 10 hour jobs
- Currently the service is being moved to UK

BOINC ………

- Some 400,000 cases completed
- Every jobs is run three times (at least) and only identical results are accepted (NO EPSILON required)
- Estimate 3% of results are erroneous, which is average for BOINC applications, but these are of course rejected

Conclusions and Questions

- Am I obsessed about a numerical difference of 1ULP? It IS a problem for tracking studies, and other “divergent” applications like climate prediction or molecular dynamics
- Having eliminated ALL numeric differences SixTrack can be run on any Pentium or compatible PC
- Support IEEE and the revised standard

The next steps

- Add parentheses to SixTrack code to provide IDENTICAL results with different compilers, different optimisation levels, and on *any* IEEE-754 compliant machine
- Extend to C/C++ C99 compliant applications and compilers?
- Other applications?
- Verify the GRID as an alternative to BOINC and CPSS