HEPIX Summary (Part 3)

Spring meeting 2006
Caspur, Rome

S.Jarp, IT/CERN

Some sessions which I found interesting and representative (and I hope you will)

Tuesday morning

- Batch:

10:30 **Batch Systems SIG** (Convener: Tony Cass/CERN)
- *Passing information to the CE (gLite)* (Francesco Prelz/INFN)
- *Interfacing BLAHP with LSF - Status Report* (Ulrich Schwickerath/CERN)

-11.30 **Batch Systems SIG, contd** (Convener: Tony Cass/CERN)
- *Experiment plans for batch system use (ATLAS)* (Laura Perini/INFN)
- *Experiment plans for batch system use (CMS)* (Stefano Belforte/INFN)
- *Experiment plans for batch system use (LHCb)* (Andrei Tsaregorodtsev/CPPM)
- *Experiment plans for batch system use (ALICE)* (Federico Carminati/CERN)
- *Conclusion & discussion* (Tony Cass/CERN)

Wednesday part 1

- Keynote + Optimisation:

09:00 **Plenary HEPiX/GDB talk**
- *Key challenges for Computer Centre Managers supporting LHC computing*
Speaker: Les Robertson/CERN

09:30 **Optimisation and bottlenecks** (Convener: Wojciech Wojcik/IN2P3)
- *Understanding and addressing performance issues in HEP* (Sverre Jarp/CERN)
- *Code/compiler problems and how to reach an improvement* (Rene Brun/CERN)
- *Usage of BQS resources to control bottlenecks upstream* (Julien Devemy/IN2P3)

11:30 **Optimisation and bottlenecks, contd** (Convener: Wojciech Wojcik/IN2P3)
- *Optimisation of dCache and DPM* (Greig Cowan/U.Edinburgh)
- *Conclusions and future plans* (Wojciech Wojcik/IN2P3)

LCG - The Worldwide LHC Computing Grid

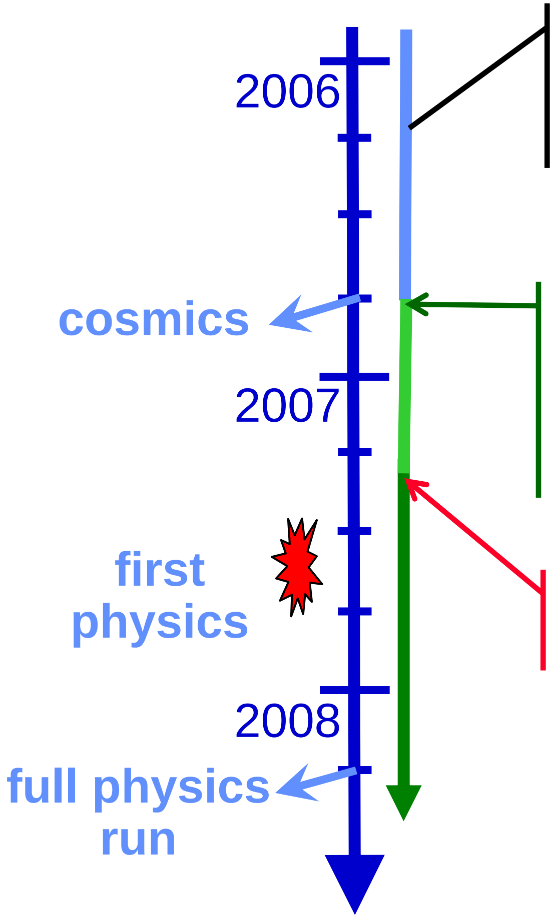
LHC Data Analysis

Challenges for 100 Computing Centres in 20 Countries

HEPiX Meeting
Rome
5 April 2006

Les Robertson
LCG Project Leader

LCG Service Deadlines



Pilot Services –
stable service from 1 June 06

LHC Service in operation – *1 Oct 06*
over following six months ramp up to full operational capacity & performance

LHC service commissioned – *1 Apr 07*

les.robertson@cern.ch

Conclusions

- LCG will depend on
  - ~100 computer centres – run by you
  - two major science grid infrastructures – EGEE and OSG
  - excellent global research networking
- We have
  - understanding of the experiment computing models
  - agreement on the baseline services
  - good experience from SC3 on what the problems and difficulties are
- Grids are now operational
  - ~200 sites between EGEE and OSG
  - Grid operations centres running for well over a year
  - > 20K jobs per day accounted
  - ~15K simultaneous jobs with the right load and job mix

BUT – a long way to go on reliability

les.robertson@cern.ch

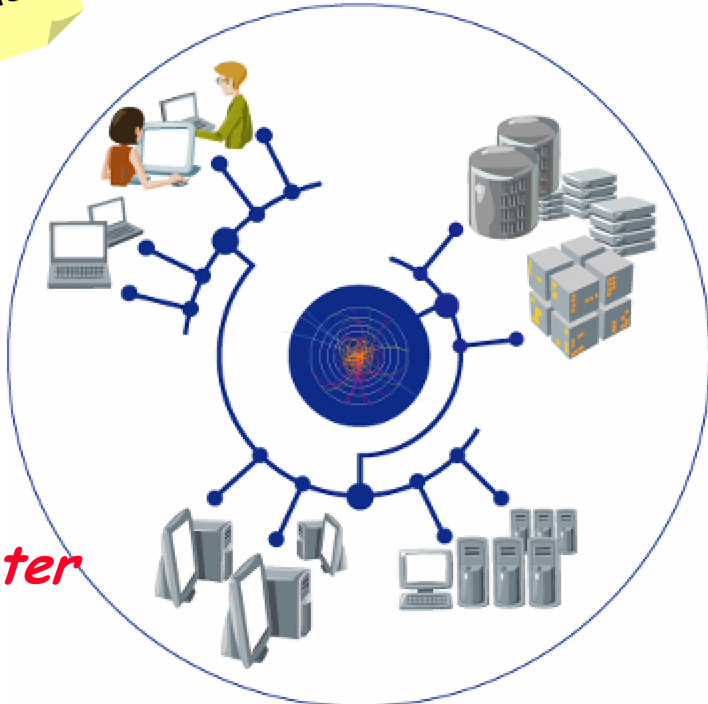
- The Service Challenge programme this year must show that we can run reliable services

- Grid reliability is the product of many components
  – middleware, grid operations, computer centres, ….

- Target for September
  - 90% site availability
  - 90% user job success

Too modest?
Too ambitious?

- Requires a major effort by everyone to monitor, measure, debug



First data will arrive next year

NOT an option to get things going later

Experiment plans for batch system usage

Federico Carminati

HEPiX

Rome, April 4, 2006

ALICE computing model

- For pp similar to the other experiments
  - Quasi-online data distribution and first reconstruction, calib and alignment at T0; prompt analysis @CAF
  - Further reconstructions at T1's
- For AA different model
  - Calibration, alignment, pilot reconstructions, prompt analysis@CAF and partial data export during data taking
  - Data distribution and first reconstruction at T0 in the four months after AA run (shutdown)
  - Further reconstructions at T1's
- **T0: First pass reconstruction, storage of RAW, calibration data and first-pass ESD's**
- **T1: Subsequent reconstructions and scheduled analysis, storage of a collective copy of RAW and one copy of reconstructed and simulated data to be safely kept, disk replicas of ESD's and AOD's**
- **T2: Simulation and end-user analysis, disk replicas of ESD's and AOD's**

Job submission

- Job agents
  - Sent only when needed
  - Avoid waste of resources and "useless" updates of the ALICE Job Catalogue
  - Eliminate "black hole" effect
- Job location determined by the data location
- WN outbound connectivity required
  - We are working on removing this constraint
- System used for large production
  - 22,500 jobs, 540 KSi2K hours, 20TB
  - 2.5% inefficiency thanks to job agents

Batch systems use in ALICE

- Past use
  - Through AliEn – use of all flavours of batch schedulers (LSF, PBS, BQS, SGE, Condor) at many computing centres worldwide
  - Few separate queues for different jobs types
  - Job priorities handled in the central TQ
- Present status
  - **Practically no direct access to batch queues: shielded by the GRID (LCG, OSG, ARC) CE**
  - **Middleware is increasingly 'taking away' the functions of the batch systems (job prioritization based on job length, queuing)**
  - **Fewer users submit jobs locally: ultimately all offline computing tasks in ALICE will be performed on the GRID (production, calibration, analysis), users will submit all jobs to the GRID interface**

ALICE requirements

- We see the interaction with the batch systems (specific submission commands, error handling and reporting, log and output files, etc...) as part of the GRID service
  - Therefore we have no special preferences to the type of batch systems deployed at the sites
- **Connected with this we still do not have a properly secured sandbox**
- For that we would probably need Job Agent to
  - Start virtual machine
  - ...or start another process under different user id using glexec/sudo mechanism
- However this is not a show-stopper for us

ALICE requirements

- From practical point of view, presently we require
  - **One single long ALICE-specific queue**
  - **Would like a uniform publishing of queue length in kSI2k•h (ultimately also a GRID function) across sites**
  - **Ability to guarantee the share of computing resources for ALICE**
  - **Ability to specify the amount of memory needed by a job**
  - **A minimum memory requirement of 2 Gb per core**
  - **Scratch space of several GB**
  - **A shared home directory for software installatinos etc..**

GridPP
UK Computing for Particle Physics

Tier-2 optimisation of dCache and DPM

Greig A Cowan
University of Edinburgh
Graeme A Stewart, Jamie K Ferguson
University of Glasgow

GridPP
UK Computing for Particle Physics

A 'typical' Tier-2

No such thing as 'typical', but there are some commonalities, i.e.

- Limited hardware resources:
  - One or two nodes attached to a few TB of RAID'ed disk.
  - Some storage NFS mounted from another disk server.
  - No tape storage.
- Limited manpower to spend on administering/configuring an SRM.
- Choice of SRM solutions (dCache, DPM, StoRM . . .)
- Require the SRM they choose to be optimised in order to be able to handle the data flows that are expected when the LHC comes online.
  - GridPP service challenge set target that all T2s should be able to sustain T1$\rightarrow$T2 transfer rate of $\geq$ 300Mb/s.

Best results observed with xfs + SLC 3.0.6 and a 2.6 kernel.
FTS parameters: Nf ~10, Ns = 1.

Plea for assistance so that Tier2 centres can benefit from existing knowledge in this area → P.Kelemen's talk

Some hints to improve compilation time and execution performance (René Brun)

- **Time to compile**
  - May be a problem in some experiments
  - Some recipes for improvement
- **Shared libs**
- **Improving the execution time**
  - Code inlining (good and bad aspects)
  - Using the right collection classes
  - Profiling tools
- **Differences between compilers or compiler versions**
- **Ready for Multithreading**

Example with smatrix

TestKalman [nx,ny] : kalman_win7.1

	2	3	4	5	6	7	8	9	10
2	0.41 0.40 1.01	0.55 0.56 1.29	0.80 0.79 1.65	1.26 1.31 2.18	2.16 2.36 3.16	3.91 3.84 7.13	5.70 5.14 8.85	7.43 6.97 11.24	9.66 8.90 13.66
3	0.52 0.51 1.24	0.70 0.70 1.50	0.98 0.98 1.97	1.49 1.52 2.61	2.43 2.62 3.68	4.35 4.15 7.82	6.07 5.46 9.53	8.23 7.55 12.26	10.09 9.17 14.95
4	0.63 0.62 1.50	0.85 0.85 1.88	1.24 1.17 2.19	1.73 1.77 3.09	2.79 2.86 4.31	4.77 4.46 8.53	6.65 5.93 10.56	8.64 7.93 13.77	10.86 10.01 16.58
5	0.78 0.83 1.81	1.04 1.09 2.24	1.41 1.45 2.91	2.11 2.10 3.49	3.12 3.22 5.02	5.12 4.90 9.40	7.17 6.53 11.56	9.64 8.70 14.88	11.45 10.56 17.61
6	0.85 0.98 2.13	1.16 1.29 2.65	1.68 1.72 3.40	2.28 2.49 4.37	3.50 3.72 5.49	5.57 5.44 10.36	8.12 7.07 12.53	9.94 9.22 16.09	12.50 11.42 19.24
7	1.04 1.10 2.44	1.50 1.48 3.09	2.01 1.99 3.95	2.79 2.80 4.95	4.03 4.15 6.47	6.24 5.89 10.88	8.48 7.64 13.44	10.76 9.80 17.59	13.30 11.96 20.76
8	1.22 1.26 2.81	1.69 1.71 3.57	2.30 2.30 4.48	3.18 3.16 5.57	4.59 4.57 7.28	6.89 6.47 12.02	9.24 8.69 14.23	11.67 10.78 18.69	14.35 13.03 22.77

N1,N2 <= 6	36.51 37.96 66.81	N1,N2 > 6	261.15 242.13 421.54	All N1,N2	297.67 280.08 488.35

SMatrix_Sym SMatrix TMatrix **SMatrix_Sym better than TMatrix**

TestKalman [nx,ny] : kalman_solaris.5.9

	2	3	4	5	6	7	8	9	10
2	2.29 1.39 2.49	4.53 2.41 2.95	7.49 3.52 3.84	13.36 5.57 5.15	27.92 9.88 8.18	30.68 20.13 34.74	42.12 29.90 42.52	56.27 41.89 51.08	74.79 56.08 61.38
3	3.36 2.05 2.83	6.28 3.43 3.49	9.88 5.09 4.74	17.60 7.79 6.23	33.32 12.81 9.61	37.58 24.26 36.25	51.27 35.10 44.61	69.29 50.11 53.69	88.88 66.09 63.78
4	4.70 2.92 3.50	8.39 4.82 4.41	13.02 7.16 5.68	21.30 10.45 7.46	38.09 16.27 11.42	44.86 28.23 38.35	62.55 43.15 47.23	83.96 60.94 56.35	108.25 79.72 67.32
5	6.45 3.84 3.87	11.09 6.42 5.10	16.75 9.42 6.78	26.01 13.43 8.88	45.35 20.22 12.96	52.92 32.57 40.86	73.96 50.84 49.51	100.57 72.06 59.60	127.69 94.24 70.80
6	8.77 5.36 4.58	14.55 8.67 6.12	21.27 12.45 8.33	32.27 17.49 10.55	51.35 25.37 14.90	63.23 39.55 43.39	87.57 60.54 52.76	118.44 84.29 63.18	152.52 112.31 75.01
7	12.58 6.85 5.27	20.21 10.88 7.13	29.16 15.45 9.41	42.12 21.34 12.36	64.82 29.96 17.47	78.81 44.96 45.91	107.49 68.27 55.53	142.03 96.24 66.99	183.05 128.52 79.38
8	17.68 10.79 6.08	28.33 17.19 8.30	40.40 24.55 10.98	57.23 33.55 14.26	84.57 46.08 19.58	103.45 64.54 48.60	139.67 95.46 58.98	184.39 132.12 70.57	232.32 170.91 83.94

N1,N2 <= 6	445.38 218.23 164.08	N1,N2 > 6	3095.72 2099.65 1673.16	All N1,N2	3541.10 2317.89 1837.24

SMatrix_Sym SMatrix TMatrix **SMatrix_Sym better than TMatrix**

TestKalman [nx,ny] : kalman_slc3_gcc323

	2	3	4	5	6	7	8	9	10
2	0.30 0.33 0.86	0.38 0.44 1.00	0.56 0.64 1.39	0.88 0.98 1.69	1.54 1.64 2.96	5.77 5.84 5.22	8.00 7.81 6.06	10.41 10.23 7.41	14.36 14.58 9.03
3	0.37 0.46 1.01	0.56 0.60 1.16	0.72 0.99 1.59	1.10 1.29 1.96	1.84 2.03 3.40	6.24 6.33 5.48	8.19 8.17 6.64	10.97 10.76 7.61	14.37 14.02 9.86
4	0.47 0.61 1.16	0.63 0.76 1.42	0.89 1.03 1.72	1.39 1.48 2.48	2.16 2.27 3.67	6.71 6.43 6.14	9.04 8.92 6.95	11.71 11.37 8.85	15.07 14.69 10.33
5	0.60 0.78 1.28	0.85 1.03 1.55	1.19 1.28 2.35	1.71 1.80 2.69	2.58 2.70 4.44	7.03 6.79 6.60	9.52 9.18 8.34	12.41 12.03 9.44	15.74 16.00 12.16
6	0.77 0.96 1.59	1.26 1.22 2.09	1.49 1.60 2.42	2.13 2.17 3.58	3.06 3.12 4.61	7.81 7.39 7.55	10.19 9.90 8.09	12.98 12.53 10.36	17.56 16.92 11.58
7	0.96 1.25 1.75	1.33 1.49 2.17	1.77 1.99 3.03	2.46 2.57 3.53	3.47 3.62 5.48	8.24 8.08 7.90	10.56 10.13 9.50	13.08 12.72 10.62	18.03 16.96 14.14
8	1.14 1.48 2.05	1.68 1.79 2.81	2.15 2.33 3.02	2.95 3.14 4.67	4.07 4.27 5.59	8.99 8.79 8.69	11.47 11.37 9.34	14.48 14.36 12.29	19.15 18.07 13.71

N1,N2 <= 6	29.45 32.23 54.07	N1,N2 > 6	340.08 334.29 283.99	All N1,N2	369.53 366.52 338.05

SMatrix_Sym SMatrix TMatrix **SMatrix_Sym better than TMatrix**

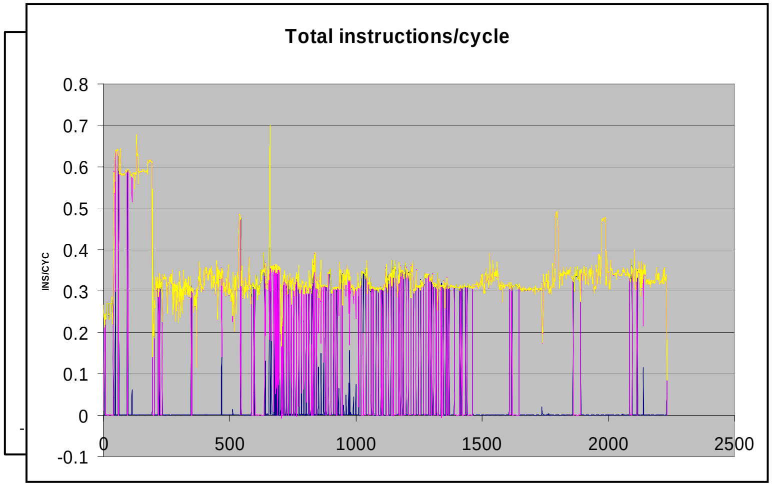
MultiCore: Impact on ROOT

- There are many areas in ROOT that can benefit from a multi core architecture. Because the hardware is becoming available on commodity laptops, it is urgent to implement the most obvious asap.
- Multi-Core often implies multi-threading. There are several areas to be made not only thread-safe but also thread aware.
  - PROOF obvious candidate. By default a ROOT interactive session should run in PROOF mode. It would be nice if this could be made totally transparent to a user.
  - Speed-up I/O with multi-threaded I/O and read-ahead
  - Buffer compression in parallel
  - Minimization function in parallel
  - Interactive compilation with ACLIC in parallel
  - etc..

Typical profile
Stop press: G4ATLAS

Samples	Self %	Total %	Module
11767458	36.64%	36.64%	libG4geometry.so
5489494	17.09%	53.73%	libG4processes.so
2283674	7.11%	60.85%	libG4tracking.so
2146178	6.68%	67.53%	libm-2.3.2.so
2057144	6.41%	73.93%	libstdc++.so.5.0.3
1683623	5.24%	79.18%	libc-2.3.2.so
933872	2.91%	82.08%	libCLHEP-GenericFunctions-1.9.2.1.so
685894	2.14%	84.22%	libG4track.so
655282	2.04%	86.26%	libCLHEP-Random-1.9.2.1.so
524236	1.63%	87.89%	libpthread-0.60.so
283521	0.88%	88.78%	libCLHEP-Vector-1.9.2.1.so
265656	0.83%	89.60%	libG4materials.so
205836	0.64%	90.24%	libG4Svc.so
197690	0.62%	90.86%	libG4particles.so
190272	0.59%	91.45%	ld-2.3.2.so
150757	0.47%	91.92%	libCore.so (ROOT)
149525	0.47%	92.39%	libFadsActions.so
126111	0.39%	92.78%	libG4event.so
123206	0.38%	93.16%	libGaudiSvc.so

G4Atlas simulation (3 events)



Total instructions

Cycles	6252 * 10^9
Total inst	2136 * 10^9
TOT INS/CYC	0.342 (0.684 on one CPU)

Floating-point instructions

FP	397 * 10^9
FP/TOT	0.186

LD/ST/BR instructions

LD	814 * 10^9
LD/TOT	0.38
L2LM	60 * 10^9
L2LM/LD	0.074

ST	528 * 10^9
ST/TOT	0.247
L2SM	0.60 * 10^9
L2SM/ST	0.00113

BR_TP	218 * 10^9
BR_TM	5.4 * 10^9
BR_TP/TOT	0.097
BR_TM/TOT	0.00252

"That's
all
folks!"