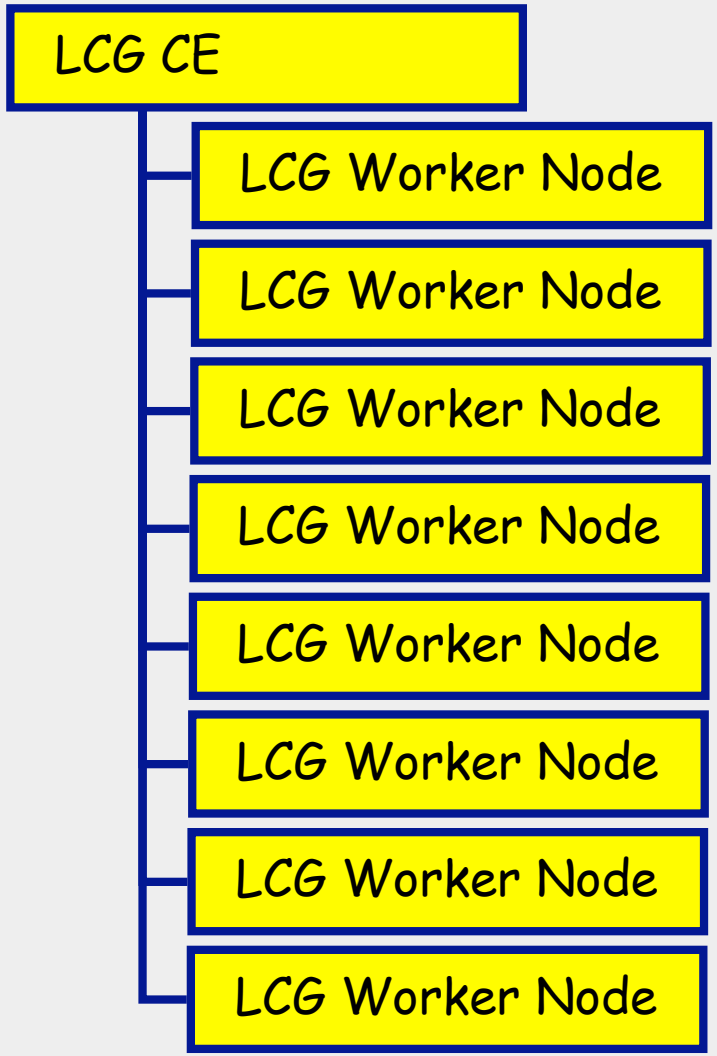
A PROTOTYPE OF A VIRTUAL ANALYSIS FACILITY

S. Bagnasco, D. Berzano *et al.*
INFN Torino

HOT TO BUILD A PROOF CLUSTER

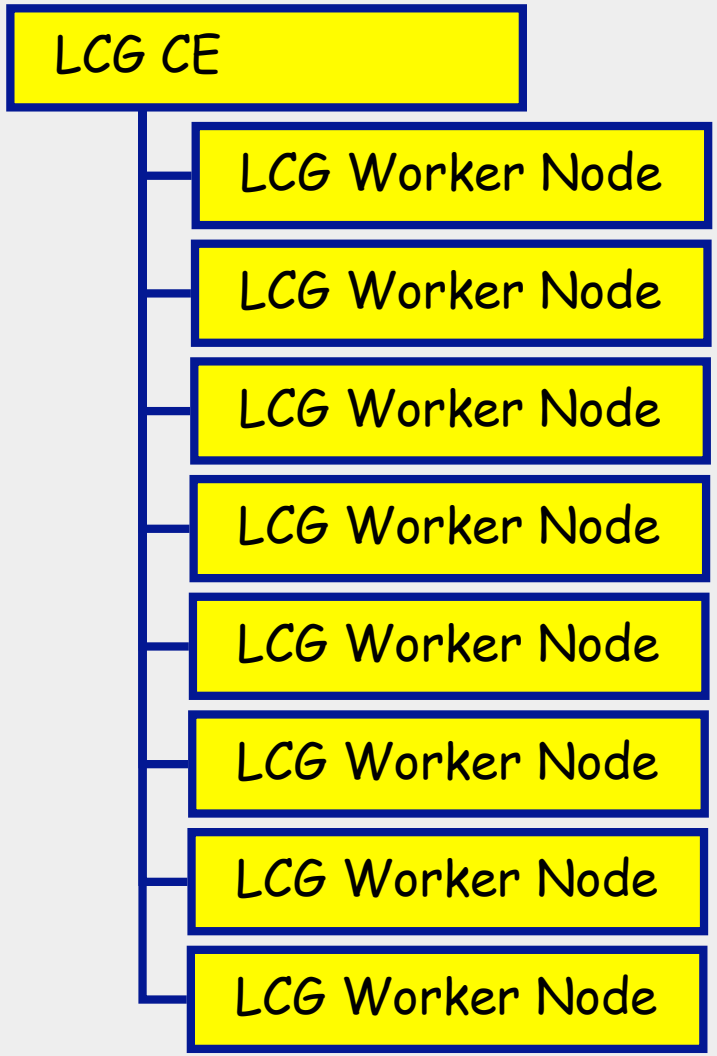
- **Statically**
  - Everybody knows the CAF…
- **Dinamically**
  - Using special local jobs (“proof-on-demand” at GSI)
  - Or AliEn JobAgents (“proof-on-the-grid”)
  - But: works well for large sites
- **Using virtual machines**
  - Semi-static approach
  - Designed for Tier-2s
  - Prototype available in Torino

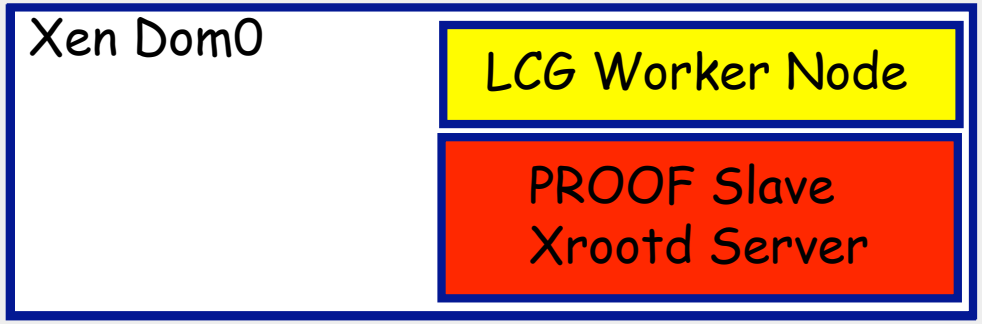
VIRTUAL PROOF CLUSTER



LCG Worker Node

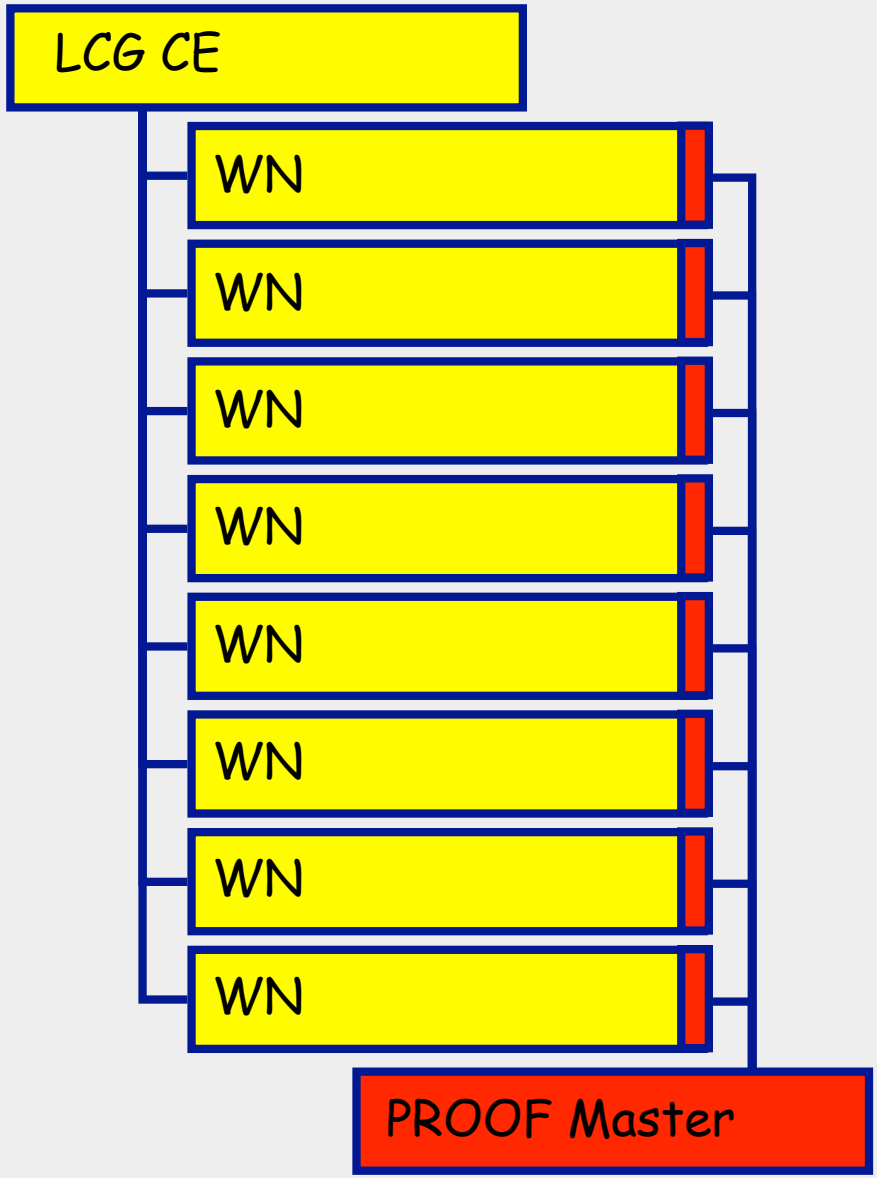
VIRTUAL PROOF CLUSTER

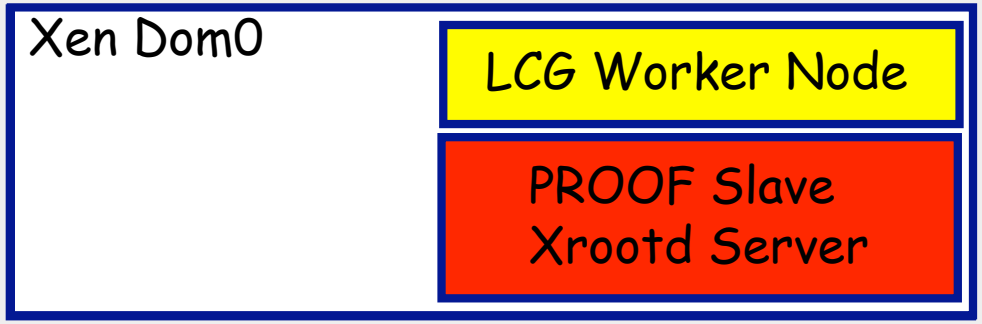




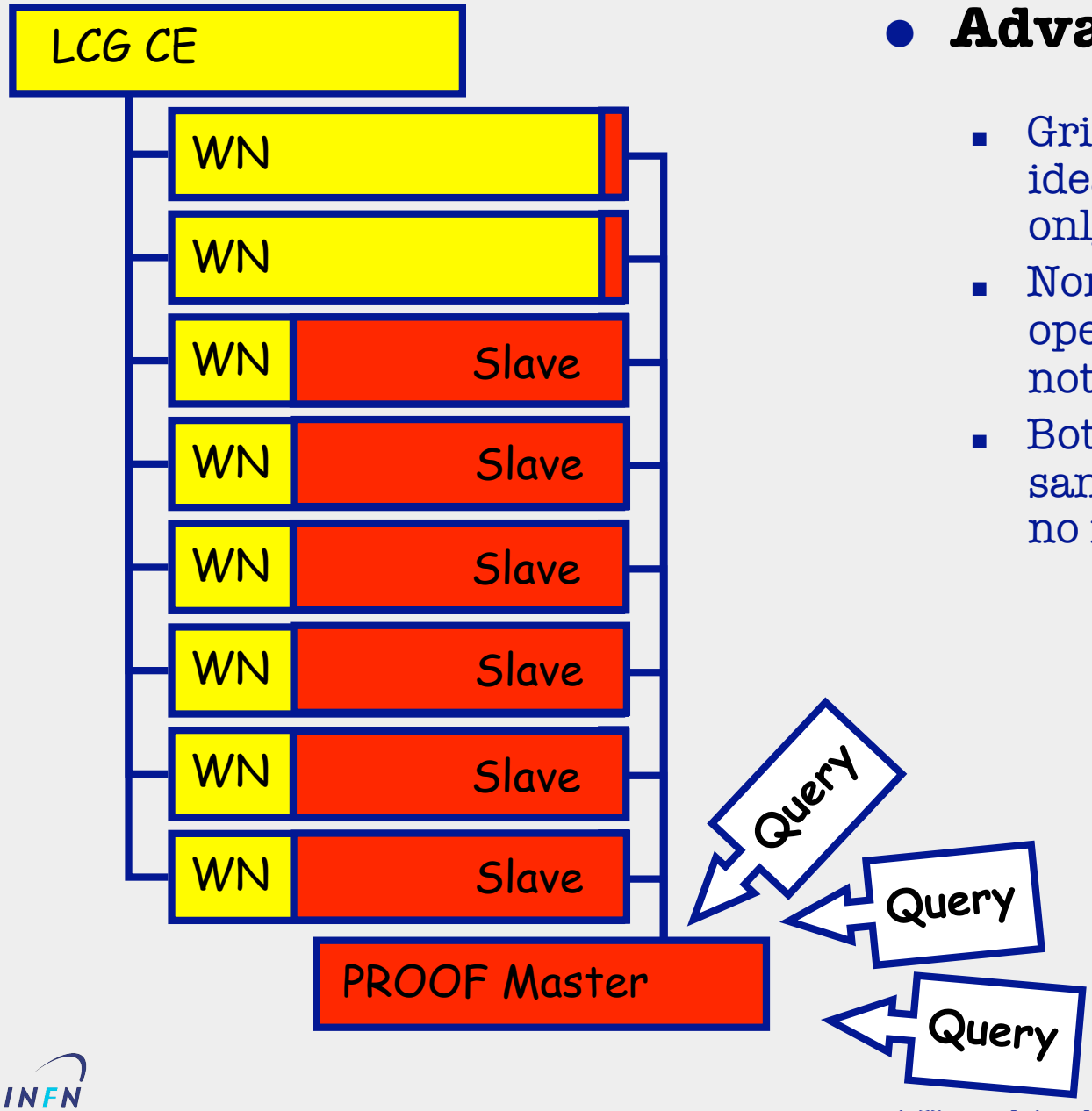
- **Xen can dynamically allocate resources to either machine**
  - Both memory and CPU scheduling priority!
  - Memory is the issue, CPU priority limit is enough
  - Normal operation: PROOF slaves are "dormant" (minimal memory allocation, very low CPU priority)
  - Interactive access: dinamically increase resources to the PROOF instances, job on WN slows down
  - Alternatively, "wake up" more slaves

DYNAMICAL ALLOCATION





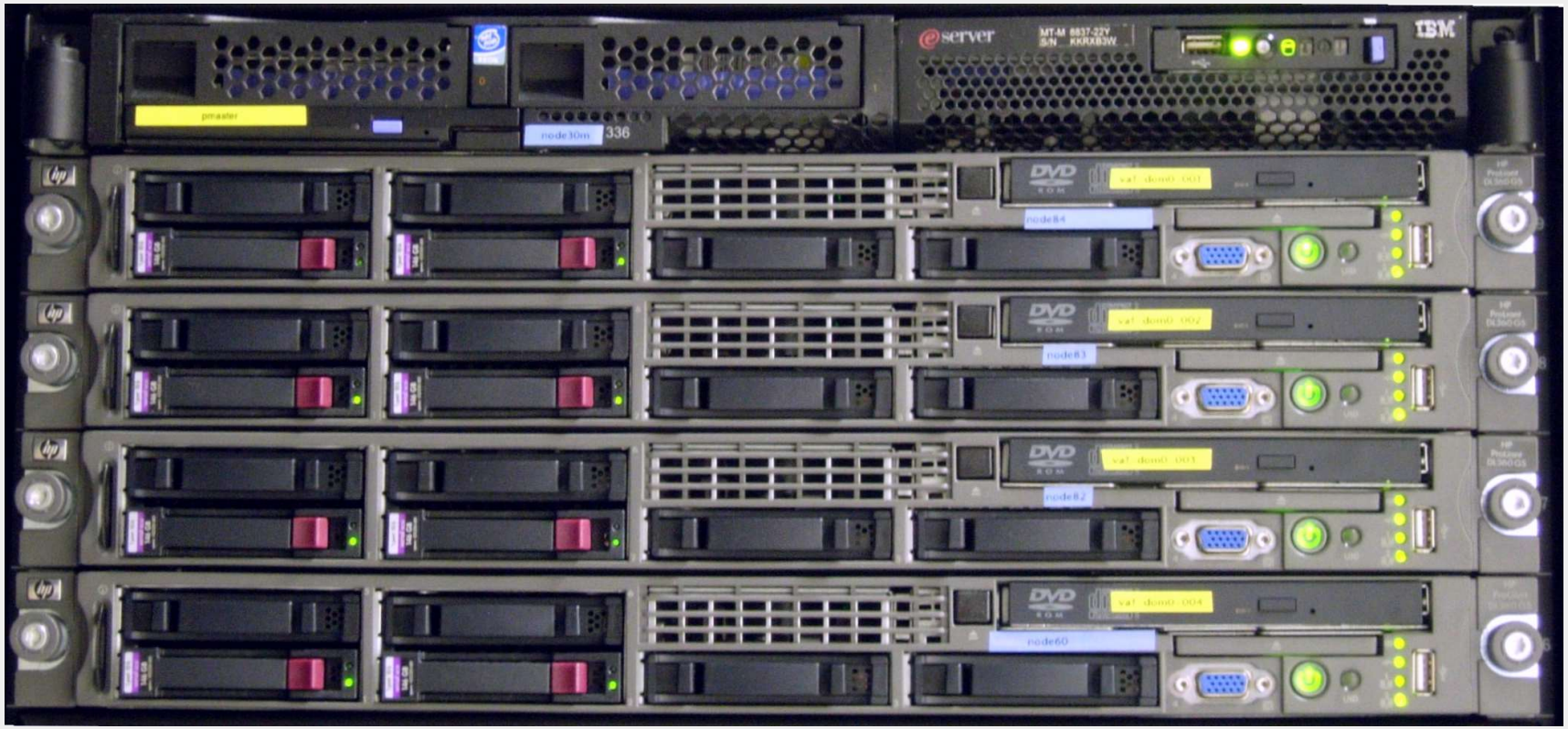
DYNAMICAL ALLOCATION



- **Advantages**
  - Grid batch job on the WN ideally never completely stops, only slows down
  - Non-CPU-intensive I/O operations can go on and do not timeout
  - Both environments are sandboxed and independent, no interference

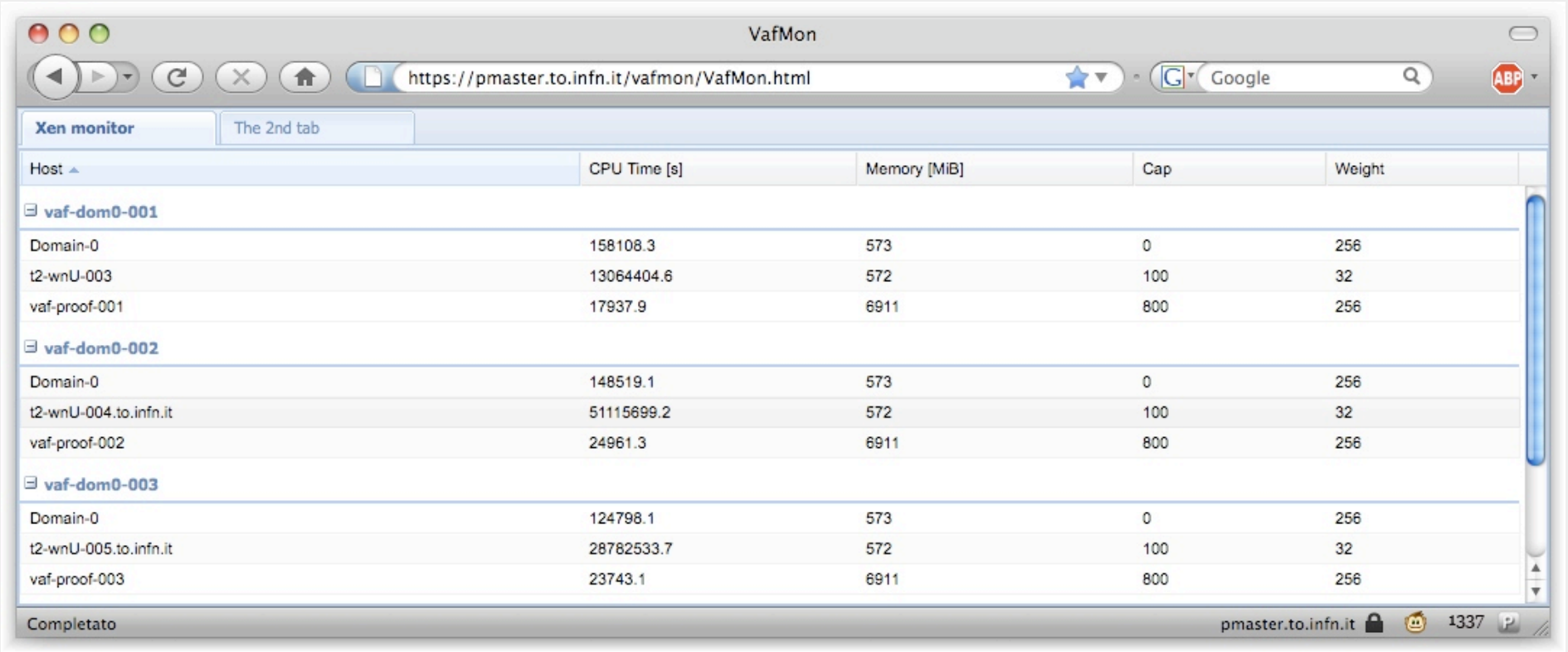
THE PROTOTYPE: HARDWARE

- 4 dual quad-core HP servers + 1 head node
- Separate disks for performance isolation
- Data access via xrootd to Tier-2 SE



THE PROTOTYPE: SOFTWARE

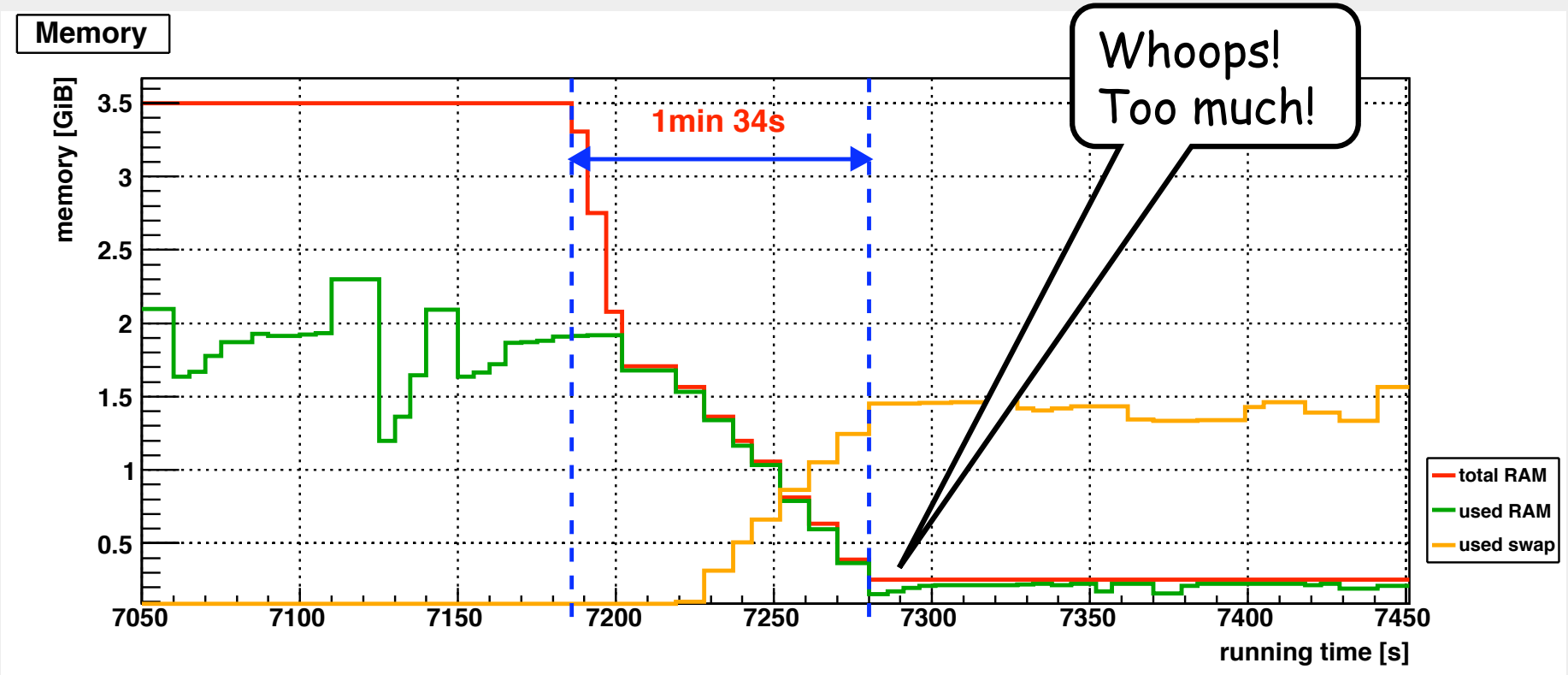
- Xen 3.2 as virtualization platform
- Simple scripts for management
- Prototype web interface
- Cobbler for installation management



Host	CPU Time [s]	Memory [MiB]	Cap	Weight
vaf-dom0-001				
Domain-0	158108.3	573	0	256
t2-wnU-003	13064404.6	572	100	32
vaf-proof-001	17937.9	6911	800	256
vaf-dom0-002				
Domain-0	148519.1	573	0	256
t2-wnU-004.to.infn.it	51115699.2	572	100	32
vaf-proof-002	24961.3	6911	800	256
vaf-dom0-003				
Domain-0	124798.1	573	0	256
t2-wnU-005.to.infn.it	28782533.7	572	100	32
vaf-proof-003	23743.1	6911	800	256

PRELIMINARY TESTS

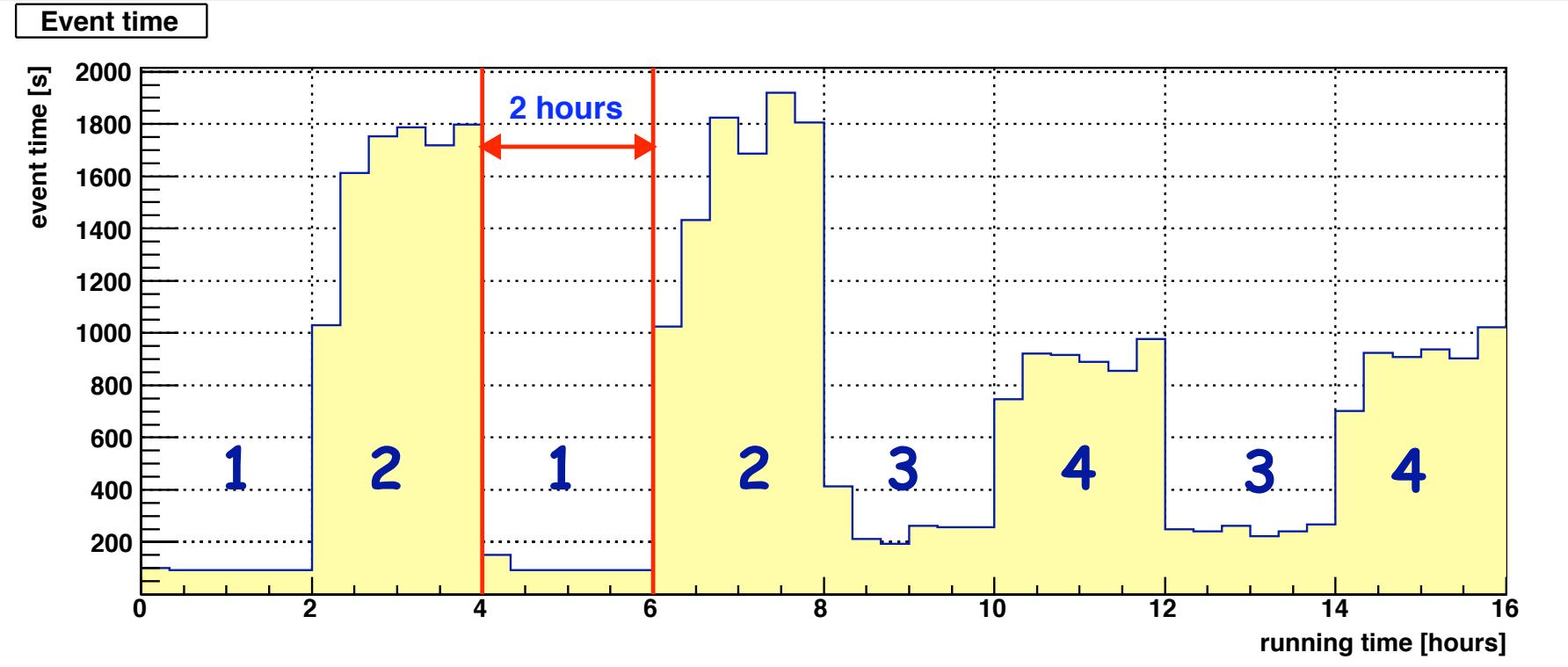
- Stealing resouces from AliRoot
- Transition time o(1min)



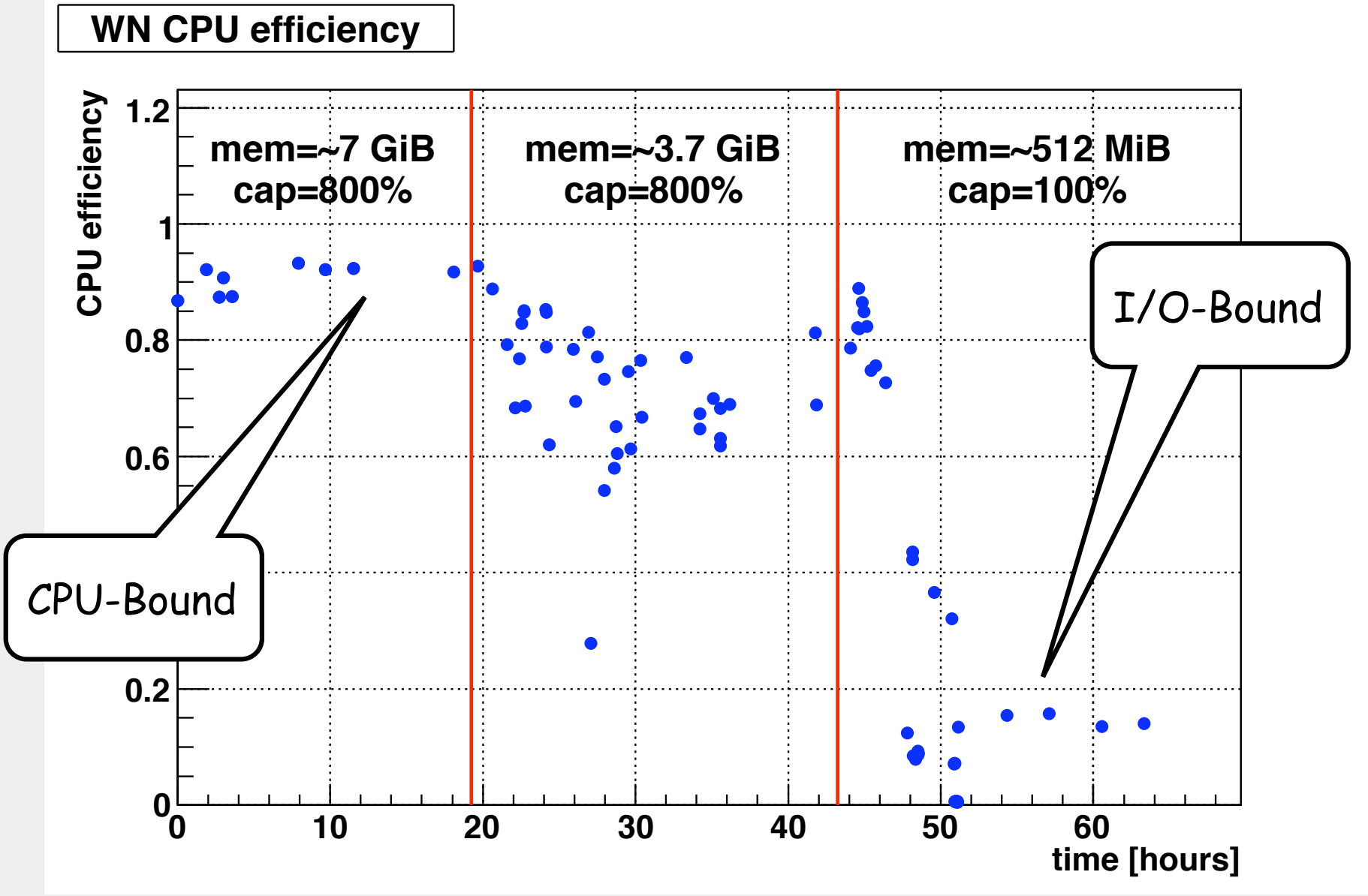
PRELIMINARY TESTS

- AliRoot generating pp events, with:

1. 3.5 GiB × 4 cores
2. 256 MiB × ½ core
3. 1.75 GiB,× 3 cores
4. 512 MiB,× 2 cores

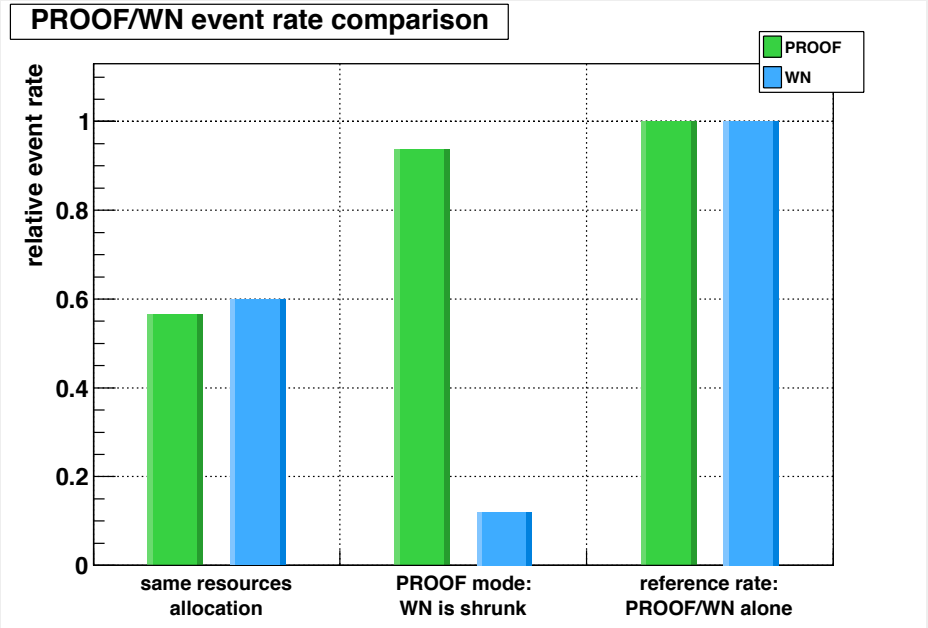


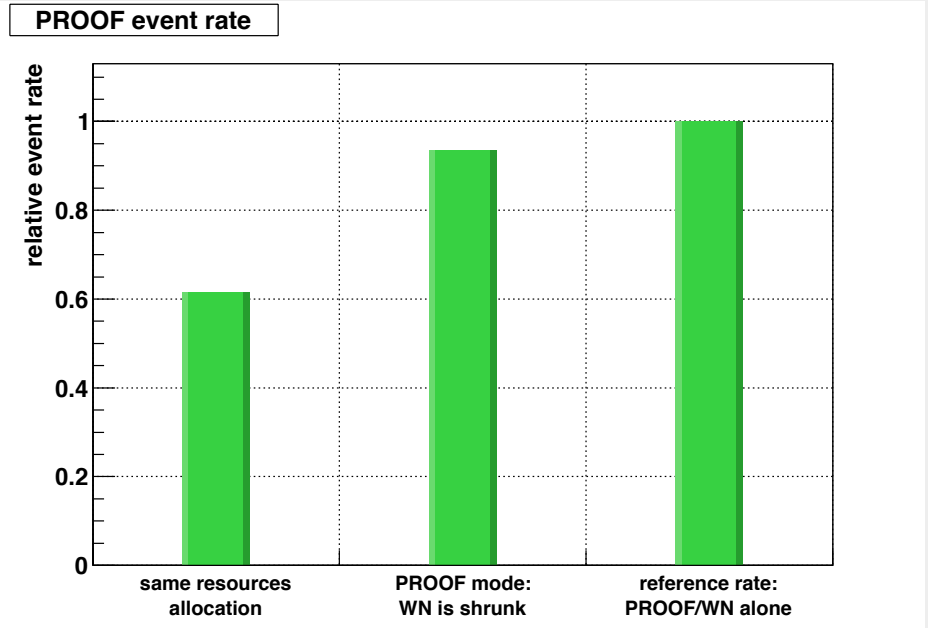
AS SEEN FROM THE GRID



PROOF EVENT RATE

- Fake load on WN
  - CPU-only
  - Small performance loss
- Real load on WN
  - Mostly AliRoot
  - Separate swap disk allows good performance isolation





ROADMAP

- Reroute network through Tier-2 core switch
  - Better access to Storage Element
  - Nearly done
- Have some real users!
- Automatic resource allocation
  - Maybe integrated with TProof::Open()
  - Load-dependent?
- Integrate with Grid Accounting system
- Replace Xen with KVM?
  - RedHat is dropping Xen support
  - Next KVM release will allow “memory ballooning”
- Explore Lustre opportunities?