Filesystems efficiency at CC-IN2P3

Yannick Perret

Preamble

- Hardware evaluation :
  - Several benchmarks exist
  - LHC uses SPEC (with particular options)

→ SPEC-HEP06

- Limits :
  - Only CPU (and memory architecture)
  - No disk
  - No network
  - No interaction between these elements

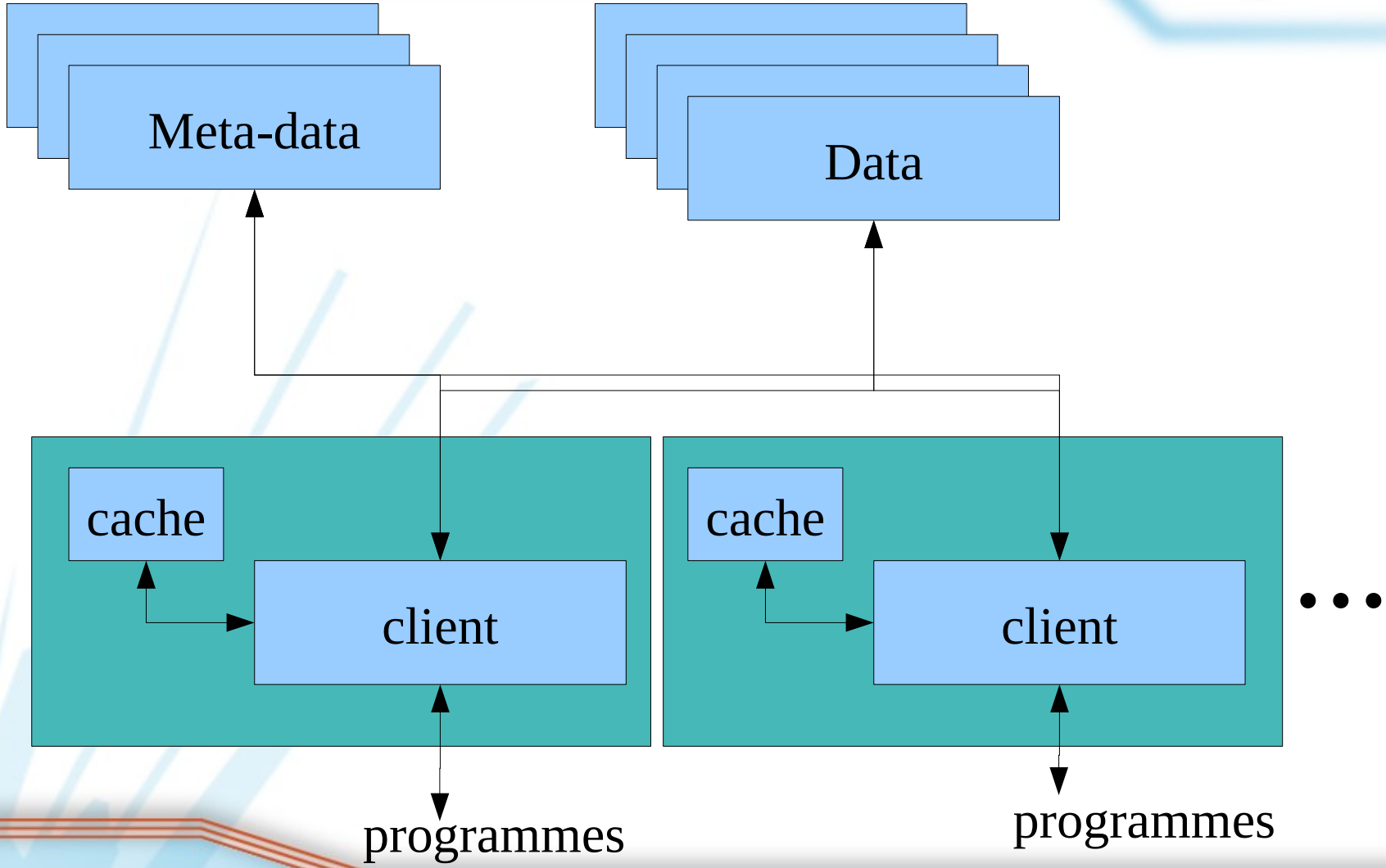
Context

- Latency/duration problems on (some) jobs:
  - « random » problems
  - No clear error
  - No technical problem

- Diagnostics are difficult :
  - Difficullt to reproduce
  - Mix of several problems

→ Points to performance problems on filesystems

Filesystems (FS)



Performance limitations

- Server side:
  - Meta-data access
  - Data access
  - Bandwidth (network, disks, …)

- Client side:
  - (servers)
  - Local I/O limits (network, disks, bus…)
  - Parameters (cache size, configuration…)

« New » : algorithm complexity of client treatments

Improve performances

- Servers side: + = better!
  - More (meta-)data servers
  - More powerful servers (CPU, network, disk…)
  - More network (bandwitdh, latency)

- Clients side: more difficult
  - More powerful nodes (CPU, network, disk…)
  - Adapt configuration to real use
  - Increase cache size, priority…

→ Side effects difficult to predict!

Side effects (clients)

Various side effects:

- No universal parameters (depend on usage)
- More powerful machines ⇒ faster jobs (more access to FS)
- Larger cache ⇒ more treatments on clients
- Cost: servers → few machines ; clients → +1000 machines

And:

No FS access is linear with the number of concurrent access!

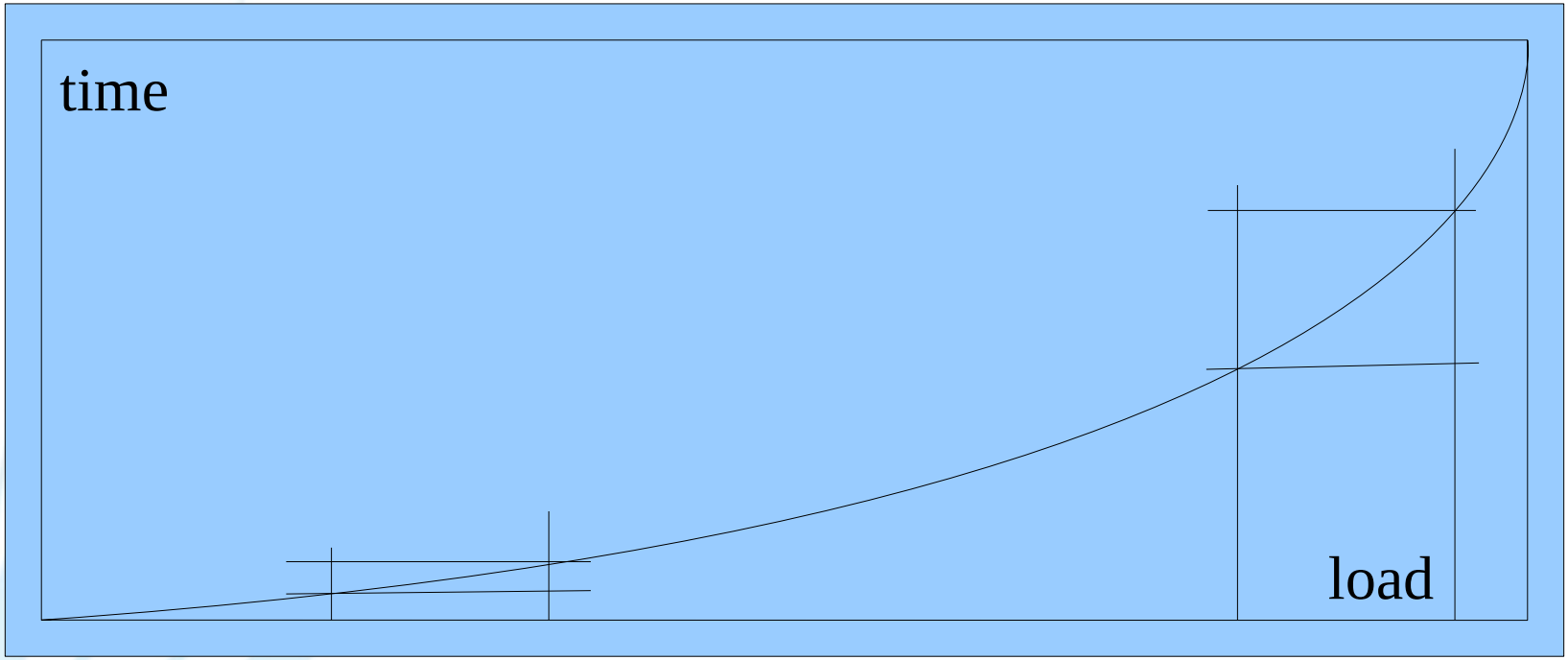
Computing nodes evolution

- In the past:
  - Only single-core processors
  - More sockets, more powerful
- Limit reached → adding cores:
  - Hyperthreading
  - Dual, quad, hexa-cores
- Today and near future:
  - Octo-cores + hyperthreading
  - quad/octo-quad/octo sockets
  - ...

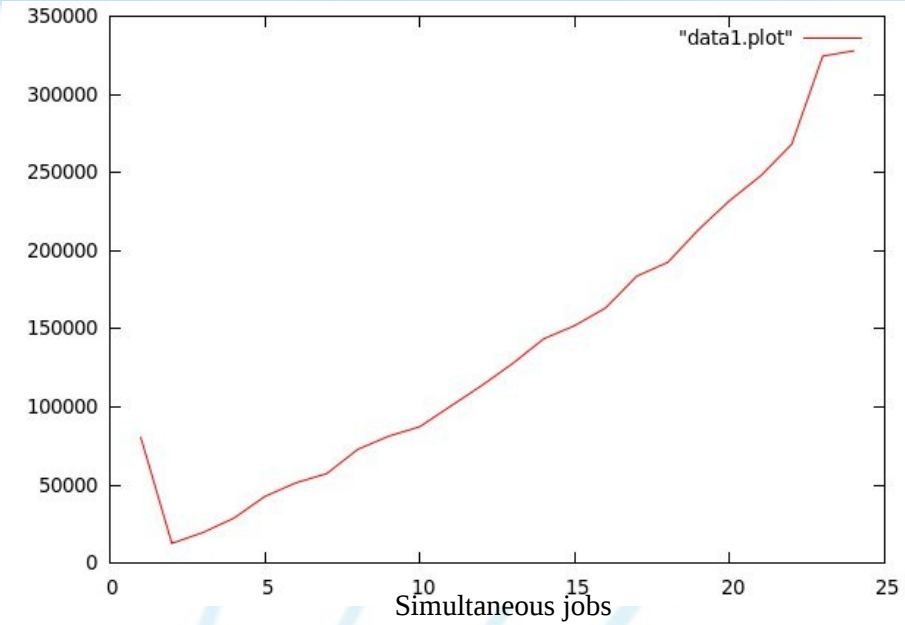
Impact

More jobs on a single machine.

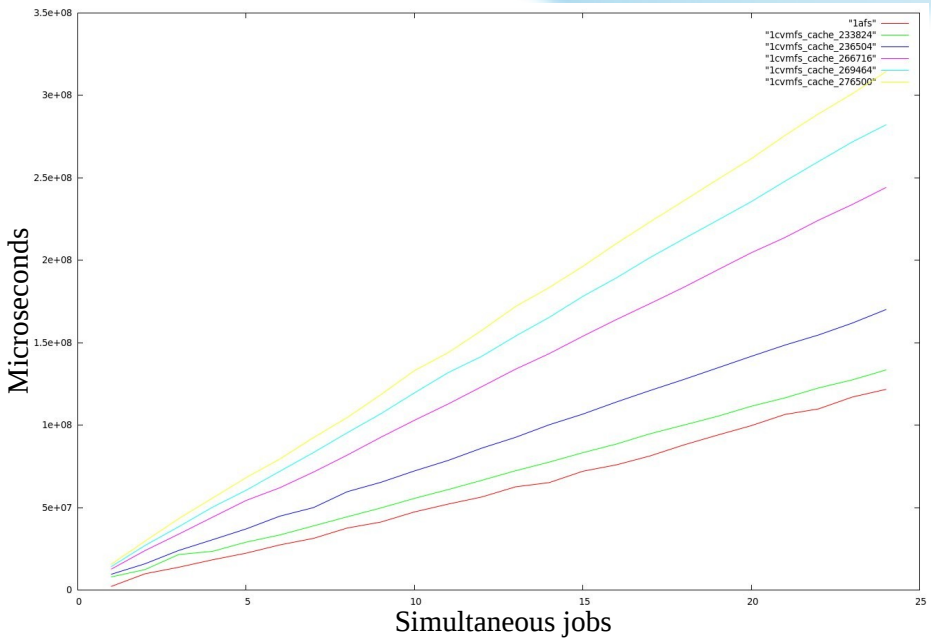
A small increase in load implies a sensible increase in time!



Examples



Non linearity (AFS client)



Bad cache management

Same behavior for AFS, NFS, GPFS, CVMFS, …

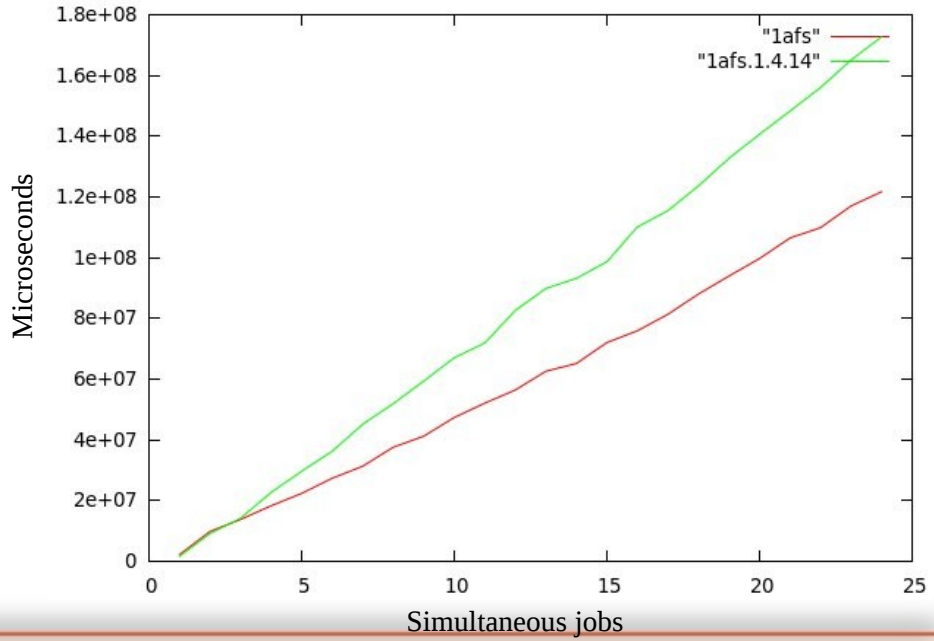
Solutions

Depend on several points:

- Type of service
- Average/max load
- Consequences of changes (cost, compatibility…)

Sometimes a « simple » update is enough

→ whatever, implies several tests to validate…



Solutions to explore

- Improve users code (possible?)
- Dispatch access on several FS
- Use local FS
- Dedicate services/hosts to particular uses
- Virtualization
- Better management of jobs (type de ressources…)
- Improve hardware

Analyses to come

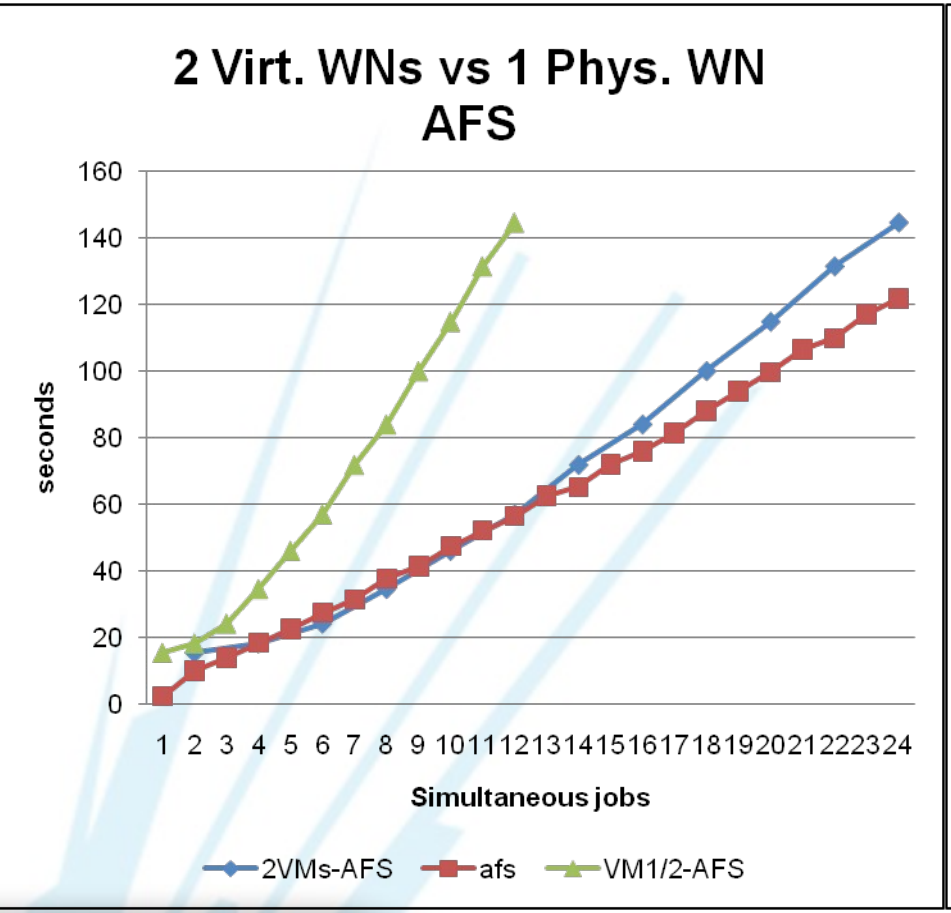
- Characterise behaviors for larger systems → +48 cores
- Compare differences beetween manufacturers:
  - Hyperthreading
  - NUMA
- Test different approaches:
  - (nearly)local storage
  - Dedicated storage network
  - …

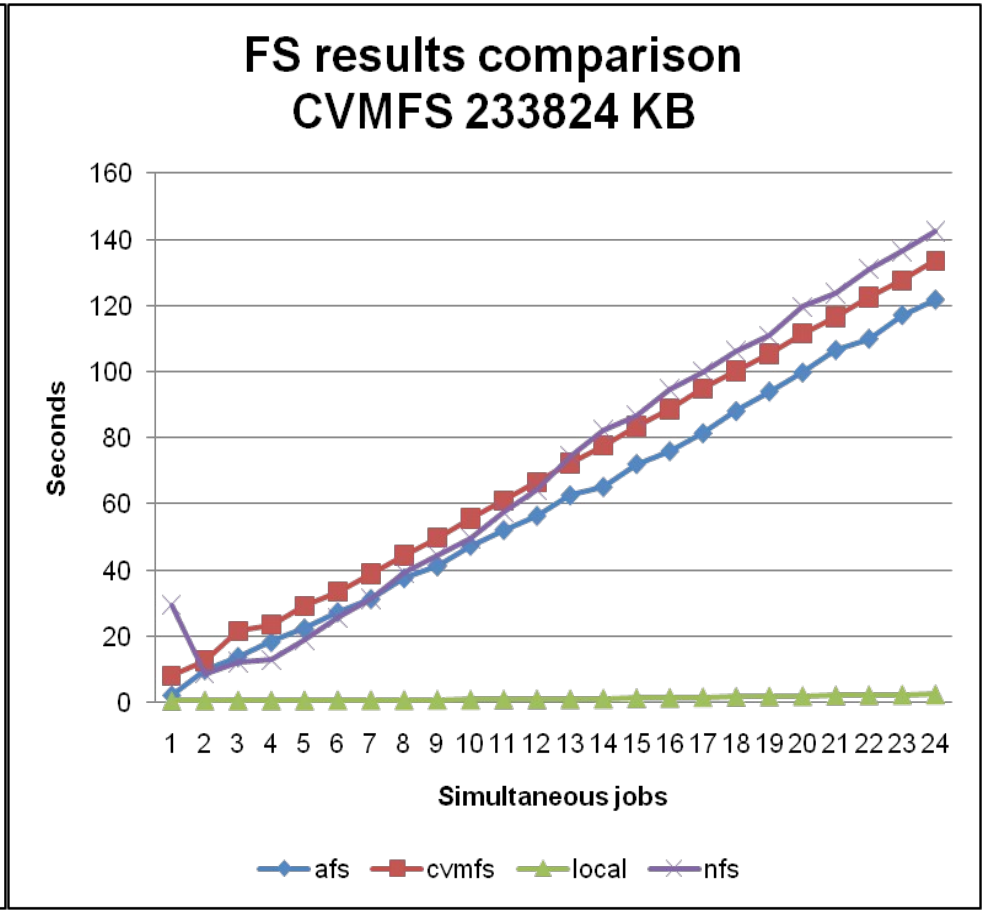
Conclusions

- Tests are complex:
  - Ressources (time, people)
  - Cost
  - Adaptation to all needs
- Several constraints:
  - Budgets
  - Requirements from users

→ Maybe need to change the way we select computers.

Other tests





Graphics graciously improved by P. Girard