Atlas Managed Production on Nordugrid

Alex Read
Mattias Ellert (Uppsala), Katarina Pajchel, Adrian Taga

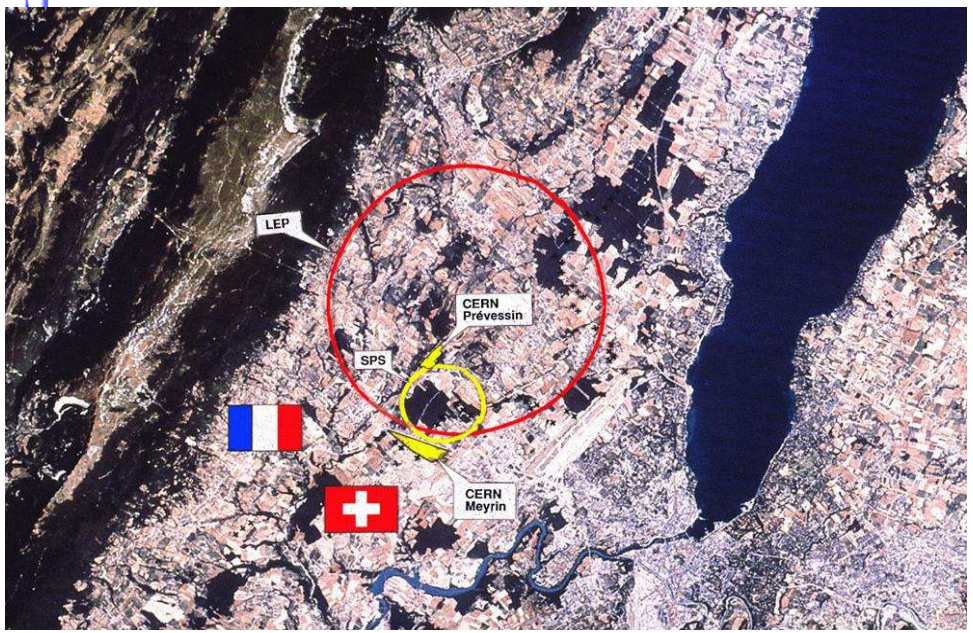
University of Oslo

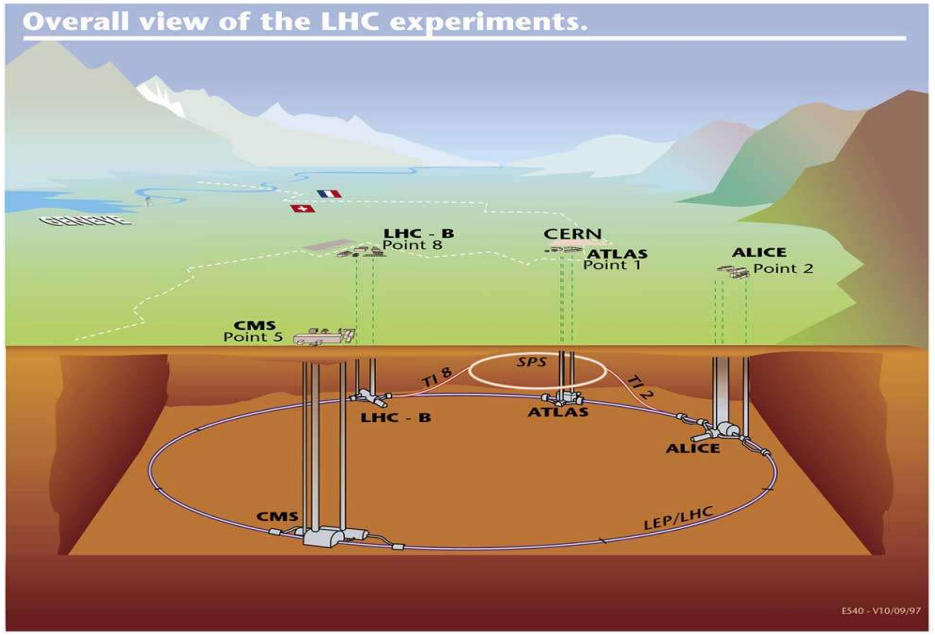
November 7-9, 2006

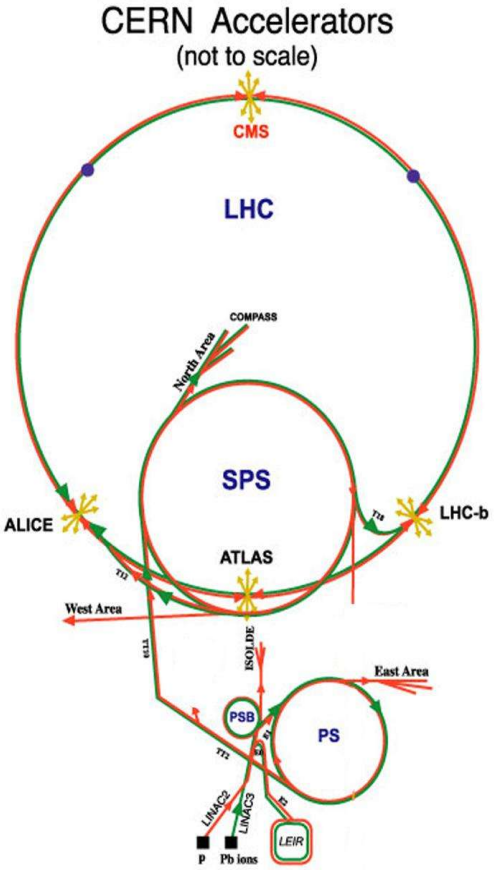
Outline

1. LHC/ATLAS
2. Background
3. The Atlas Production System
4. Dulcinea – the NG executor
5. Performance
6. Job throughput
7. Error analysis
8. Challenges
9. Conclusions

CERN - LHC

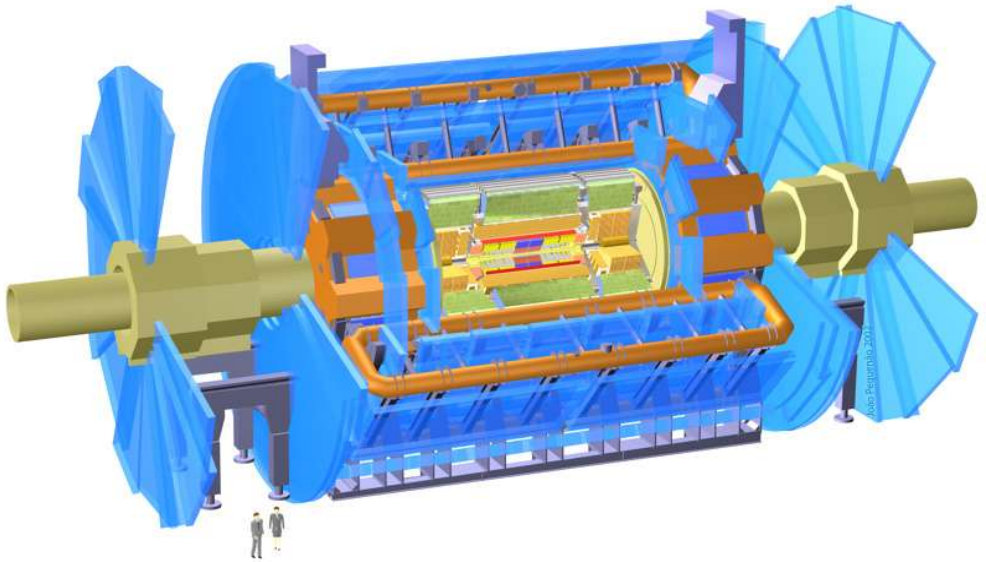






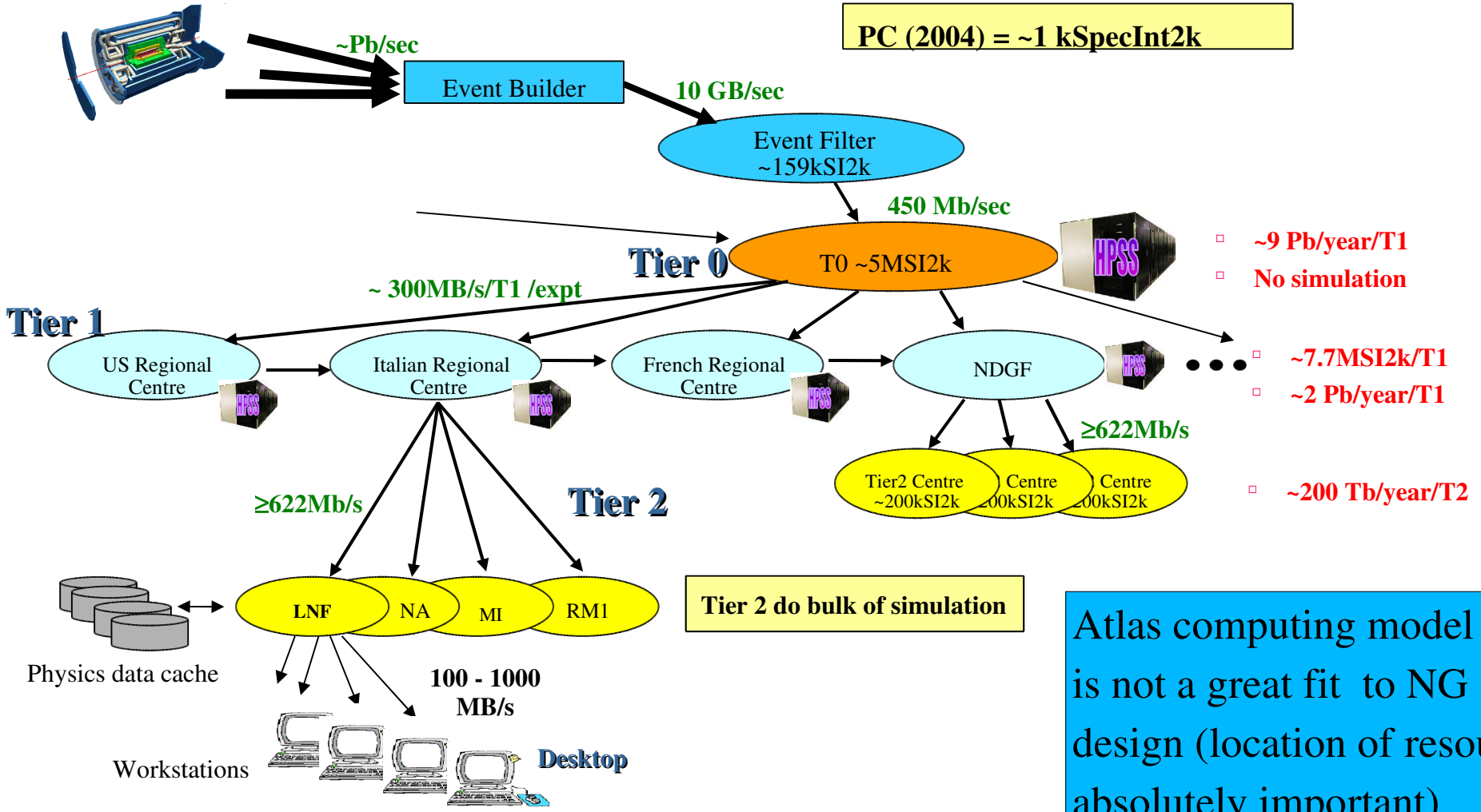
- p-p (Pb-Pb) collisions with E=14 TeV (about 15k proton masses)

Atlas



- Can we understand how pointlike particles get mass (Higgs)?
- Is nature supersymmetric?
- Are there extra spacetime dimensions?
- How did the universe get to be matter-antimatter asymmetric?
- Does dark matter consist of heavy, non-interacting particles?

Atlas Computing System



Atlas computing model is not a great fit to NG design (location of resources absolutely important)

Background

- Nordic groups joined the first so called Atlas Data Challenge (DC1) ~ 15 M events, full chain, FORTRAN framework (spring 2002 to spring 2003), carried out on Nordugrid resources.
- DC2 - second large scale production ~ 15 M events, full production chain using Geant4 using C++ algorithms and Athena framework (mid-2004 to mid-2005).
- Large scale production for Atlas Physics workshop, June 2005 in Rome. ~ 5 M events, short term, new data format.
- From November 2005: DC3 - Computer System Commissioning (CSC) **ongoing** (!) large-scale production
  - test of Atlas/LCG computing model and system
  - physics validation (real users)
- Commissioning run November, 2007
- Physics data-taking starting in 2008

Production system dependencies

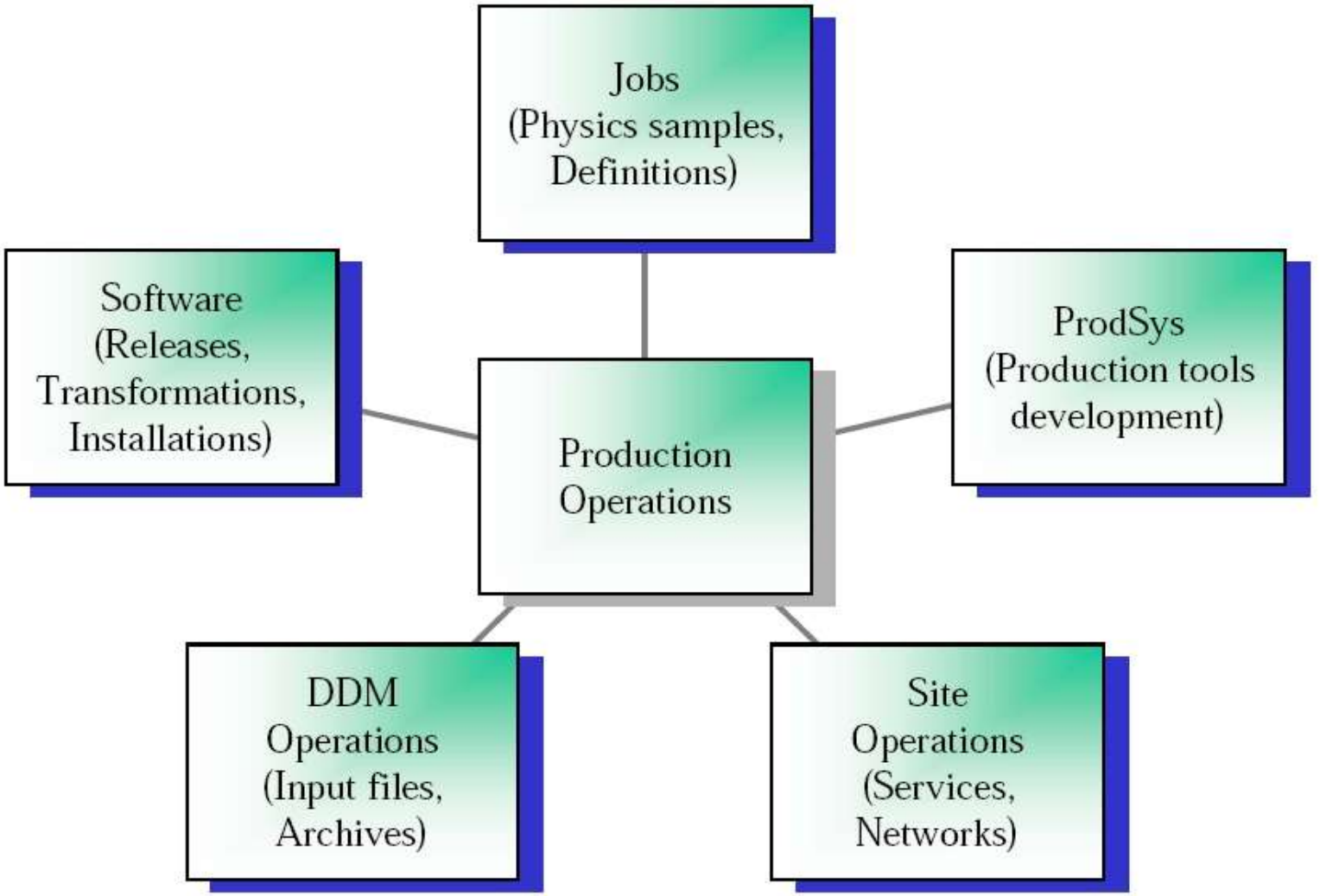


Fig: Kaushik De

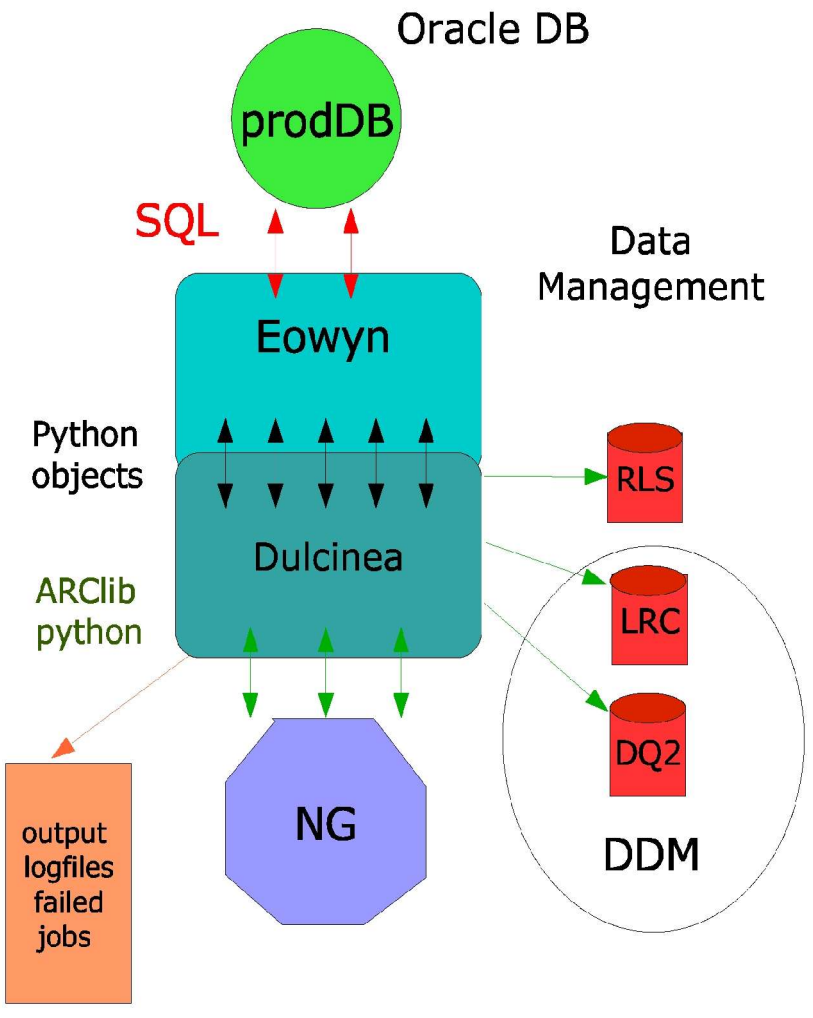
Atlas Production System - ProdSys

- Where the production starts:
  - A Physics group provides validated joboptions and a sample request
  - Jobs are defined in the production database (prodDB) as a task with a number of jobs
  - Tasks are assigned to one of the 3 grids (OSG, LCG/EGEE, NorduGrid)
- The executors of the different grids pick up the various jobs from prodDB and run them.
  - Event generation - physics simulation tools: "evgen" jobs
  - Detector simulation+digitization (Geant4): "digit" jobs
  - Reconstruction: "recon" jobs
  - Merging recon outputs: "merge" jobs
- Each job type has its special features (walltime, memory, I/O, errors)

Atlas Software

- Several GB per release
  - several pre-releases before a large production release is validated
  - installed for the moment by non-Grid means, releases available via pacman (standard sl3~RHEL3 kit) or rpm's (RHEL4, FC1,2,3, etc).
    - kit validation neglected the last ½ year, but it will return – NG was first to do the KV via the grid
  - job “transformations” (scripts) are more dynamic, installed in session for each job (pacman mirror of cache on http-server in Oslo) – common for many jobs, could be made more efficient
  - detector database files treated like input files per job
    - NG's grid-manager cache manages this masterfully
  - job wrapper completes the RTE, executes the Atlas command (athena.py) and collects information about the output files (guid, md5sum, lcn, size, date)

Production Supervisor<-->Executor



http://guts.uio.no/atlas/jobinfo/

- Supervisor (Eowyn) and executor (Dulcinea for NG) are written in Python.
- No brokering in Eowyn, common for all grids.
- Eowyn and executors exchange python objects.
- Eowyn runs in cycles calling routines from the grid-specific executors.
  - Query for new jobs
  - Prepare and submit jobs
  - Check status and if finished post-process jobs
  - Clean post processed jobs
- Throughput (NG): ~4k jobs/day/executor
  - don't know for sure that N executors gives 4k*N jobs

Atlas ProdSys

- The output (data, log files) are stored on NorduGrid Storage Elements (SE)
  - gacl registration (to be stopped in favor of directory gacl's)
  - file attributes added to RLS
  - registration in LRC (new, simple file catalog inspired by ATLAS/OSG)
  - registered in the Atlas Distributed Data-management System (DDM) DQ2
- Users and other computing centers can then download or “subscribe” (built on top of FTS) to datasets.
- If (when!) jobs fail
  - retry automatically 3 times (clean possible previous outputs from RLS – lose a few s/job)
  - save logfiles on http-server (done automatically)
  - report persistent cluster errors to sysadmins
  - report persistent middleware errors to Nordugrid
  - report persistent software errors to Atlas

Dulcinea

- NG implementation of executor routines.
- Based on python binding to ARClib (grid application toolbox)
- Create an xrsl job description from jobdef in prodDB
- Submit up to 100 jobs at a time to enabled clusters
  - ARClib does the brokering
    - spoiled by badly configured clusters
    - full brokering done once for (up to) N=100 jobs
    - brokering then uses locally updated broker information for jobs 2 through N
  - crashes here orphans jobs
    - Aborts are fixed
    - still some occasional segment violations and hangups
      - almost always coincident with problems at a cluster

Developments

- Huge improvements in ARC and Dulcinea over the past year
  - brokering works (when clusters configured correctly)
  - infosys more correct (cluster config, Condor backend)
  - memory leaks in ARClib stopped (was MB/job a year ago)
  - `ngresume` on clusters with ARC 0.5.x saves time and resources
  - better exception handling and error reporting (to prodDB, logger currently ~useless)
  - RLS stability dramatically improved with new release, dedicated hardware, increased number of connections
  - uploading/downloading doesn't hang in 0.5.56
  - SE's upgraded to 0.5.56 much more stable (no silly max-connections)
    - ingrid an exception (can't use or build latest NG-globus)

Resources

- Only sites intending to install Atlas SW and allowing production jobs to be run
- 2 (soon 3) executors
- Great support from the sysadmins
- CPU's are shared, lots of competition
  - record is ~400 jobs concurrently
- Storage: 81.5 TB, 22 TB free
  - how much of this can be used by Atlas is not clear
  - all gsiftp
- Atlas 12.0.3 – 16 sites, 1445 CPU's
- Atlas 12.0.31 – 12 sites, 786 CPU's
  - large production with 12.0.4 will start soon

ATLAS Grid Monitor

2006-11-02 CET 16:30:10

Processes: Grid Local

Country	Site	CPUs	Load (processes: Grid+local)	Queueing
Denmark	Benedict - Aalborg pr>	50	2+0	0+0
	Morpheus (NBI)	17	0+0 (no queue info)	0+0
Norway	Bergen Grid Cluster	10	0+3	0+0
	EPF (UiO/FI)	27	14+1	5+-2
	Norgrid@NTNU	58	0+37	3+3
	Titan (USIT/UiO)	339	0+335	0+3386
	UiO Grid	10	2+8	0+0
Slovenia	SiGNET	149	51+87	0+0
Sweden	Bluesmoke (Swegrid,NS>	96	28+2	267+48
	Hagrid (SweGrid, Uppm>	100	96+0	174+0
	Hive (Swegrid, UNICC)	100	96+0	46+2
	Ingrid (SweGrid,HPC2N)	100	88+0	20+-5
	Sigrid (SweGrid, Luna>	99	89+10	358+0
	SweLanka SE	10	3+0	0+0
Switzerland	Bern ATLAS Cluster	8	6+1	0+0
	Geneva-DINF/DPNC	25	19+0	0+0
	Geneva-DPNC	8	0+0	94+0
	PHOENIX (CSCS)	33	0+22	351+7
	UBELIX (University of>	272	0+243	0+0
TOTAL	*19 sites*	*1511*	*494 + 749*	*1318 + 3439*

Performance

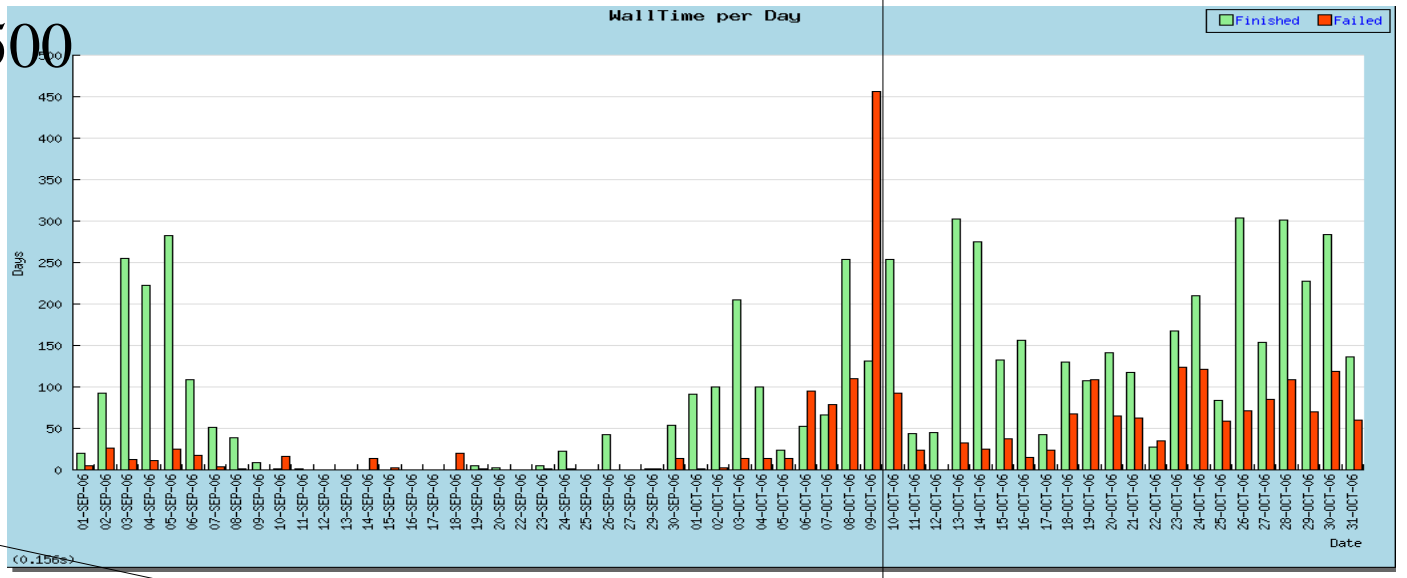
#	EXECUTORTYPE	FINJOBS	FINCPU	FINWALL	FAILJOBS	FAILCPU	FAILWALL	SUBMITTED	RUNNING	JOBEFF	WALLEFF
1	CondorG	135837	5373769671	4592992198	422991	1500744786	2168015918	101157	112532	24.3	67.93
2	LCG-DQ	32164	1110829697	1077391357	82948	436900489	501135593	99357	56609	27.94	68.25
3	panda	143753	5512453508	5633088572	107022	2528041987	3156508751	208651	100620	57.32	64.08
4	LCG	42056	2033342036	1878189016	63251	179133182	437217599	71852	45443	39.93	81.11
5	Dulcinea	85213	1202486150	1122289706	40524	363665117	530819963	16750	23757	67.77	67.88
6	Cronus	910	14222765	12477122	1515	10004983	9364973	123	271	37.52	57.12

- Table for June-October this year (159k jobs since 11/2005, ~70% efficiency)
- But ~impossible to compare efficiencies
  - Atlas failure rate and walltime/job **highly** task dependent
  - NG doesn't use walltime to stage data

Performance

Walltime/day
Sept/Oct 2006

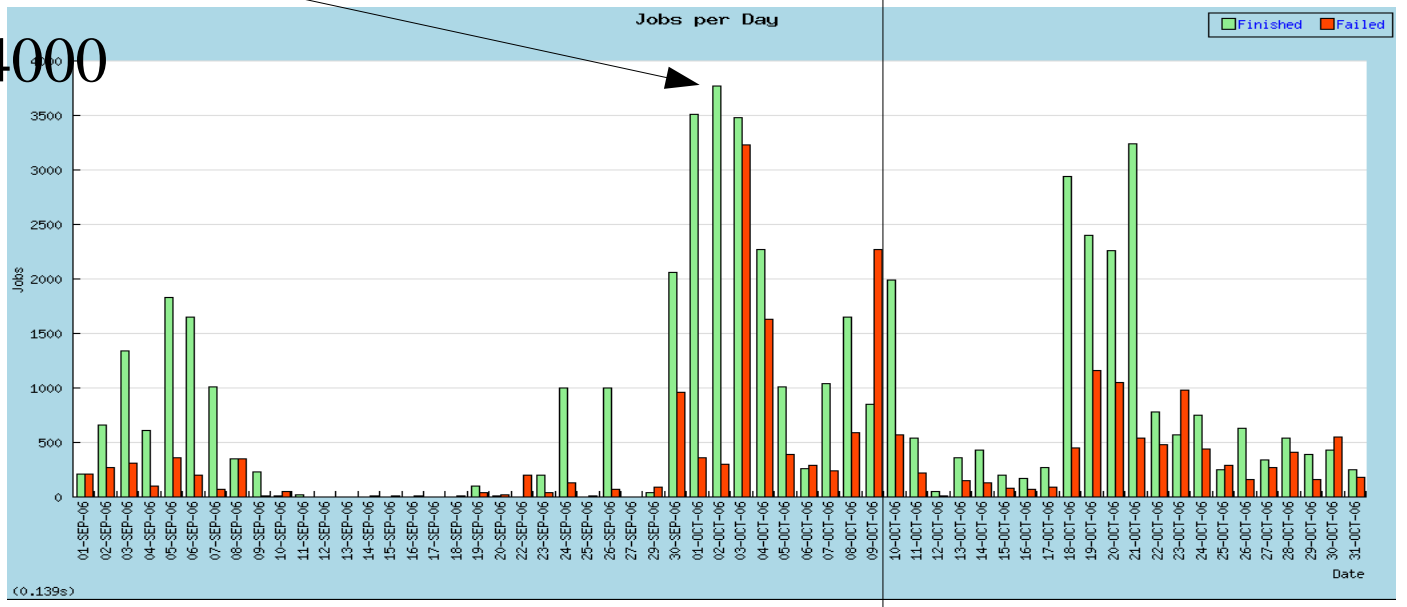
500



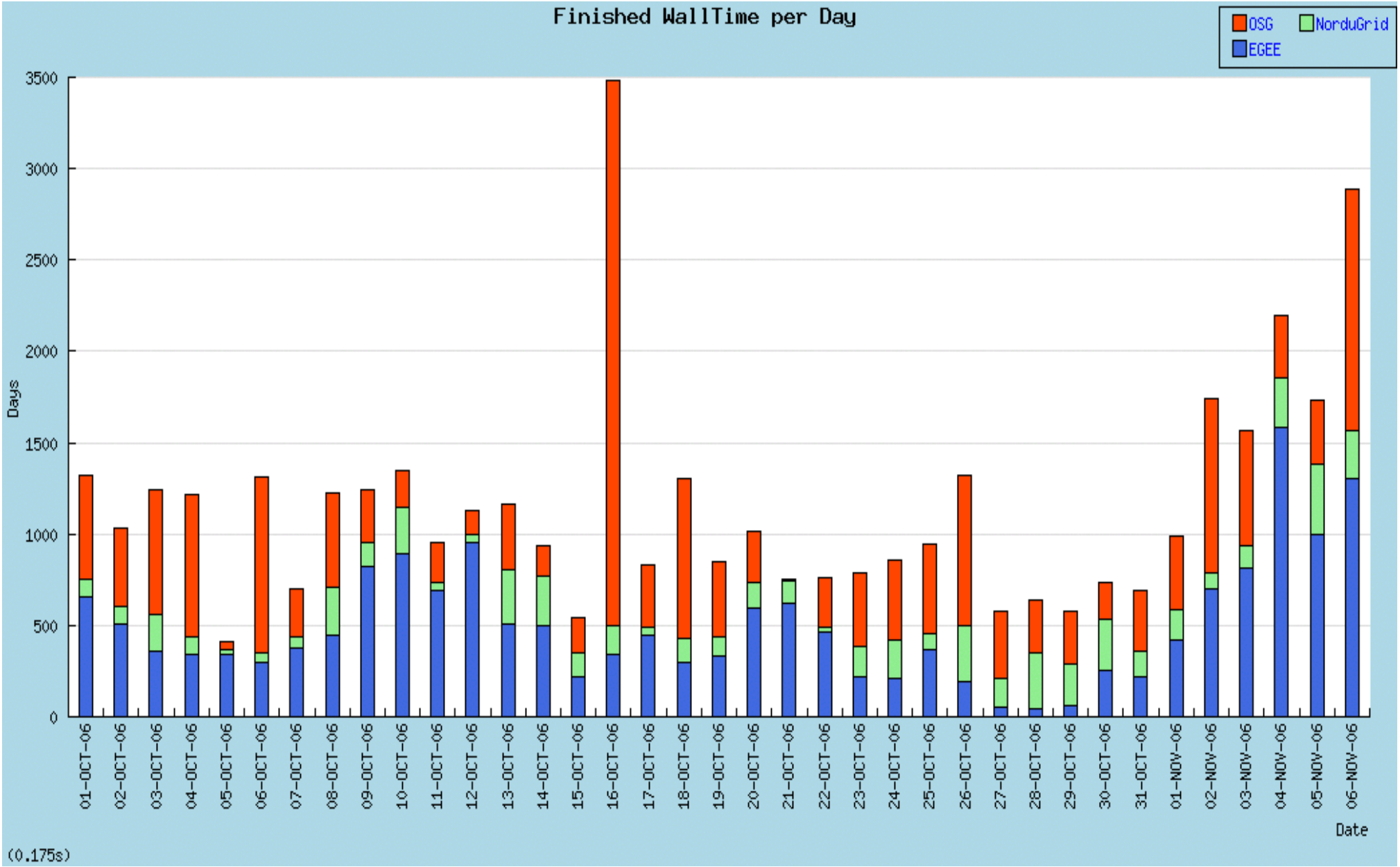
Many jobs with little walltime gave new jobs/day record

NG Jobs/day
Sept/Oct 2006

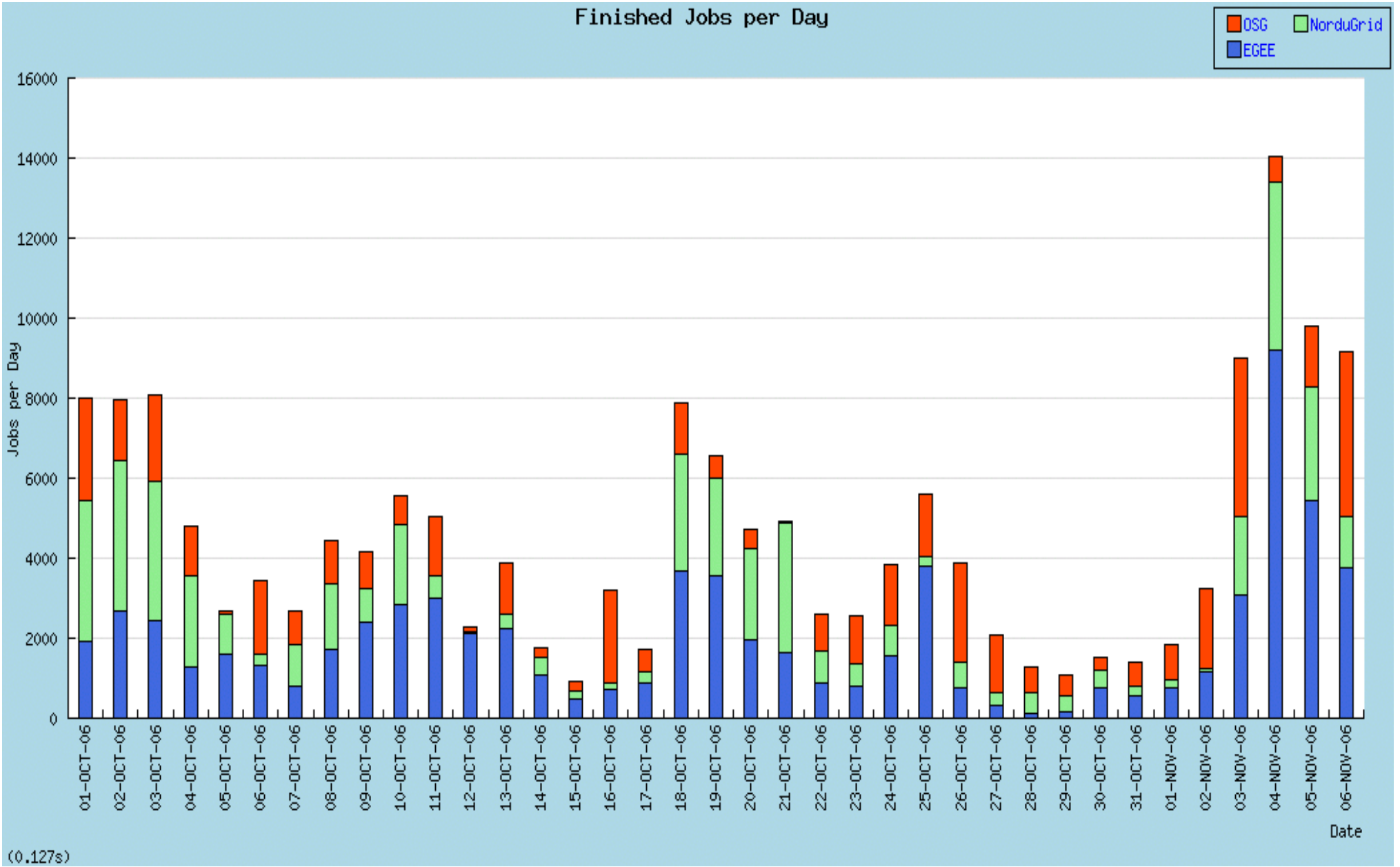
4000



Walltime/day



Jobs/day


Finished Jobs per Day
OSG
NorduGrid
EGEE
Jobs per Day
16000
14000
12000
10000
8000
6000
4000
2000
0
01-OCT-06
02-OCT-06
03-OCT-06
04-OCT-06
05-OCT-06
06-OCT-06
07-OCT-06
08-OCT-06
09-OCT-06
10-OCT-06
11-OCT-06
12-OCT-06
13-OCT-06
14-OCT-06
15-OCT-06
16-OCT-06
17-OCT-06
18-OCT-06
19-OCT-06
20-OCT-06
21-OCT-06
22-OCT-06
23-OCT-06
24-OCT-06
25-OCT-06
26-OCT-06
27-OCT-06
28-OCT-06
29-OCT-06
30-OCT-06
31-OCT-06
01-NOV-06
02-NOV-06
03-NOV-06
04-NOV-06
05-NOV-06
06-NOV-06
Date
(0.127s)

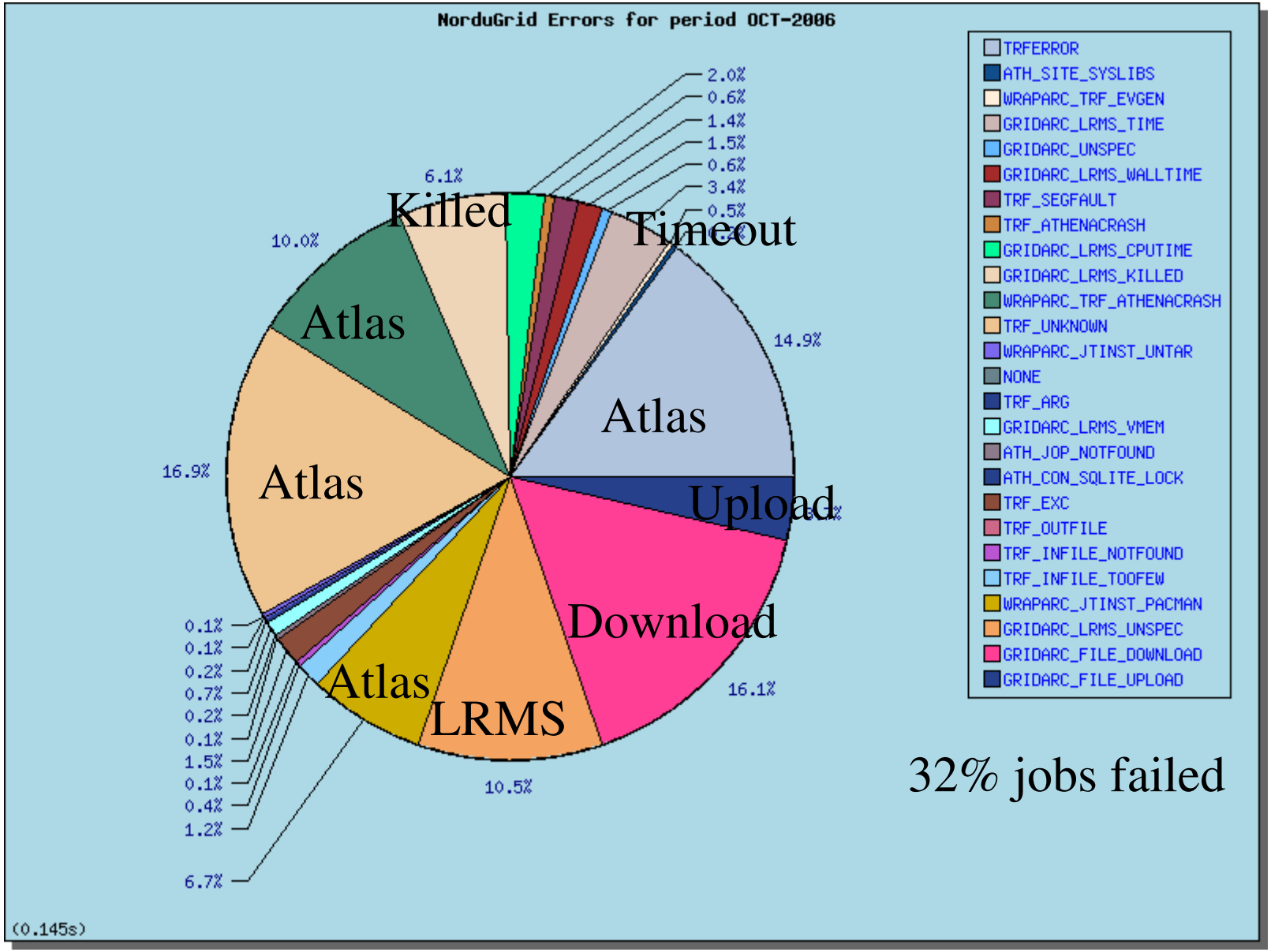
Job throughput

- Short job throughput currently limited by job management
- Downloading 2 small xml files
  - minimum 1.5 s per file
  - 4-5 s per file not unusual
  - can take 1-2 minutes on slow/loaded clusters
- Checking status of job
  - a few times during lifetime of job
  - huge variation
    - 0.01 s/job typical for close clusters
    - 0.3 s/job typical for distant clusters
    - 5-10 s not unusual for slow/loaded clusters
- Setting gacl
  - 0.5-1.0 s typical
  - not necessary if all writable SE's configured correctly!
- DDM registration 3-4 seconds per job (typically 3 files/job)

Job throughput

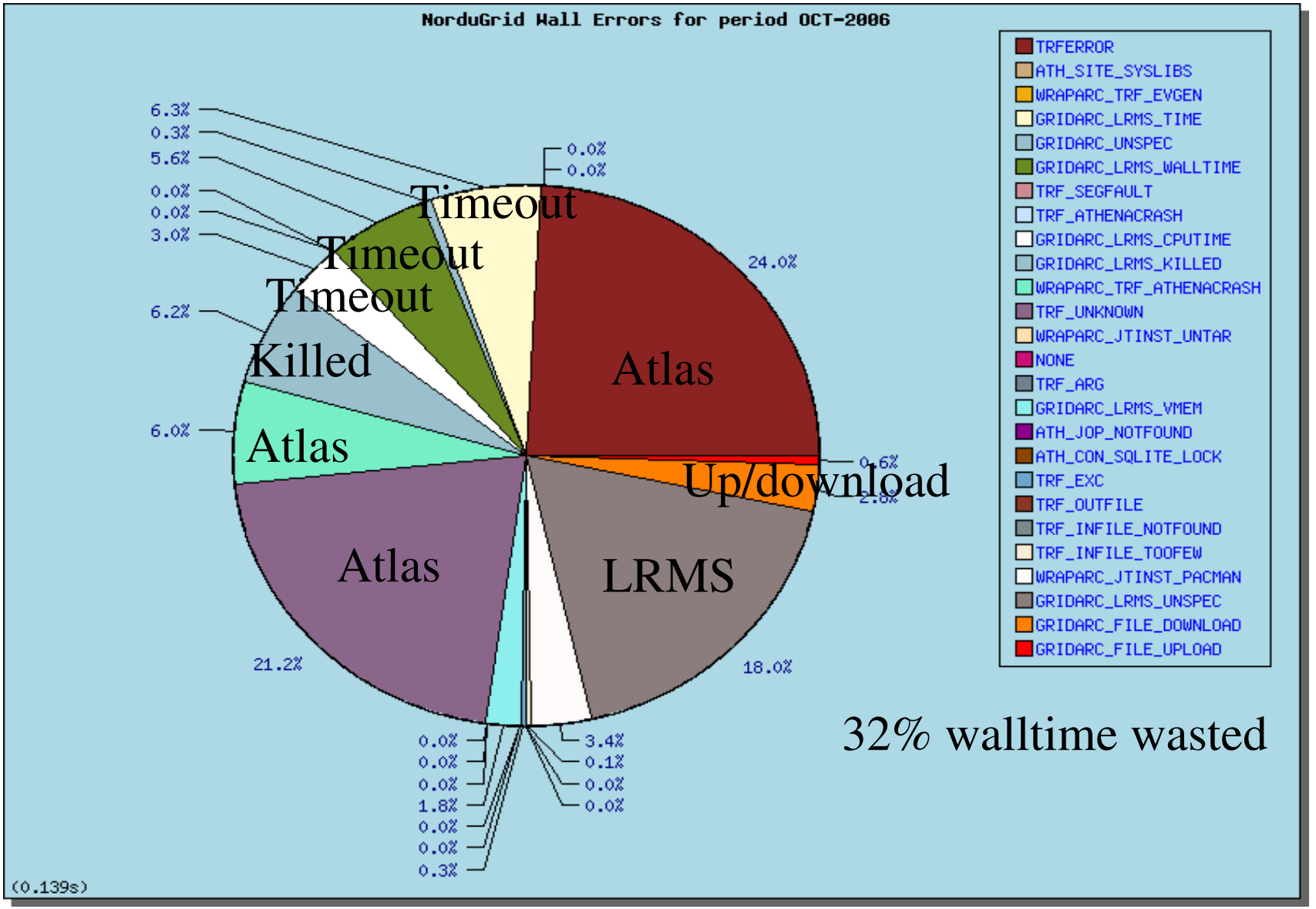
- Job submission
  - varies a lot due to cleanup on retry, number of input files, cluster response, timeout on deadbeat clusters during (common) brokering
  - 3-60 seconds/job, ~2 s for a typical good cycle
- Job cleanup
  - typically ~1 s for FINISHED jobs
    - but slow/loaded clusters use 5-10 seconds
  - O(1 minute) for FAILED jobs
    - gm-files
    - Atlas logfiles
- Eowyn
  - A couple of seconds per job?
  - Some deadtime due to cycling, but probably less than 10% when job activity is high
- Record for single executor during a 24h day is ~3600 jobs, i.e. 24 seconds per job

Job Errors – Oct. 2006



32% jobs failed

Walltime Errors – Oct. 2006



32% walltime wasted

Current challenges

- Many Atlas errors are in fact warnings (according to developers)
  - many unnecessary failures
- FATAL errors do not stop processing – some CPU wasted
- Need to further reduce time between release announcement and production start
- Throughput of short jobs (< 1hr.) should be increased (24 s/job => 3600 jobs/day)
- The agreed (MoU) CPU and storage resources need to be made available (Norway, Denmark working on it)
- Need to reduce ARClib hangups and seg.faults by another order of magnitude – ideally the executor should run unmanaged 24/7
- Clusters and SE's need to upgrade to 0.5.56 or better (job resuming, stability, max connections)
- Job management/communication on slow clusters must be sped up

Current challenges

- Cluster/Queue config and information needs improvement on some clusters to avoid black holes, “manual grid”, invisible log files
- Clusters need better monitoring of disk space (/var, /tmp, session)
- DDM integration
  - stopgap SRM-less solution provided (LRC) to get NG data to LCG+OSG
    - works to Brookhaven, to CERN under test
    - Single srm endpoint requested (FTS “channels”)
      - otherwise compete on open “star” channel, ~unmanaged
  - we get the necessary OSG+LCG datasets into NG by hand, not DDM
    - components on the way...
      - “VObox” - DQ2 site service (Atlas - Oslo)
      - FTS (NDGF)
      - Single (srm) endpoint (NDGF)

Current challenges

- Management/monitoring of CA certificates at the clusters needs to be improved
  - plenty of jobs failures due to new ca certificates not being installed on the clusters
- The logging service is practically impossible to use
  - the simplest query brings grid.uio.no to its knees for many minute
  - For the moment we use only Atlas tools for monitoring.
- Documentation

Conclusions

- The Atlas production is still a great testbed for ARC
- Very hard to prove but it could seem that NG gets more out of its Atlas resources than OSG, LCG/EGEE
- Important progress during the last year, but there is still room for improvement
  - cluster configuration and performance
  - resources
  - Atlas software and job definitions
  - SW and RTE installation procedures
  - middleware stability and throughput
  - data management
  - logging/accounting services