Hardware requirements for a Castor instance

Castor deployment team

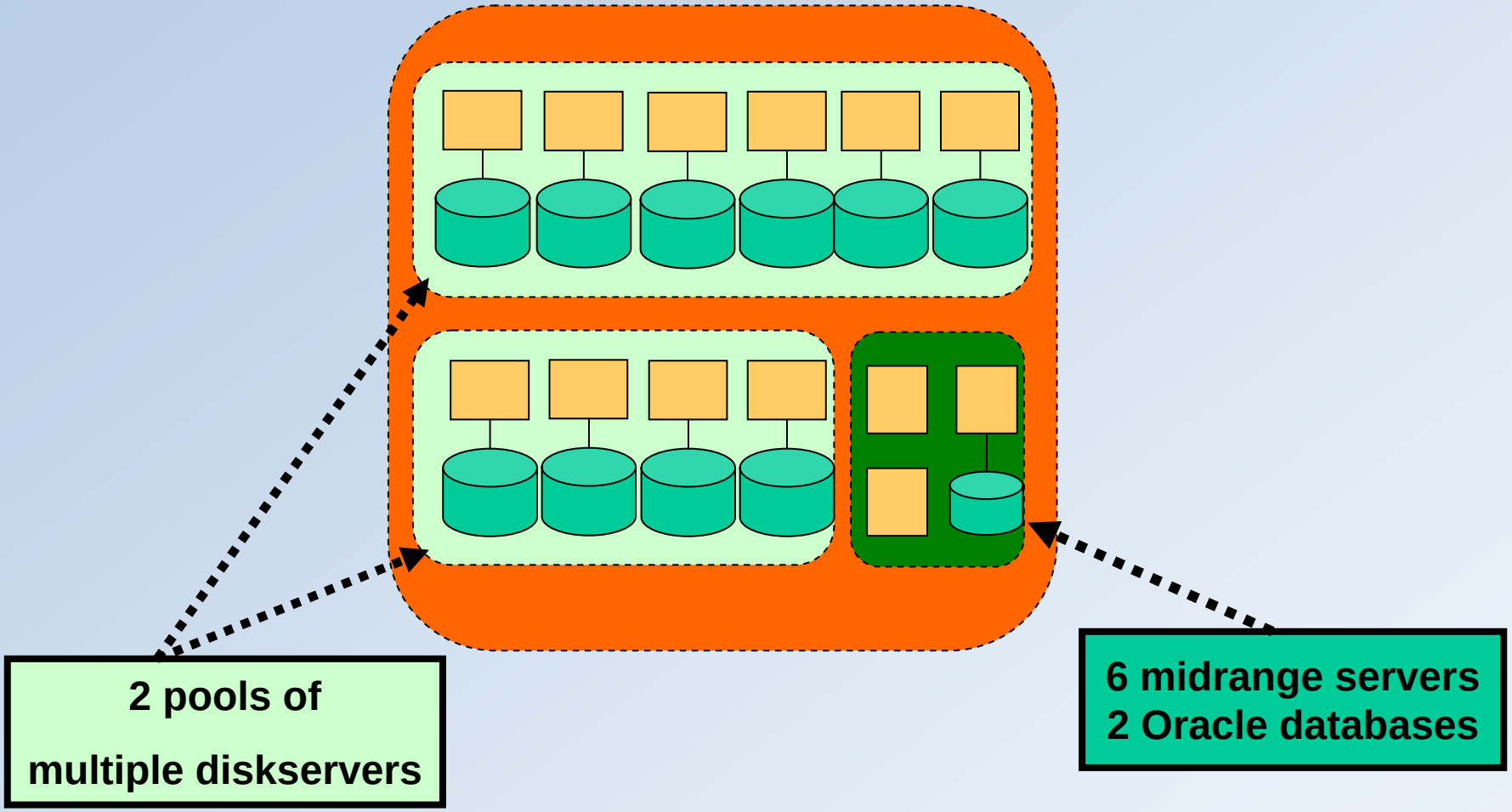
Outline

- Anatomy of Castor instances
- Hardware characteristics
  - midrange servers
  - diskservers
- Software characteristics
  - main Castor services
  - OS, filesystems
- Achieving operational scalability
  - automated installation + configuration
  - monitoring, (some) automated recovery actions
  - remaining manual interventions…

Hardware used for Castor

- Anatomy of a Castor-2 instance
  - cluster of headnodes *to run main services*
  - lots of diskservers *provide disk cache, grouped in pools*
  - two database servers *stager and dlf Oracle databases*
- Anatomy of a Castor-1 instance
  - single headnode to run stager
  - lots of diskservers *provide disk cache, grouped in pools*
- Shared infrastructure *both for Castor-1 and Castor-2*
  - nameserver cluster *database server + 4 CPU servers*
  - admin cluster *4 CPU servers*
  - tape libraries *robots, drives, servers, media*

Castor-2 instance



Hardware used for Castor

- Today
  - 6 Castor-2 instances
    - 4 LHC experiments, SC4, ITDC
    - 180 diskservers, ~800 TB
  - 18 Castor-1 experiment stagers
    - Compass, NA48, LEP, Harp, Ntof, public, lhcb, …
    - 110 older diskservers, ~200 TB
    - plus ~20 infrastructure stagers…
- "Tomorrow"
  - more Castor-2 instances
    - **`castorpublic`** for "small" experiments
    - dedicated instances for experiments with CDR? What about LHC CDR?
  - bigger Castor-2 instances
    - LHC experiments: 1.6 PB of diskcache on ~500 servers by Feb 2007
  - fewer Castor-1 stagers…

Scaling Castor-2 instances

	May 2006		Sep 2006		Feb 2007	
	space [TB]	*servers*	*space [TB]*	*servers*	*space [TB]*	*servers*
Alice	78	20	231	~60	500	
Atlas	123	25	176	~45	370	
CMS	138	27	176	~45	370	
LHCb	121	26	188	~45	370	
total LHC	**460**	**98**	**771**	**~180**	**1610**	**~480**
SC4	187	40				
ITDC	169	42	169	42	170	~40
public					~200	~100
total	**816**	**180**	**940**	**220**	**~2000**	**~600**

Experiment capacity should grow to 1.6 PB by Feb 2007, by adding ~300 servers

Assuming no SC instance needed, and a catch-all **`castorpublic`** *instance*

we will need to operate 600 diskservers

Head nodes cluster

head nodes

- Oracle databases
  - stagerdb, dlfdb
  - currently, on production instances, we use 2 “standard” diskservers *(dual CPUs, H/W RAID-1, RHEL 3)*
  - future? see Eric’s presentations…
- Castor-2 services
  - LSF scheduler, rtcpclientd, stager, request handler, dlfserver, rmmaster, expertd
  - could be run on a single PC, but…
  - currently, on production instances:
    split them over 6 “midrange servers”

“midrange server” hardware

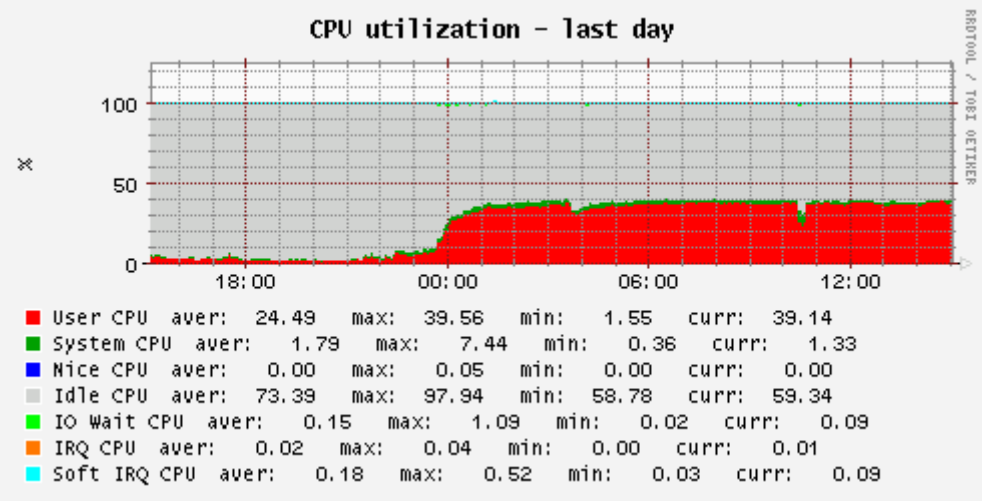
- commodity hardware, with some redundancy features
  - 2 or 3 hot-swappable power supplies
  - RAID controller (typically 3-Ware)
  - 2 hot-swappable hard disks, in a RAID1 mirror
  - 2 or 4 GB RAM
  - 2 CPUs
  - GigE
- currently, 5 different (but similar) types of H/W
- running SLC3 as the operating system

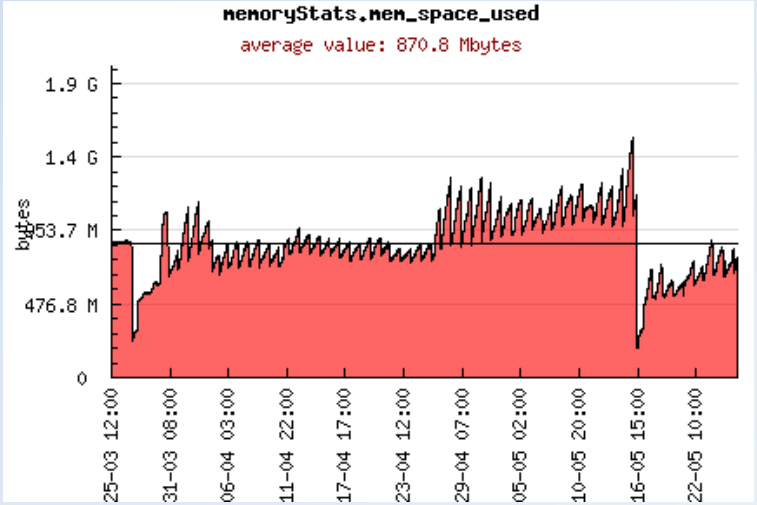
hardware performs fine


RA 48
RA 50

Building head node cluster

- Some components are “gourmand”
  - LSF schedules in a very tight (CPU intensive) loop
  - Stager, rmmaster use (leak…) memory





Current practice: use 6 servers for 7 services

To be deployed: run S/W components with failover

diskservers

general remarks

- Provide cache between analysis jobs and tapes
- But experiments also want to store ‘durable’ data
- Grouped in diskpools for distinct activities (example: CDR, experiment data sets, user files)
- Different protocols for data access
  - rfio
  - root
  - gridftp
- Lots of them…

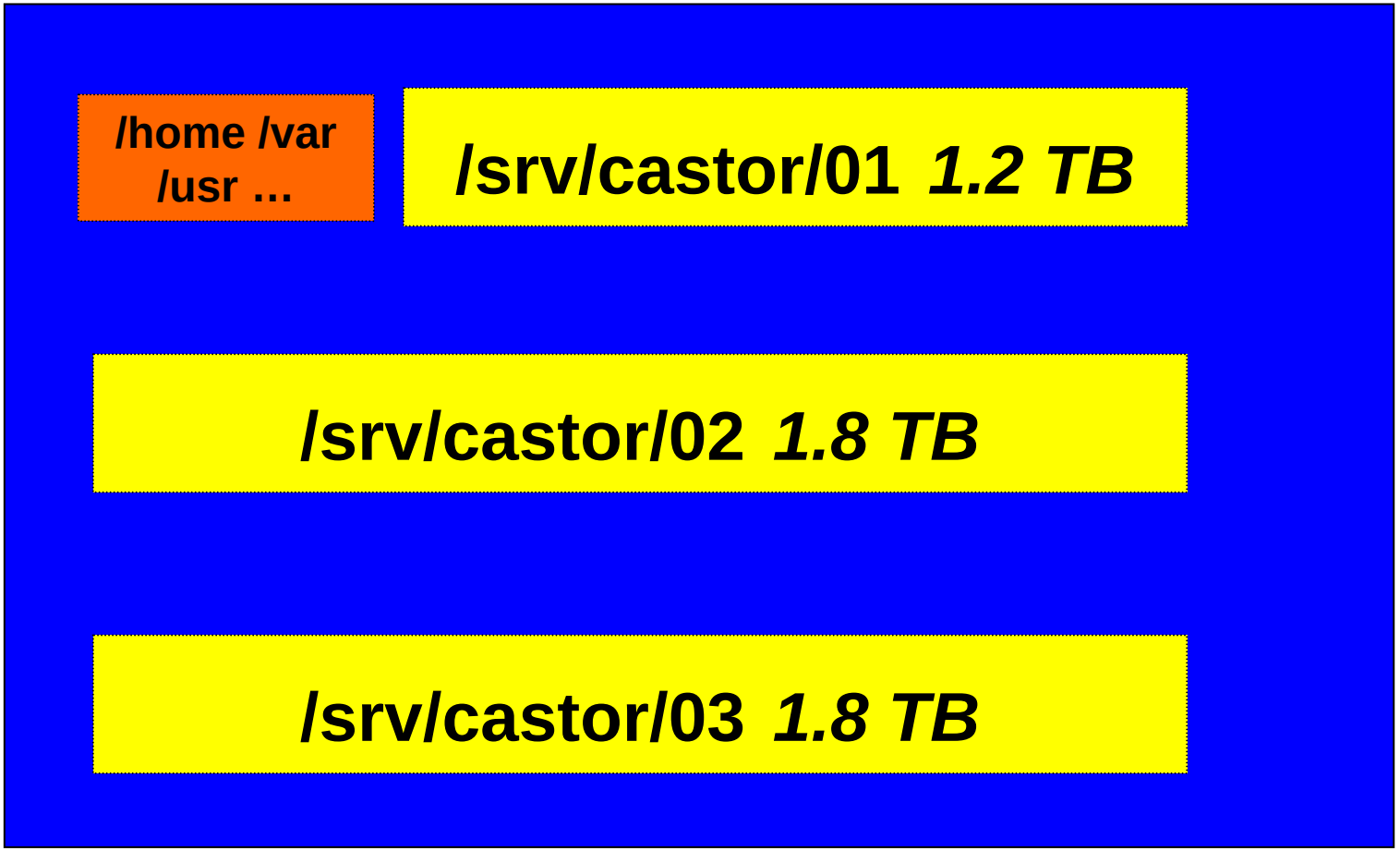
diskserver hardware

- commodity hardware, with redundancy features
- 2 or 3 hot-swappable power supplies
- 2, 3 or 4 RAID controllers *(typically 3-Ware, now also Areca)*
- 12 -- 24 hot-swappable SATA/IDE hard disks
- disksizes vary 100 – 400 GB
- 1, 2 or 4 GB RAM
- 1 or 2 CPUs
- GigE network interface
- “midrange server plus diskspace”

Diskserver configuration

- running SLC3 as the operating system *(SLC4 for SC4)*
- disks grouped in RAID5 + SPARE per controller
- XFS filesystems, one per controller
  - xfs used in production since 2 years
  - Linux support recommendation: mature and performant
  - nightly defragmentation cron job *(on mounted filesystem!)*
  - but xfs is not recommended on SLC4/32bit
- modern servers provide ~5 TB (*maybe too much…)*
- no S/W RAID (*bad operational experiences)*

3 XFS filesystems for data



Diskserver hardware

- Strong points
  - Commodity (affordable!) H/W *we already run 100's of these servers*
  - RAID does its job: individual disk failures (and the replacements) do not impact services
- Difficulties
  - we run many generations, with large variations
    - 2, 3 or 4 RAID controllers, of different vendors, with 6, 8, 12 disks each
    - disksizes between 100 and 400 GB
    - filesystems between 400 GB and 2 TB (Linux limit)
  - complex systems, requiring experienced experts ☺
    - RAID controller failures, 3Ware erratic behaviour, internal connectivity problems, backplane failure, power failures

Managing 600 diskservers…

need for automation

- going to many Castor instances means increasing need for automation:
  - installation + upgrades
  - configuration
  - monitoring
  - recovery actions…
- cannot afford handwork! *error prone, not reproducible, too expensive*
- we operate the instances in the same way as the other FIO production clusters
  - Quattor — software installation and configuration
  - Lemon — performance and exception monitoring
  - SysAdmins — H/W failures, OS-level problems

Quattorizing Castor

- describe Castor instances in CDB templates
  - describe headnodes, diskpools
  - software packages, filesystems, configuration information
  - consistent descriptions between OSes and architectures
  - not always trivial…
- install software **only** through RPMs
  - work closely with software responsibles
  - many improvements in many packages
    - *Castor, LSF, oracle instant client, rootd, STK-SSI, lcg-mon-gridftp, yaim*
    - benefits other sites and projects (hopefully)
- configure software **only** with Quattor NCM components
  - re-use/enhance existing components
    lsfclient, sysctl, access_control, rgmaclient, yaim, sindes
  - write new components
    castorconf, stageconf, rmmaster/rmnode, localgridmap

Quattorizing Castor (2)

- in quite a good shape…
- adding diskservers to diskpools (and registering them to central Castor services) is now a straightforward procedure, and scales with the number of diskservers ☺
- still lots of odds and ends to be improved
  - management of grid-mapfiles
  - mighunter, expert system configured by hand (once per instance)
  - srm.cern.ch and castorgrid.cern.ch provide very similar functionality, but software packaging and configuration is very different
- need to port to SLC4, and to **`x86_64`**

Lemonizing Castor

- Exception metrics
  - Castor-specific daemons, error messages in logfiles
  - Associated operator alarms and instructions
  - Automatic recovery actions *if really, really necessary…*
  - Automatic disabling of diskservers with alarms
- Performance metrics
  - standard Lemon plots per host/diskpool/instance
  - LSF activity plots
  - lots of Castor-specific metrics

Intrusive H/W interventions

- SysAdmins coordinate intervention between vendor and Castor service managers
  - SysAdmin analyses Lemon alarm, decides to call in vendor
  - SysAdmin asks CastorOps for downtime
  - CastorOps drain server *(stop new requests, make sure all data is safe, wait until active requests finish)*
  - Vendor fixes the node
  - SysAdmin informs CastorOps, who put node back in production
- many intrusive hardware interventions
  - May 2006: 15 intrusive interventions, 45 simple ones
- **This procedure is too heavy for the number of intrusive interventions**

Conclusion

- Castor instances use different types of hardware
  - midrange servers for head nodes
  - diskservers for database servers
  - diskservers for diskpools
- midrange servers are behaving fine, diskservers are more problematic
- automated fabric management and procedures are in place to install, configure, monitor, maintain the servers

  but always need to be improved!