Multi-core job submission and grid resource scheduling for ATLAS AthenaMP

International Conference on Computing in High Energy and Nuclear Physics 2012

D. Crooks, P. Calafiura, R. Harrington, M. Jha, T. Maeno, S. Purdie, H. Severini, S. Skipsey, V. Tsulaia, R. Walker, A. Washbrook
on behalf of the ATLAS Collaboration

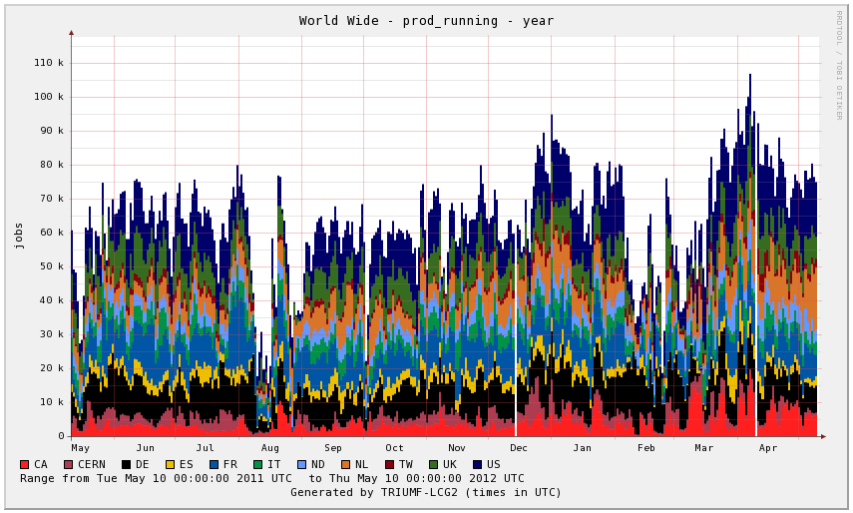
University of Edinburgh

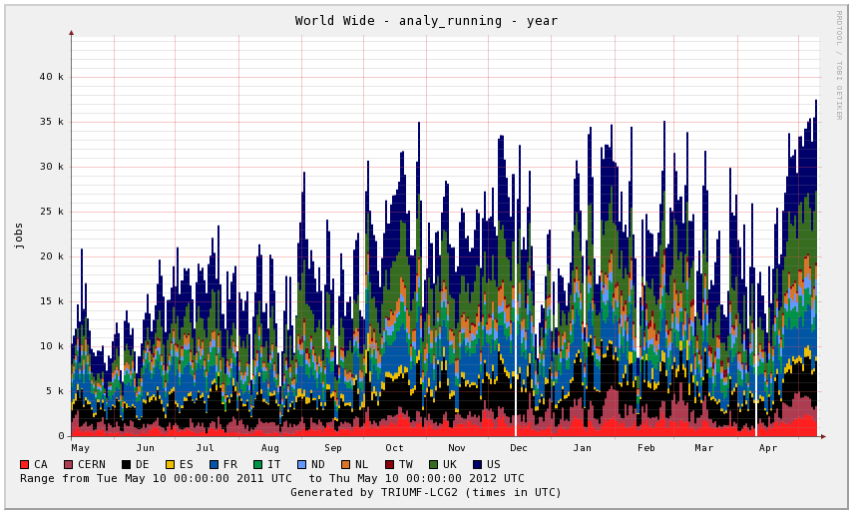
21st May 2012

Outline

Motivation for Multicore

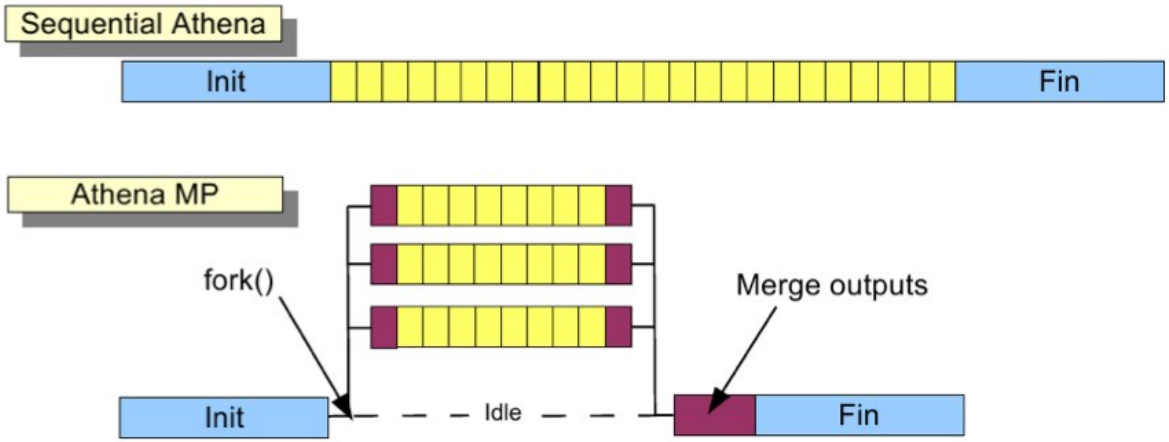
- ATLAS Monte Carlo simulation, data reprocessing and user analysis jobs are run at over 100 computing sites worldwide on a variety of grid infrastructures





- Number of cores has increased on worker nodes $\Longrightarrow$ Allocate one job per core to maximise resources
- Ratio of physical memory to number of cores remained constant

AthenaMP



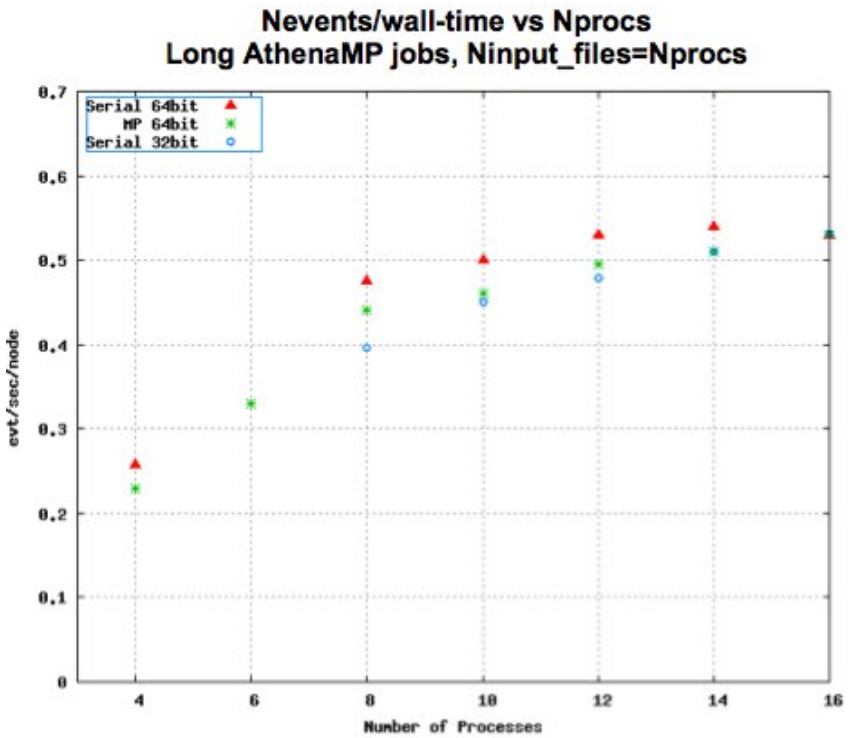
- AthenaMP provides maximum memory sharing between multiple Athena worker processes
- Copy On Write mechanism (CoW) provides an effective memory sharing technique
- Python multiprocessing module provides worker process management
- Event based parallelism retained by running one process per core
- Almost 80% memory sharing can be achieved with negligible CPU overhead

AthenaMP Serialisation

- Main areas of serialisation are job initialisation and file merging
- Timing of process `fork()` crucial to enable maximum amount of memory to be shared

Amdahl's Law

$$S(N) = \frac{1}{(1-P)+\frac{P}{N}}$$



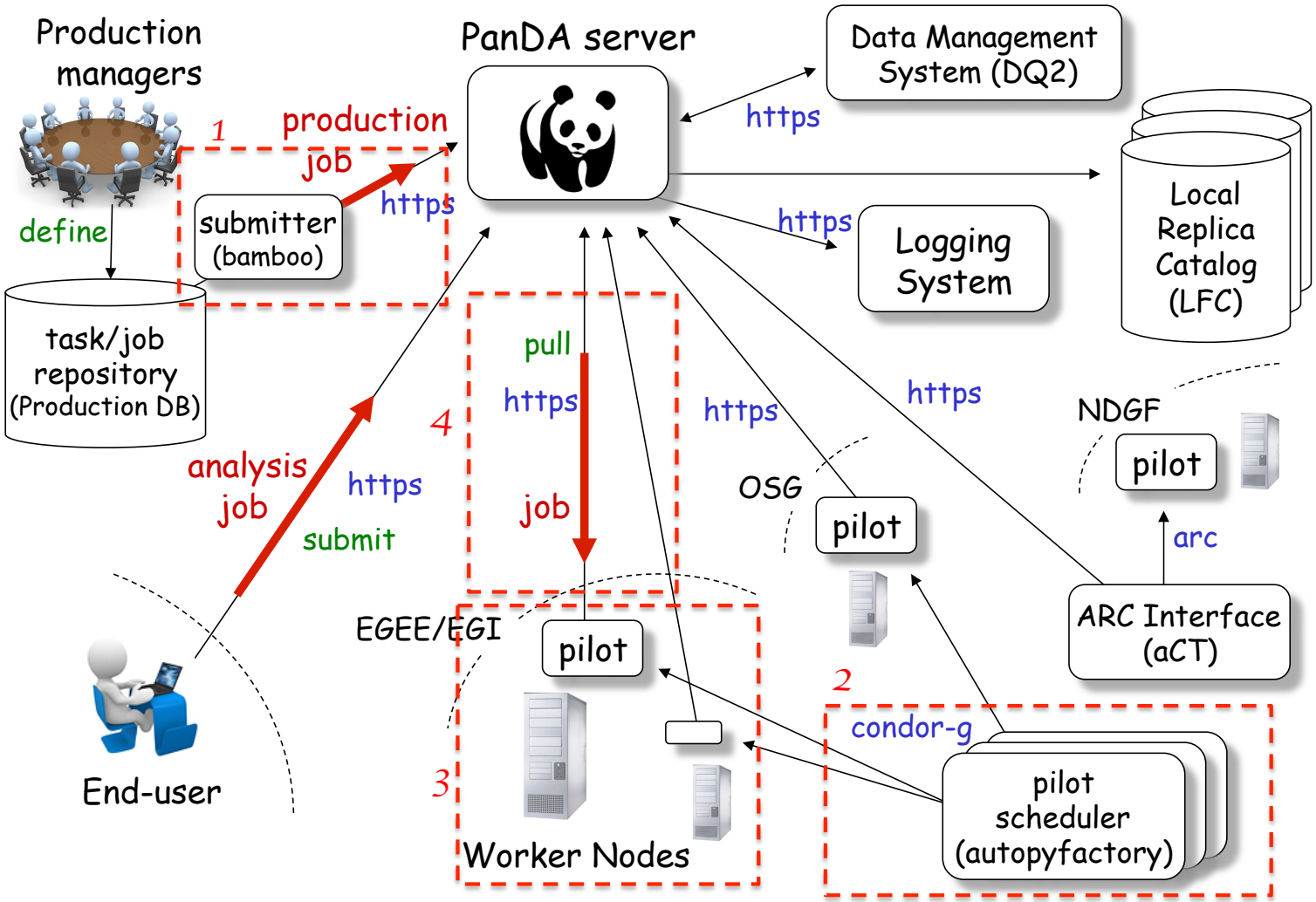
- Increase the parallel section of the job by increasing the length of the event loop
- A common approach is to scale the number of input files with the number of processes

AthenaMP Running Modes

MP Option	Default
EventsBeforeFork	1
doFastMerge	True
doRoundRobin	False
AffinityCPUList	[ ]

- Faster merge algorithm concatenates event data and metadata files rather than full event validation
- Event queue model generally performs better than using fixed allocation of events per worker
- Workers pinned to specific cores may be helpful to mitigate undesirable NUMA effects

ATLAS PanDA System



Multicore in the PanDA system

How can we incorporate multicore jobs into PanDA?

- Difficult to make major changes during data taking operations
- Continue single core job brokerage with increasing number of AthenaMP tasks to new multicore queues

Should multicore jobs reserve all cores on a worker node?

Advantages	Disadvantages
Runtime check of worker node hardware to define number of cores	Lack of scheduler flexibility
All memory and CPU resources dedicated to the AthenaMP job	Large variation in core count

Should a dedicated set of resources be provided to multicore queues?

Advantages	Disadvantages
Fits well with existing submission framework	**Low usage leads to wasted resources**

Dynamic Job Resource Allocation

Can these resources be allocated dynamically?

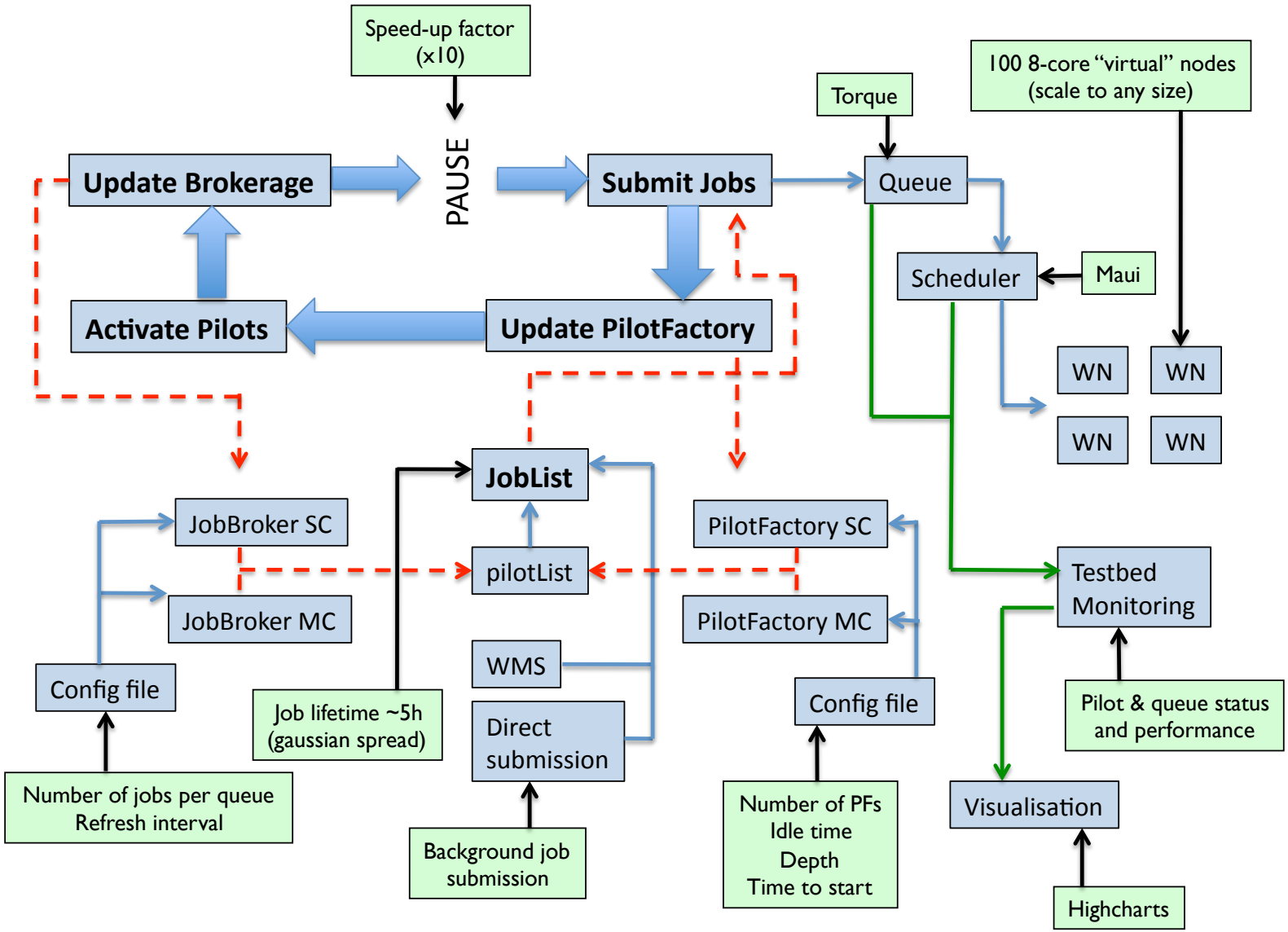
- More flexible option than static allocation
- Jobs with differing resource requirements already handled successfully by leading scheduler implementations

Additional Factors

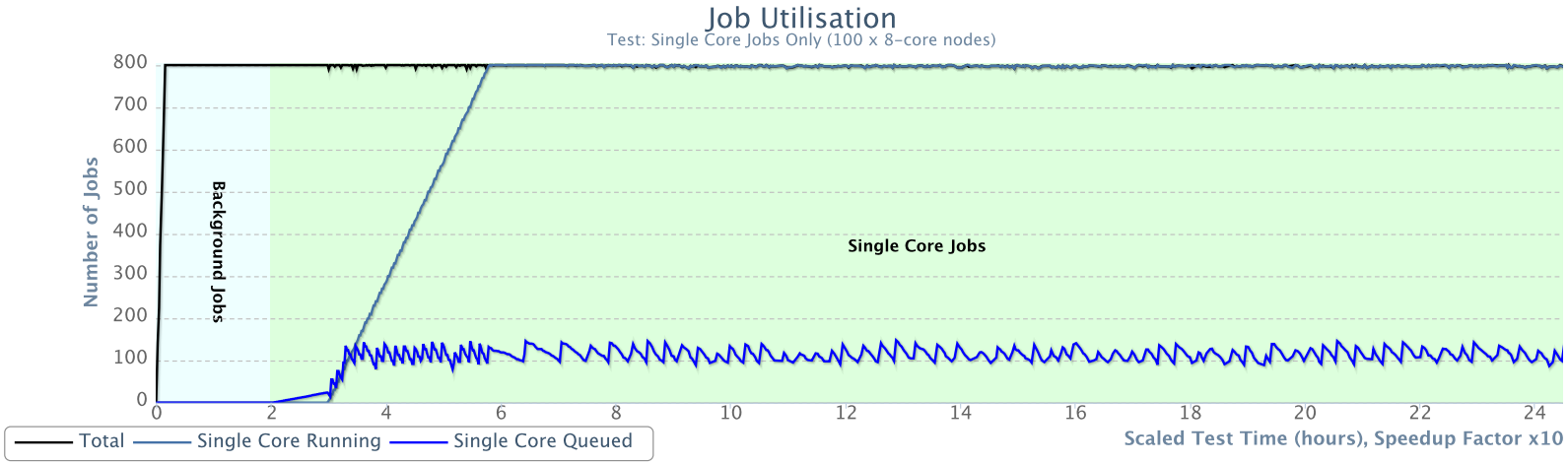
- Job submission rate is dependent on batch system load
- Job lifetime depends on external brokerage which in turn is decided in part by batch system load
- Job queues are (in general) not exclusive to ATLAS

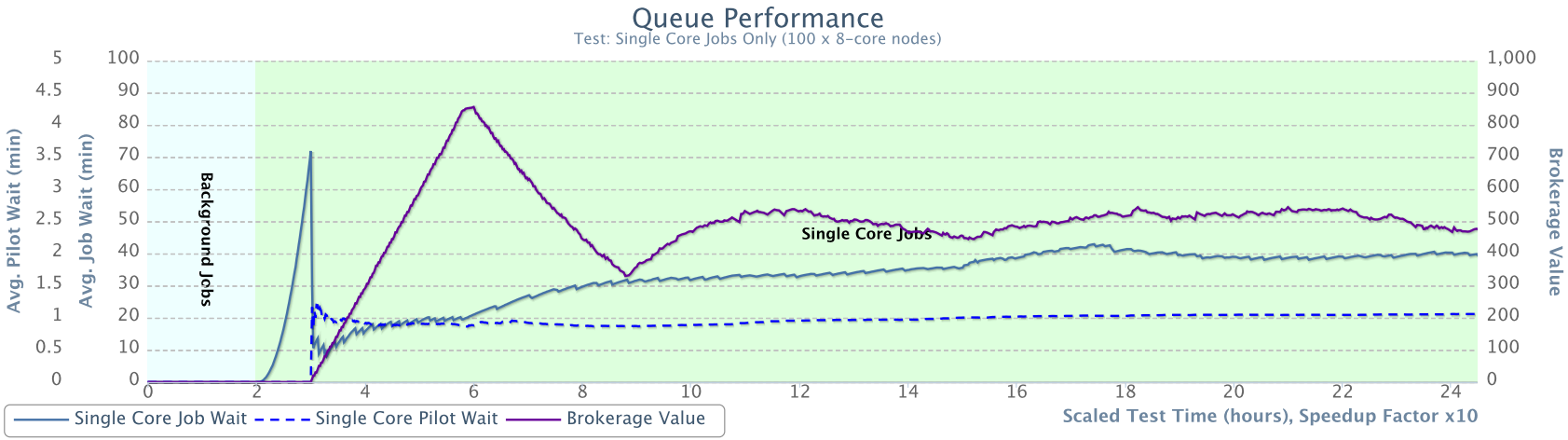
Investigate the best approach to run both single core and multicore jobs on the same underlying resources

Scheduling Simulation

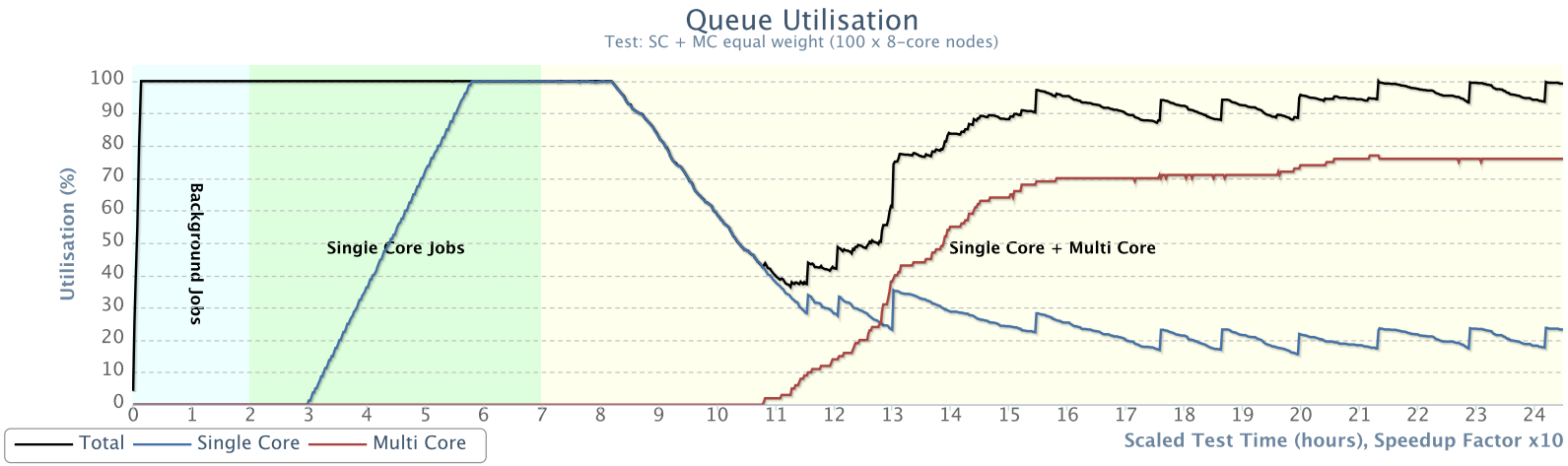


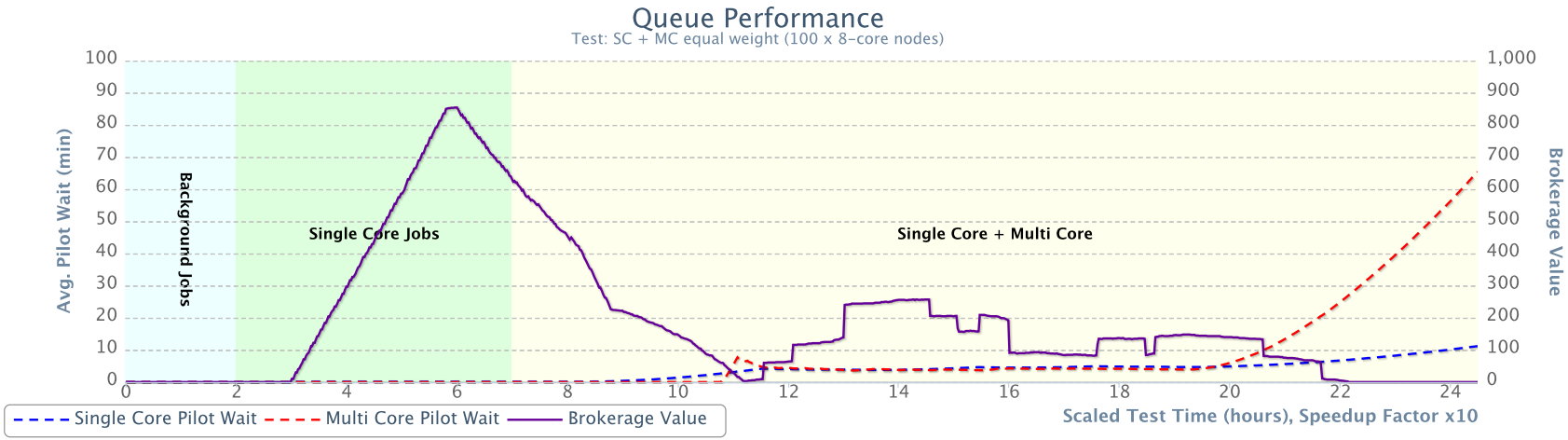
Scenario 1: Single Core Pilots



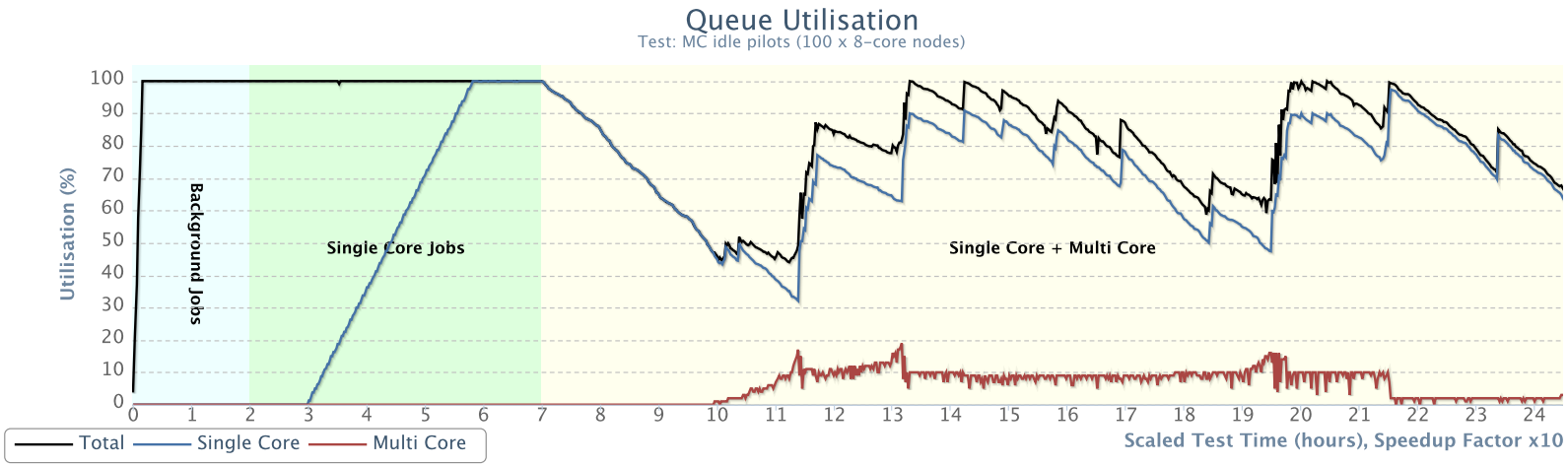


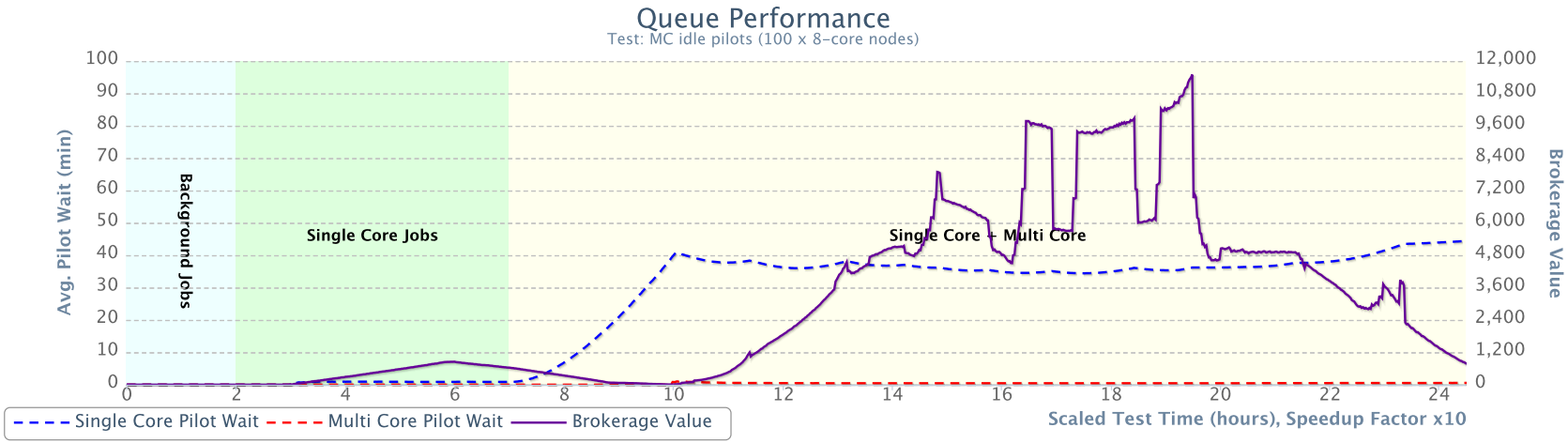
Scenario 2: Single and Multi Core Pilots



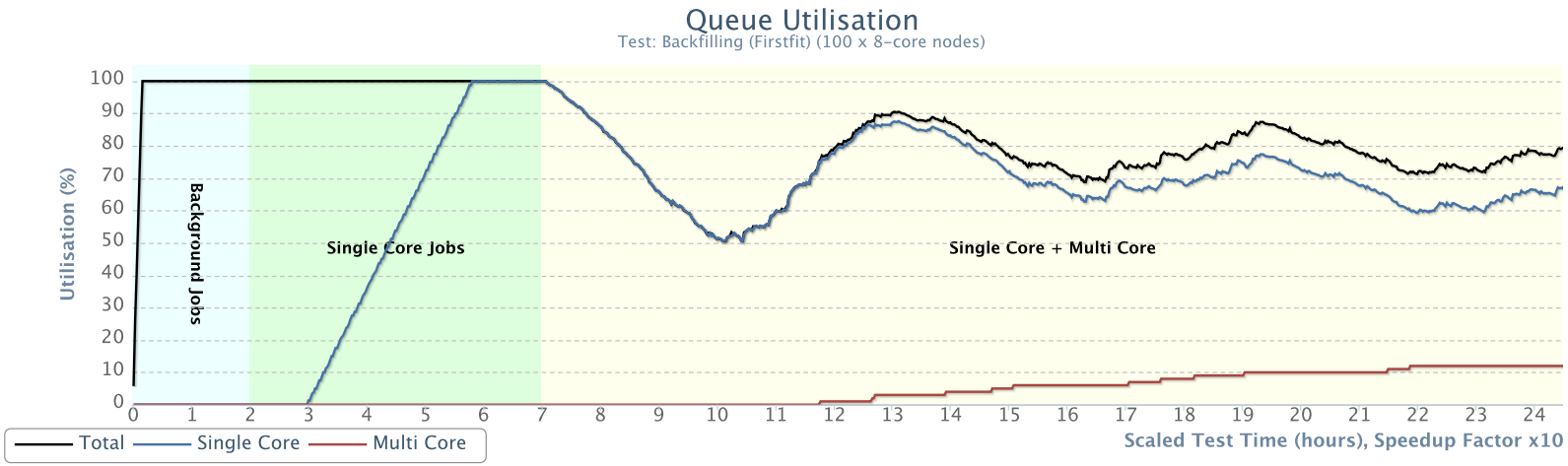


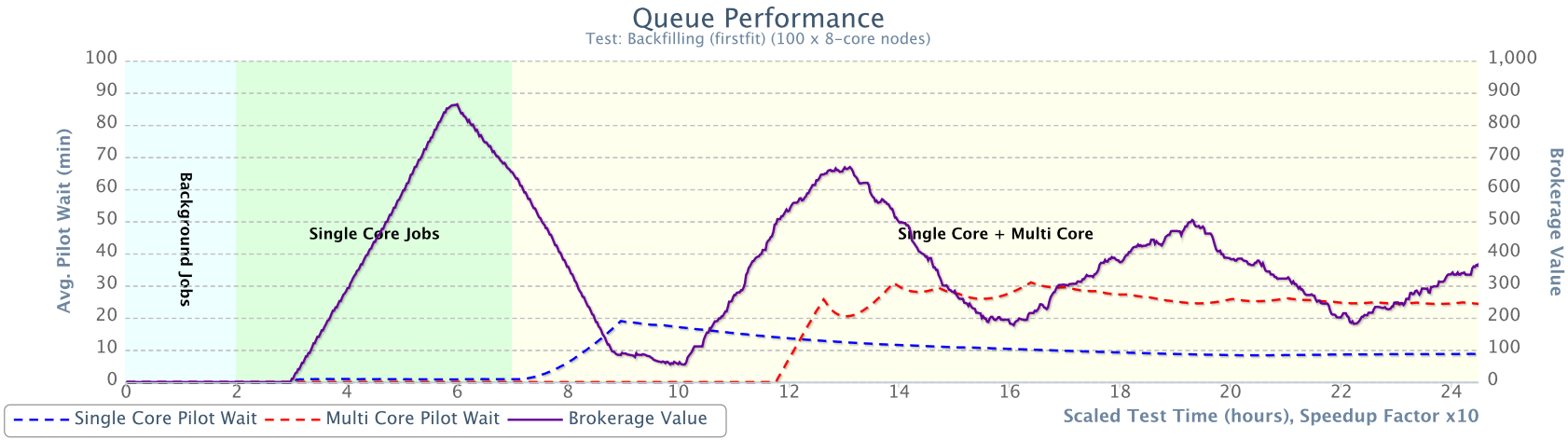
Scenario 3: Idle Multi Core Pilots





Scenario 4: Multi Core pilots with Backfilling





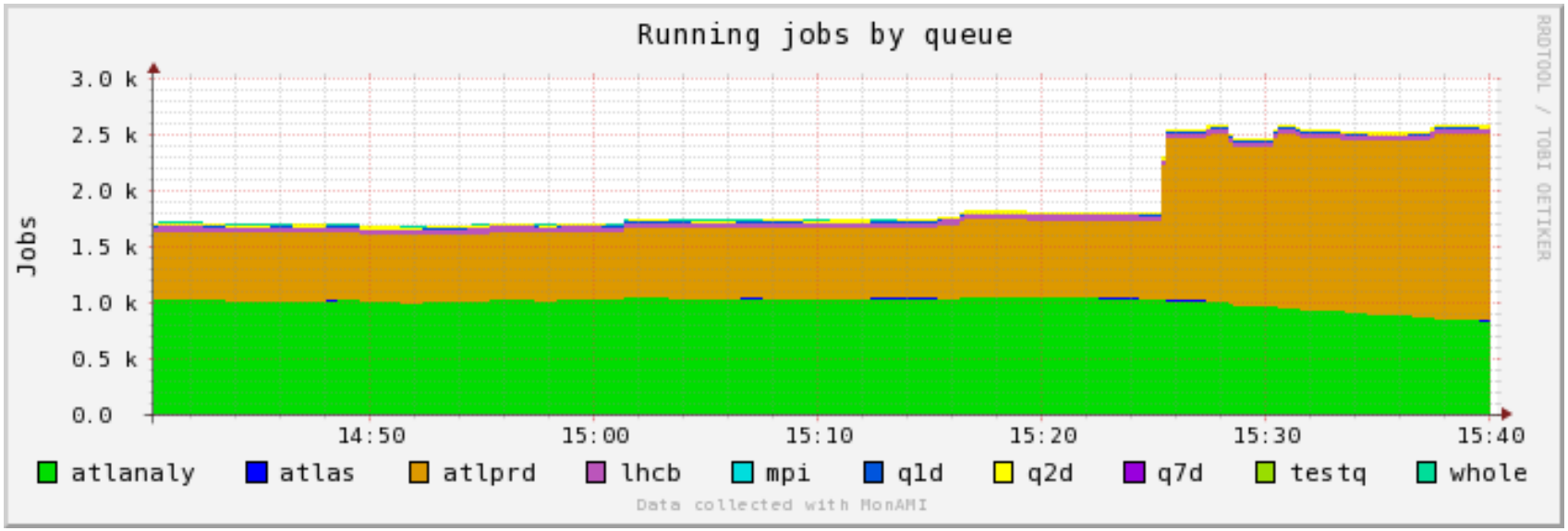
Multicore Production Status

Site	Logical Cores	Cores/ Node	LRMS & Scheduler	Grid Exclusive	Multicore Queue	Pledge (nodes)	LHC Exclusive
BNL (US)	10,611	8/24**	Condor	No	Dedicated	50	Yes
ECDF (UK)	2,896	8/12	SGE	No	Shared	N/A	No
Glasgow (UK)	2,616	8/12/64	Torque/Maui	Yes	Shared	N/A	Yes
INFN-T1 (IT)	9,216	4	LSF	Yes	Dedicated	8	Yes
Lancaster (UK)	2,096	8	Torque/Maui	Yes	Dedicated	8	Yes
OSCER (US)	4,096*	8	LSF	No	Shared	N/A	No
RAL (UK)	2,872	8	Torque/Maui	No	Dedicated	15	Yes

*Under upgrade
**Each multicore worker node divided into two 8 core slots

- A number of grid sites have already pledged multicore resources
- Core count queue parameter determines number of AthenaMP worker processes
- Assume core count = total number of worker node cores?

Multicore Production Issues



- Whole node queue priority boost causes reduction in overall job throughput
- Priority tuning required for each site not partitioning dedicated resources

Middleware Requirements

TEG Multicore Recommendations TEG Wiki page

- WLCG Technical Evolution Group investigated possible strategies for multi-core access using requirements from both experiments and resource providers

Include multi-core request details expressed in the JDL

- number of requested cores
- total memory for the job (or memory per core)
- job requires wholenode
- min/max number of cores (optional)
- min amount of local disk space (optional)

Information System

- Specify the maximum number of cores supported
- Whether site accepts wholenode and/or multicore

Summary

AthenaMP motivation

- ATLAS has created a multicore implementation of their software framework to reduce the memory footprint of pileup reconstruction jobs

AthenaMP in Production

- Increasing number of sites are delivering resources for multicore use
- Most opted for small amount of dedicated wholenode resources
- Sites need consistent job flow to maximise resources

Middleware and Scheduling

- Efficient job scheduling for single core and multicore pilots needs to be addressed
- Middleware could be extended to include additional multicore specifications