Future of Batch Processing at CERN

HEPiX Spring 2014

Belleman Jérôme – Pék János Dániel
– Schwickerath Ulrich
CERN IT

May 2014

Status report

Outline

1. Reminder from last year
2. More HTCondor results
3. Potential integration of HTCondor

Section 1

Reminder from last year

http://cern.ch/go/Nnj8

Goals and concerns

- CERN currently uses IBM LSF 7.0.6

Goals	Concerns with LSF
30 000 to 50 000 nodes	6 500 nodes max
Cluster dynamism	Adding/Removing nodes requires reconfiguration
10 to 100 Hz dispatch rate	Transient dispatch problems
100 Hz query scaling	Slow query/submission response times

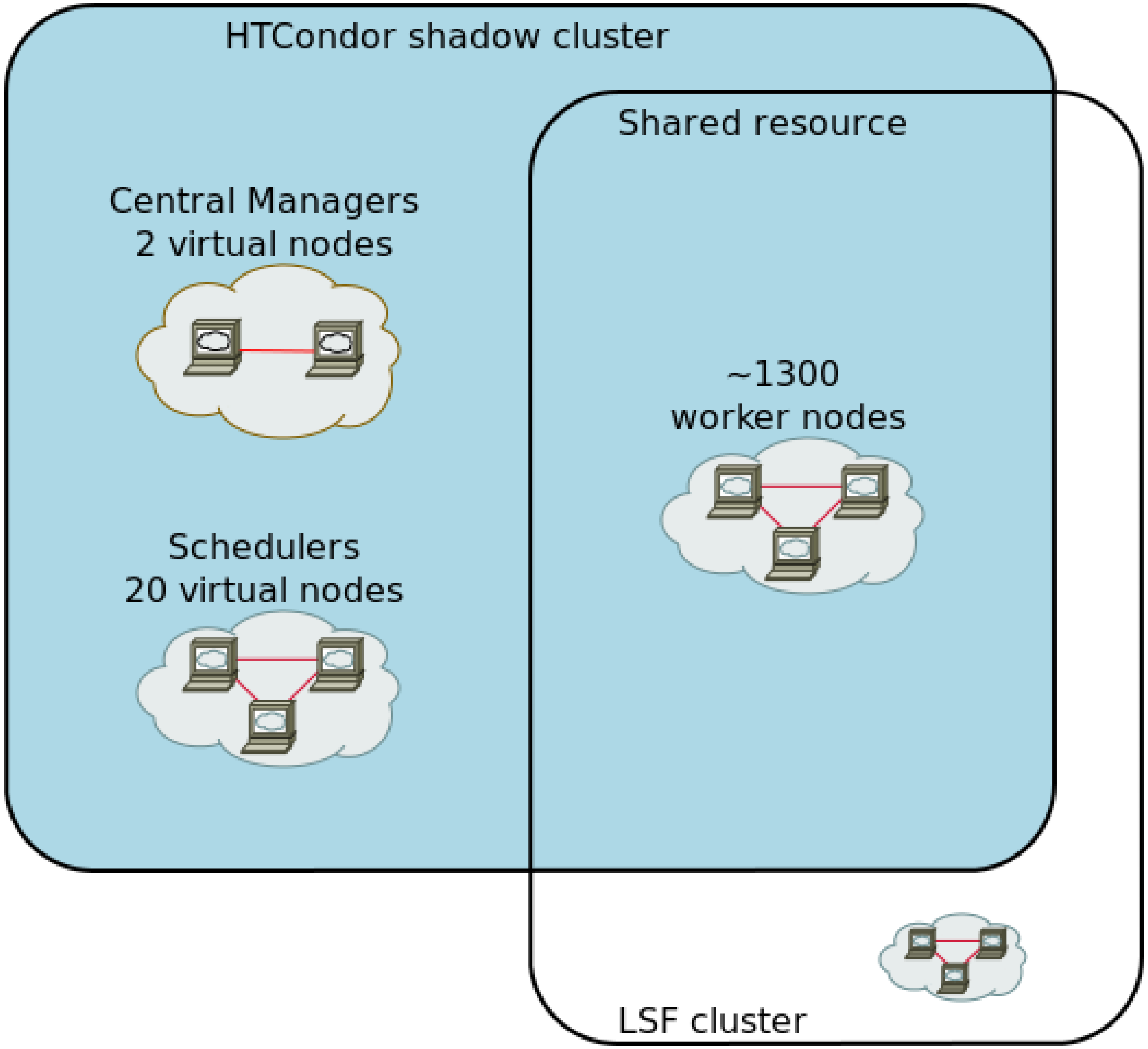
Alternatives

- LSF 8/9 – it is not said to scale much higher than LSF 7
- SLURM 2.6.4 – concerns on scalability
- Son of GridEngine 8.1.6 – slightly tested
- HTCondor 8.1.5 – seems promising

Section 2

More HTCondor results

Testbed architecture (1)



Testbed architecture (2)

- 2 Central Managers
  - VM: 4 cores, 8 GB RAM
  - 1 negotiator, 50 collector instances
  - 1 gangliad, 50 + 1 (main) collector instances
- 20 Schedulers + submission nodes
  - VM: 4 cores, 8 GB RAM
- ~1300 Machines (worker nodes)
  - VM and physical
  - 48 slots forced by configuration → 62 500 slots

General experience (1)

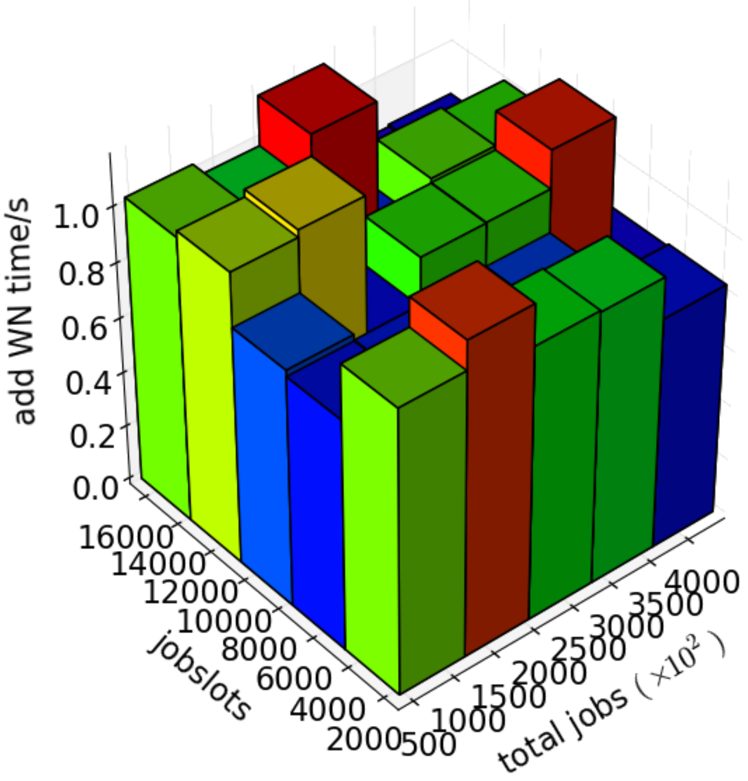
- Configuration
  - + Fine-grained control over almost everything
    - + Macros: e.g. calculate queue size based on memory
  - + Nicely structured, and documented
  - – Sometimes not that intuitive
    - – MAXJOBRETIREMENTTIME for disable eviction
  - – “Abundance of choices”

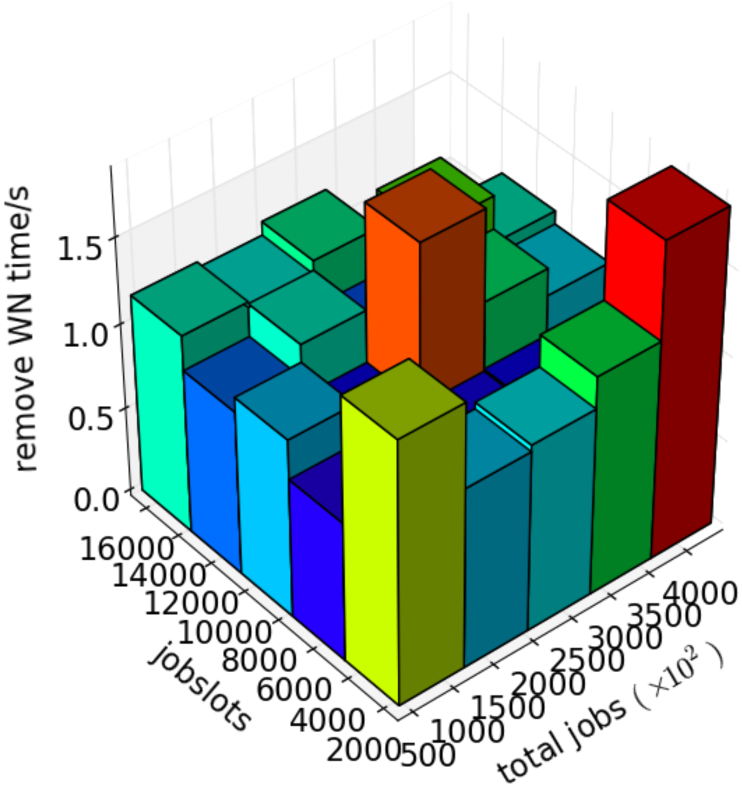
General experience (2)

- Automation, Puppetisation
  - + Self-registering decoupled components
  - + Python API
    - + Automate everyday operational tasks
    - + e. g. waiting until all job slots are claimed
  - + Plenty of useful user-space tools
    - + condor_status, condor_q, condor_on/off, condor_advertise, condor_submit, condor_rm, …
    - + condor_sos: “prefix” for emergency operations

General experience (3)

- Flexibility
  - + No need of restart daemons almost ever
  - + Easy and fast to add/remove worker nodes



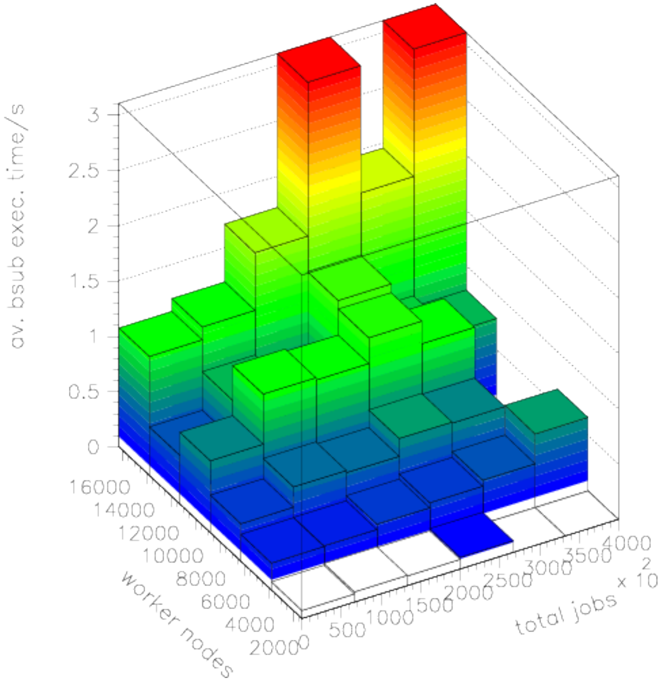


HTCondor addition and removal of WNs

PES

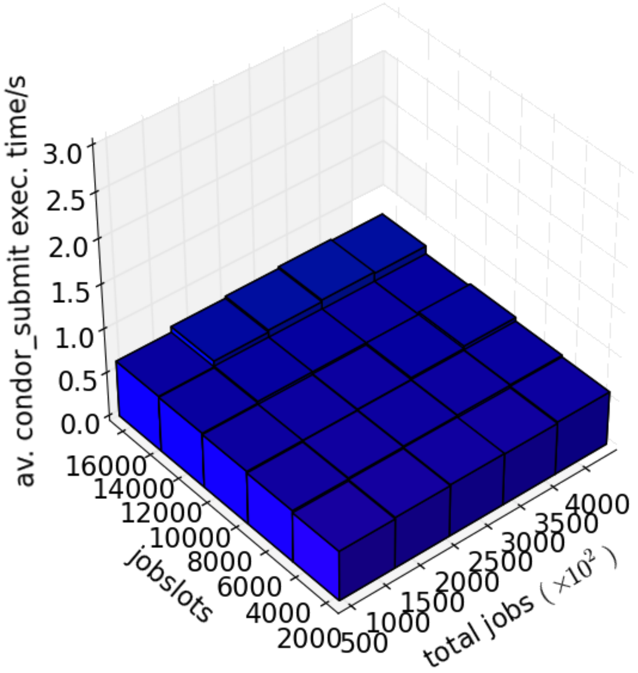
General experience (4)

- Scalability
  - + Scales well horizontally in
    - + Number of job slots and nodes
    - + Number of jobs
    - + Submission rate and delay



Submission on LSF

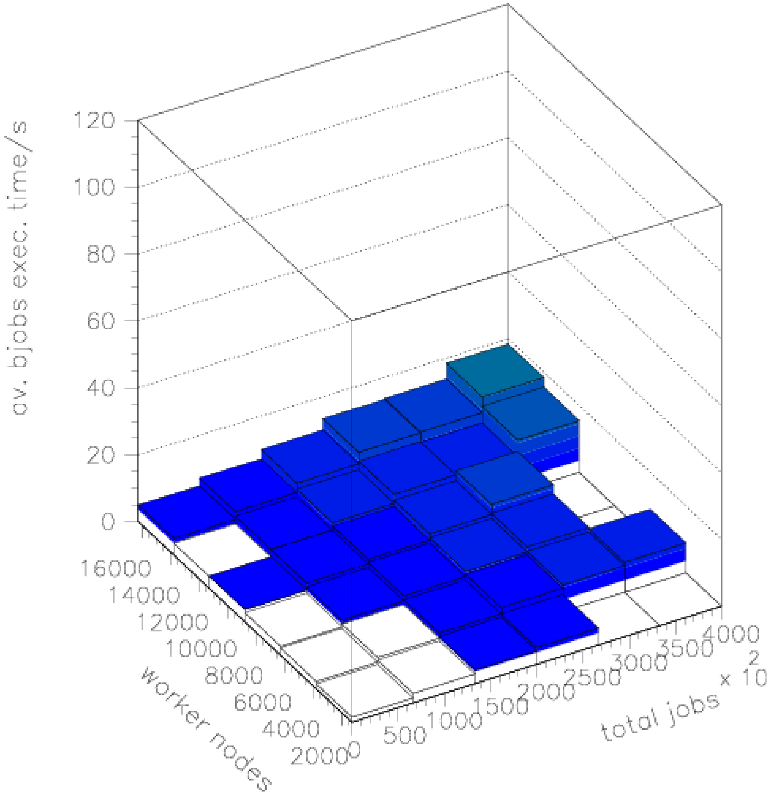
Submission on Condor



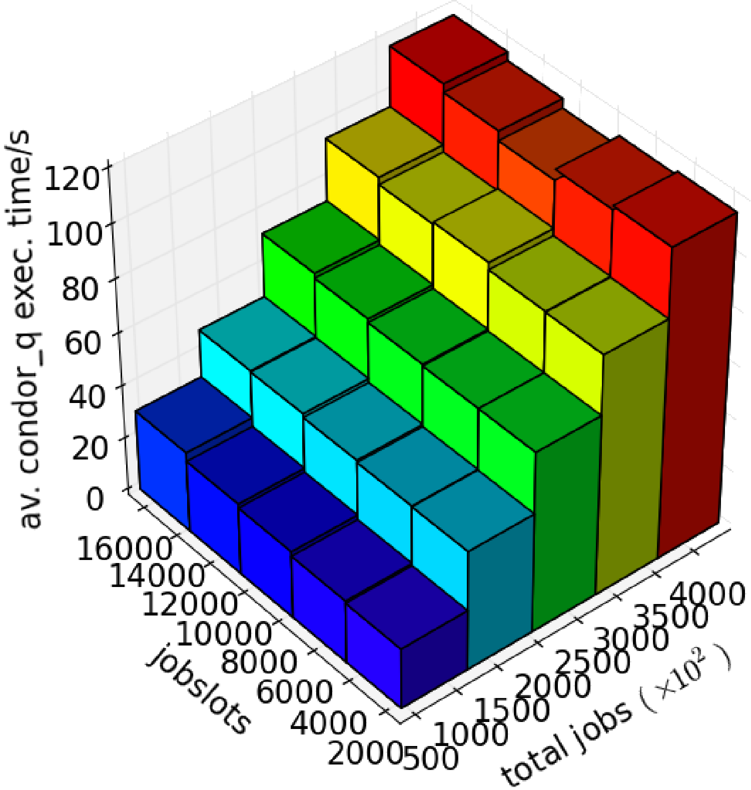
General experience (5)

- Scalability
  - Schedd and shadowd are memory-eager
  - Scales poorly in query rate

Query on Condor



Query on LSF



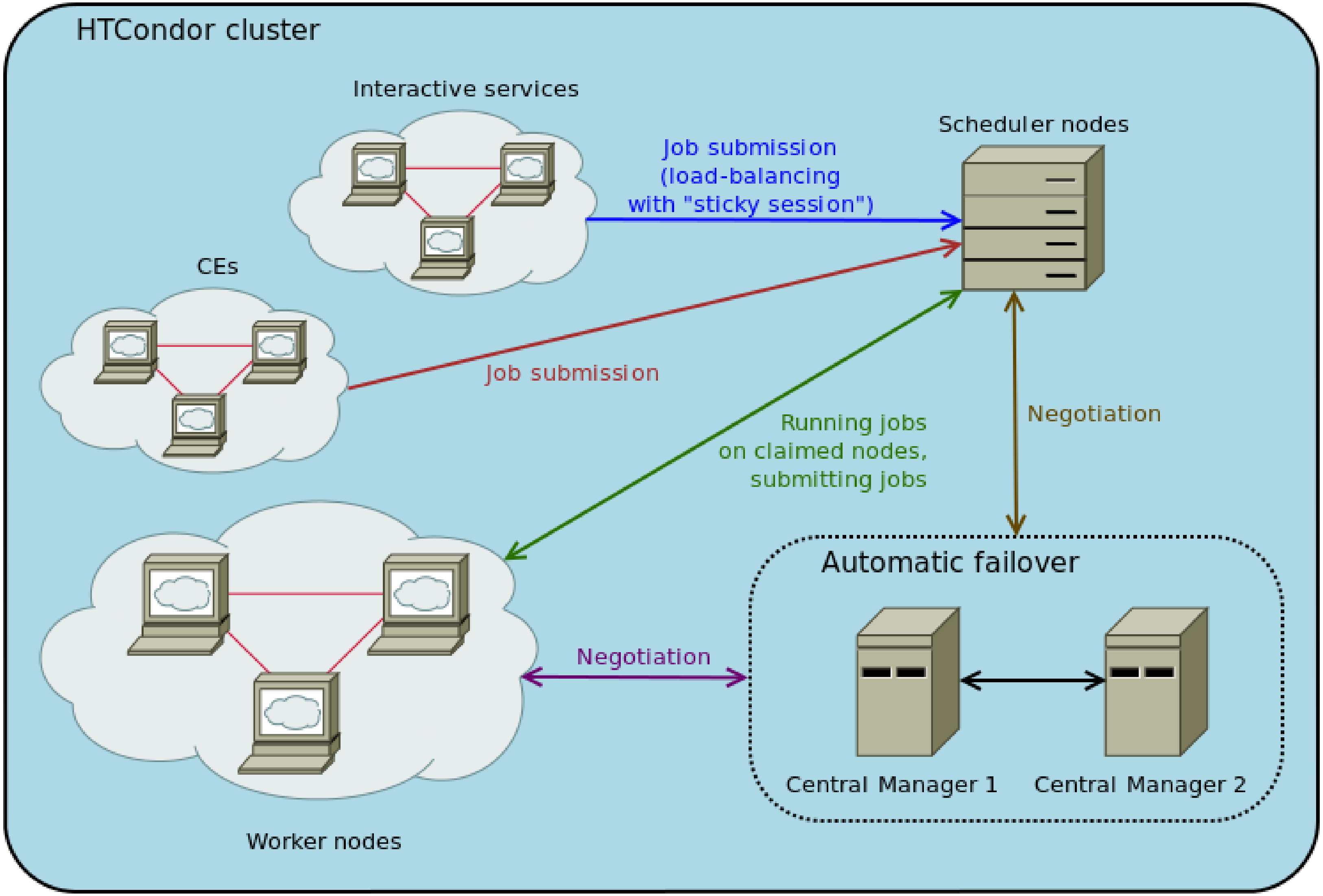
General experience (6)

- Fault-tolerance
  - + Automatic fail-over works fast
  - + No single-point-of-failure
  - + Designed for heterogeneous infrastructures
- Maturity, community
  - + Feels robust and mature
  - + Very active community
  - + Frequent development releases
  - + We're in touch with the HTCondor project lead

Section 3

Potential integration of HTCondor

Production architecture



Functional requirements

- To be implemented
  - Kerberos/AFS authentication support
- To be tested
  - Accounting
  - Host normalisation
  - Fairshare

Conclusion

- Scaling tests are reaching a conclusion
  - Host scalability tests carried out
  - Query load tests carried out
  - HTCondor is a strong candidate
- What's next
  - Integration
  - Pilot project

Thanks!

Questions?