IDEAS FOR A VIRTUAL ANALYSIS FACILITY

Stefano Bagnasco, INFN Torino

INFN

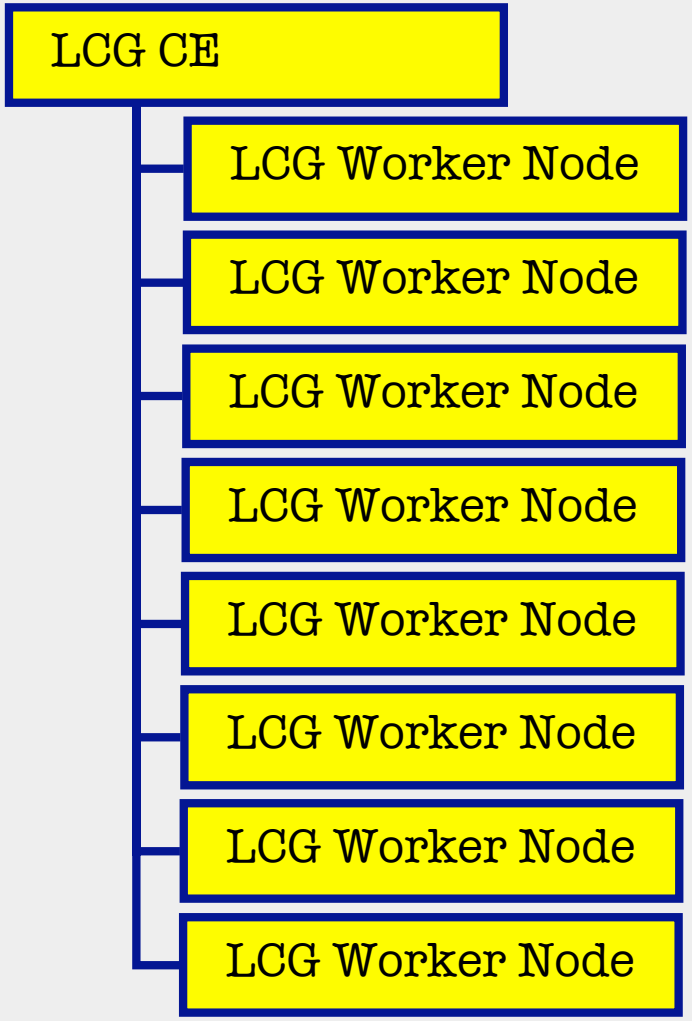
CAVEAT!

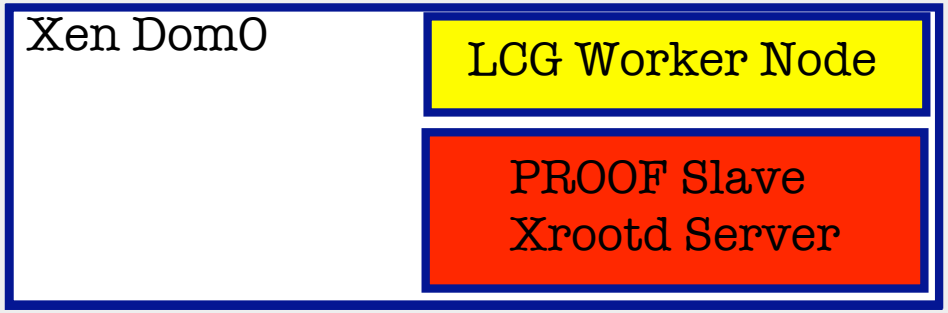
- **This is just an idea we're starting to work on in Torino**
  - We don't even have a prototype yet
  - But Federico urged for a presentation...
  - ...so I substituted facts with brightly coloured animated diagrams.

ANALYSIS IN THE TIERED MODEL

- **At Tier-1s**
  - Large number of CPUs
  - Feasible to take some out of the Grid infrastructure to build a PROOF-based Analysis Facility
  - Or may even be possible to "drain" jobs and switch to interactive mode quickly
- **At Tier-3s**
  - Very small number of CPUs
  - Probably not a Grid site, at least with gLite middleware
  - Use PROOF
- **And Tier-2s?**
  - Most resources are provided ad Grid WNs
  - In the ALICE computing model, this is where user analysis runs

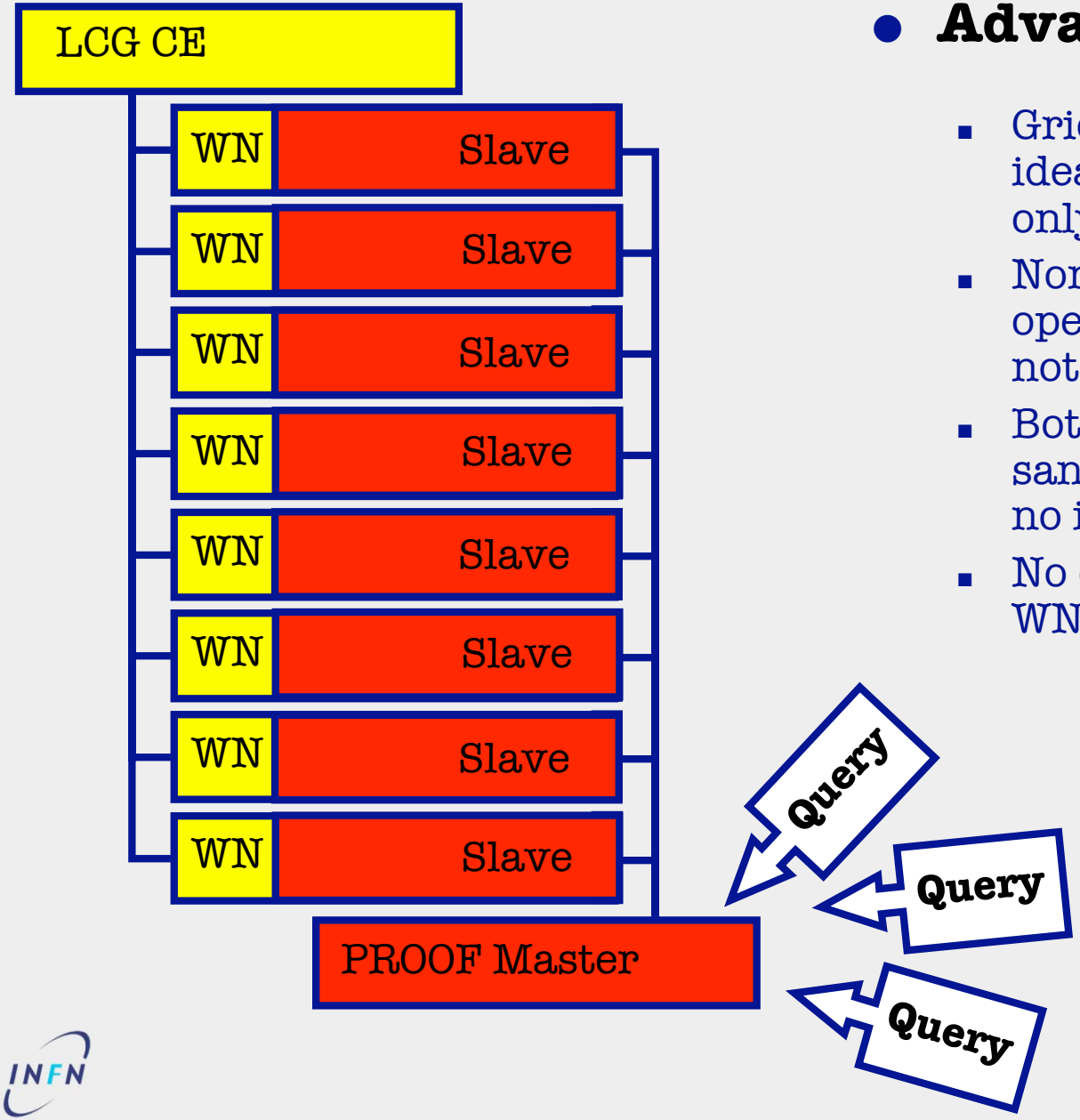
VIRTUAL PROOF CLUSTER





- **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



- **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
  - No don't actual need to be LCG WNs at all, can be anything

PROS AND CONS

- **But…**
  - Needs well-stuffed boxes to be viable
  - LCG Deployments don't mix up well with other stuff
  - There is an issue with advertised CPU power (e.g. in ETT). In a multi-VO environment is this acceptable?
  - Is it clearly possible that some WN-side batch jobs will crash even if one provides a huge swap space. Will this be acceptable?
  - …
- **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
  - No actual need to be LCG WNs at all, can be anything
  - Can have quickly a working prototype, and add advanced features later

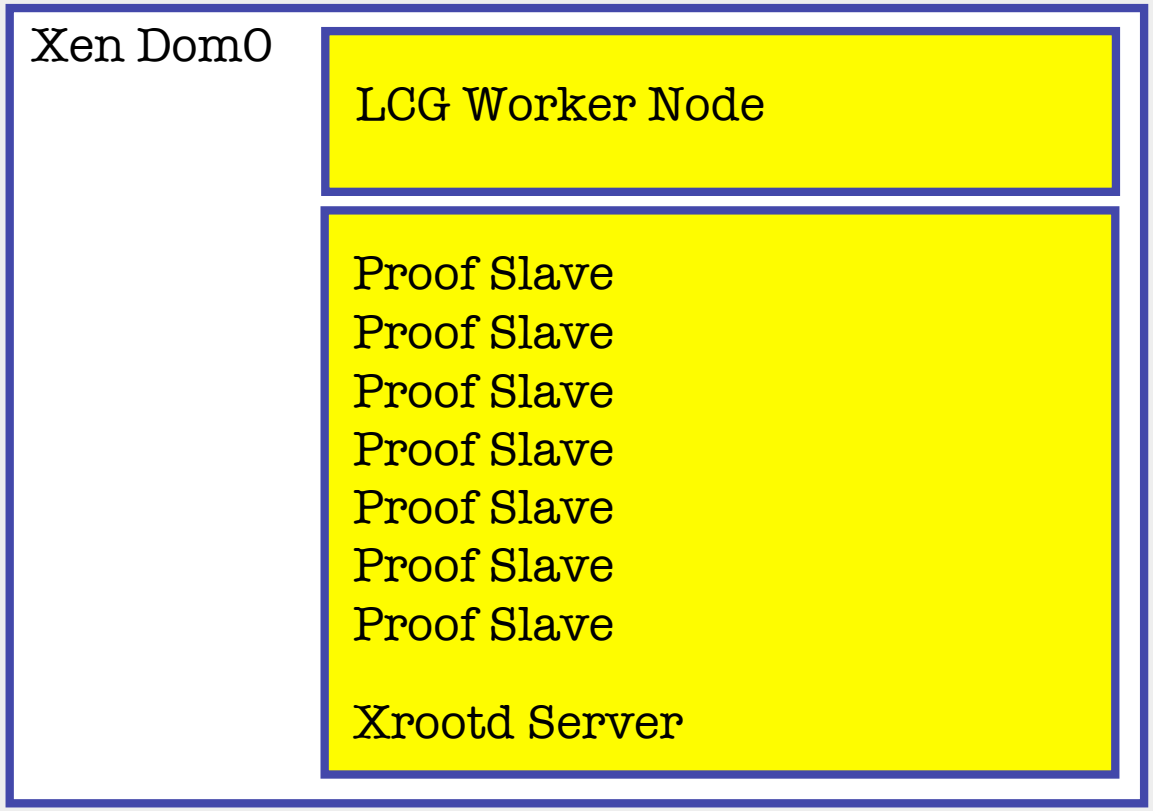
Shopping list:

- Xen
  - Two (or maybe more) virtual machines per physical one
- LCG WN (or whatever)
  - On one of the virtual machines
- PROOF + xrootd
  - One (or more) slaves per physical machine
  - One head node (master)
- "Director"
  - Globally manages the resource allocation
  - This is the missing piece to be developed
  - Next slide!

THE MISSING PIECE

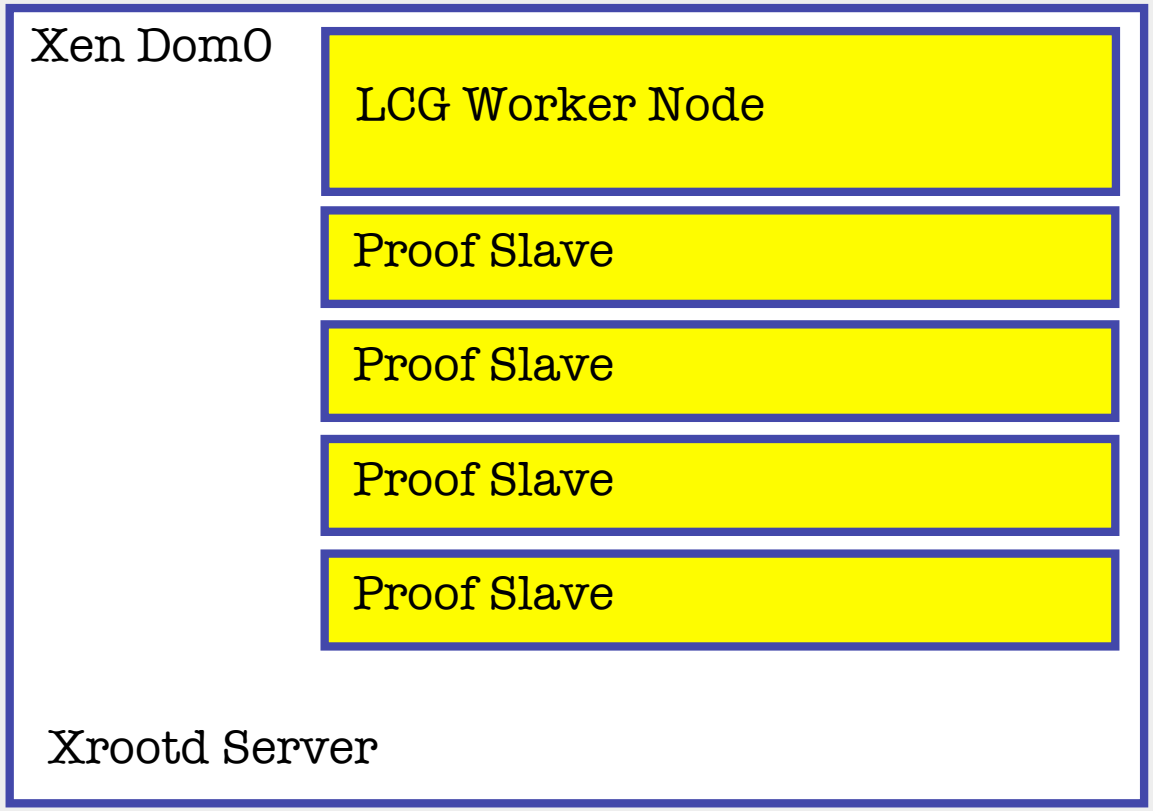
- Can easily have a semi-static prototype
  - Or completely static, just setting CPU limits
  - Just a "slider" to move resources by hand
  - This is not very far, essentially a deployment issue
- An idea by P. Buncic: use SmartDomains
  - https://sourceforge.net/projects/smartdomains
  - Developed by X. Gréhant (HP Fellow at CERN Openlab)
  - This application is not its primary use case
  - Not all the needed functionality is there
- Following step is a truly dynamical system
  - Coupled with PROOF Master
  - Measures load and automatically starts more workers/assign more resources as needed

DEPLOYMENT ON MULTICORE MACHINES



- One VM, several PROOF Workers
  - Assign more resources to the VM when starting a fresh worker

DEPLOYMENT ON MULTICORE MACHINES



- One VM per PROOF Worker
  - Maybe running xrootd on Dom0?

Ideas, advice & suggestions please!

bagnasco@to.infn.it

INFN