A prototype of a dynamically expandable Virtual Analysis Facility

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

INFN Torino

ACAT, Erice – November 5th, 2008

Introduction

Interactive analysis with PROOF

Some tasks aren't meant to be submitted to the Grid:

- Some results must be obtained promptly: no time to wait!
- What if the analysis code is buggy!? Much time wasted!

⇒ *See yesterday's talk by Gerri Ganis*

Need for a "local" *(i.e. physically close to the user)* interactive analysis facility:

- Just like the CAF: PROOF plus a xrootd disk pool
- As users increase, CAF won't suffice
- "Local" ⇒ very fast assistance when the facility gets stuck

But...

- Computing power is very expensive
- Interactive analysis facility would be idle most of the time *(i.e. at night)*

Introduction

The tiered model in HEP computing

Tier-1s

- Large number of CPUs ⇒ take out some to dedicate to interactive analysis *(i.e. PROOF at CAF)*
- Even possible to drain jobs and switch to interactive mode quickly if number of slots is very high

"Tier-3s"

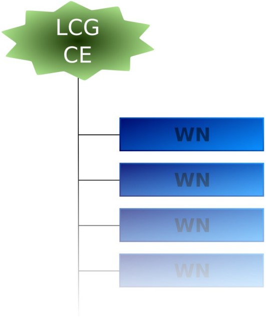
- Very small number of CPUs
- Mostly not even Grid sites ⇒ parallel analysis with PROOF

...and Tier-2s?

- Most resources are provided as Grid WNs
- Here is where user analysis run
- Not feasible to dedicate nor reboot machines to PROOF

The Xen approach

A Virtual Analysis Facility



"Night" configuration

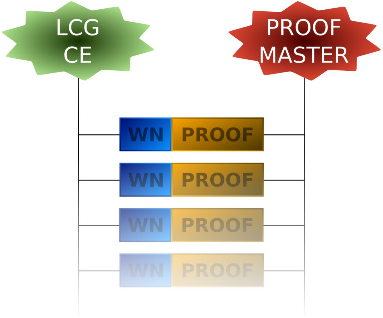
low memory and low CPU priority to PROOF

Xen can dynamically allocate resources to a virtual machine!

- CPU scheduling priority ⇒ credit scheduler: cap, weight
- Xen can dynamically change memory too!
- When shrinking WNs jobs slow down (swap!) without crashing (tested)
- Sandboxing: PROOF failures do not propagate to WN

The Xen approach

A Virtual Analysis Facility



“Day” configuration
much memory and higher CPU priority to PROOF

Xen can dynamically allocate resources to a virtual machine!

- CPU scheduling priority ⇒ credit scheduler: cap, weight
- Xen can dynamically change memory too!
- When shrinking WNs jobs slow down (swap!) without crashing (tested)
- Sandboxing: PROOF failures do not propagate to WN

Xen feasibility benchmarks

Introduction

sysbench 0.4.8 used (simple and scalable)

CPU

- Primality test on the first 20000 integers

Memory (RAM)

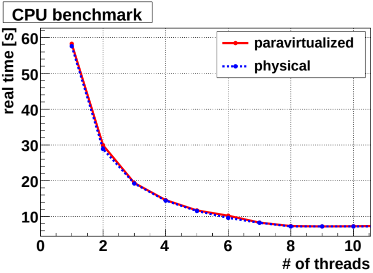
- Variable concurrent threads that write 5 GiB on memory

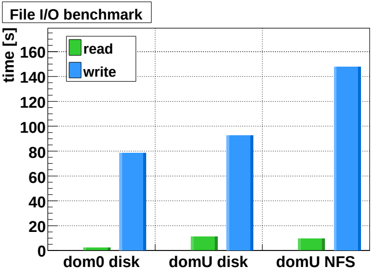
File I/O

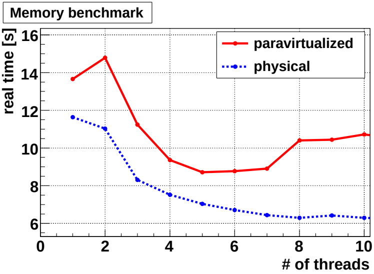
- Read/write 5 GiB (128 files, 40 MiB each)
- 10 concurrent threads (one per core + overbooking)
- Different measures ⇒ average (first read measure discarded because of caching)

Xen feasibility benchmarks

Results and comments


CPU benchmark


File I/O benchmark


Memory benchmark

- Paravirtualization ≠ emulation
  ⇒ *hypervisor schedules everything*
- Memory is up to ∼ 40% slower
  ⇒ *but our tasks are CPU-bound*
- Disk I/O is slower
  ⇒ *but network storage is used*
- Swap is an issue
  ⇒ *separate physical disks*

The Prototype

Technical details

Hardware

- Four 8-core machines (plus one head node)
- 8 GiB RAM each (more underway)
- Six disk slots, two used (no RAID ⇒ swap!)

Current configuration

- PROOF on virtual slaves, gLite on virtual WNs (2 of them in production since several weeks)
- Script wrapper to Xen commands to dynamically control resource allocation (inc. # of PROOF workers per node)
- An install server (Cobbler) to ease mass installations
- Under developement: web interface based on GWT/PHP
- Currently no auth in PROOF (but will be GSI soon)

The Prototype

Data access models in PROOF

PROOF xrootd pool

- CAF-like
- In place for tests now
- Limited by current hardware (SAS disks), may be expensive with blades currently used for WNs
- Not useful for a small cluster (pool too small)

Direct access to SE

- Using AliEn + xrootd ⇒ needs GSI authentication
- Not tested even at CAF
- Seems the way to go ⇒ ideal solution for Tier-2s

Prototype benchmarks

Introduction

Grid

- Both fake, CPU-only load and real world ALICE tasks
- Number of crashed jobs *(is it greater than average?)*
- Main test: running virtual WNs in production for some time (underway)

PROOF

- A PROOF analysis is run on 32 workers
- 200 evenly distributed files (exactly 50 files per node)
- Event rate is measured three times and averaged

The real ALICE jobs tests were performed on two machines only, because we've only 2 out of 4 WNs configured right now.

Prototype benchmarks

Introduction

Grid

- Both fake, CPU-only load and real world ALICE tasks
- Number of crashed jobs *(is it greater than average?)*
- Main test: running virtual WNs in production for some time (underway)

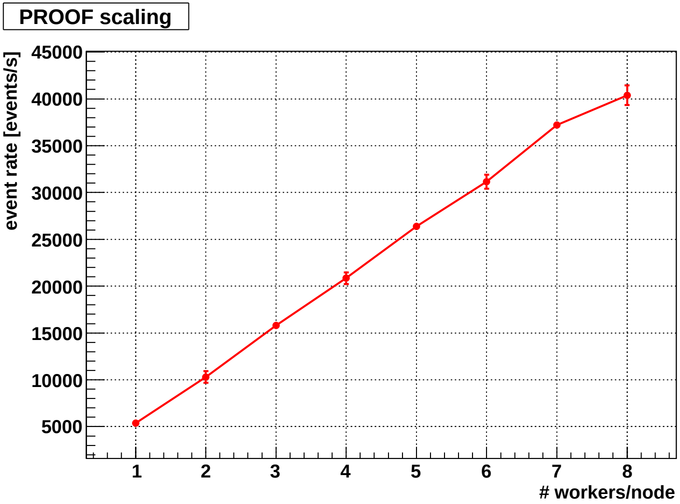
PROOF

- A PROOF analysis is run on 16 workers
- 100 evenly distributed files (exactly 50 files per node)
- Event rate is measured three times and averaged

The real ALICE jobs tests were performed on two machines only, because we've only 2 out of 4 WNs configured right now.

Prototype benchmarks

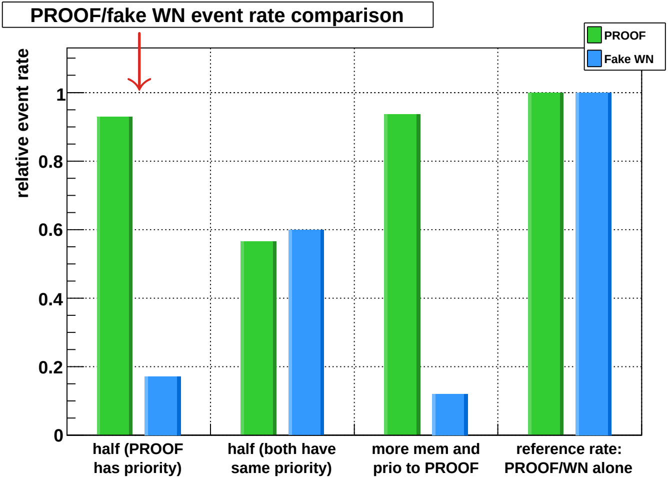
PROOF scaling



PROOF does scale (as expected)

Prototype benchmarks

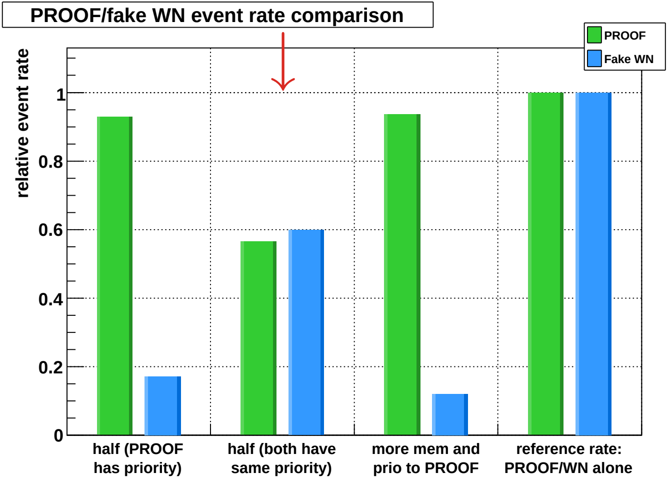
Event rates with different resources: fake load on WN



- Half RAM each domU ($\sim$ 3.7 GiB), cap=800%, PROOF has priority
- PROOF takes much of the CPU as expected (weight=domU “nice”)

Prototype benchmarks

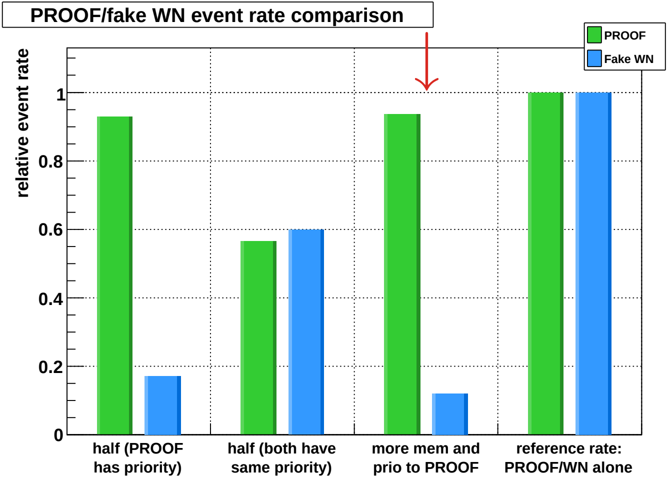
Event rates with different resources: fake load on WN



- Half RAM each domU ($\sim$ 3.7 GiB), cap=800%, same priority
- CPU time is equally divided (as expected)

Prototype benchmarks

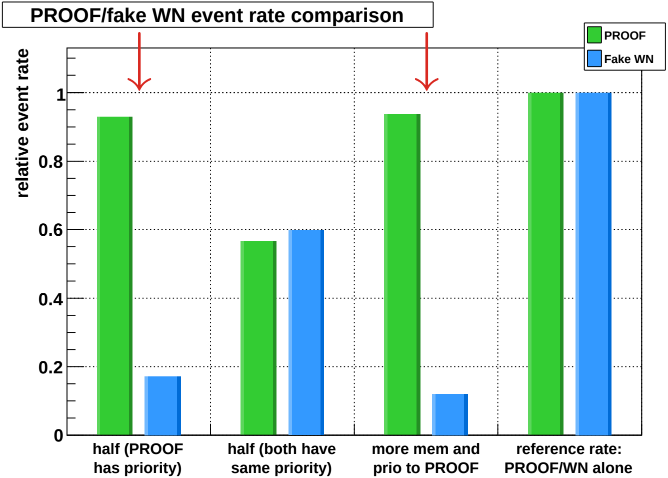
Event rates with different resources: fake load on WN



- ∼ 90% RAM to PROOF (∼ 7 GiB), cap=800%, same priority
- Same results as with less RAM ⇒ our tasks don't use much memory

Prototype benchmarks

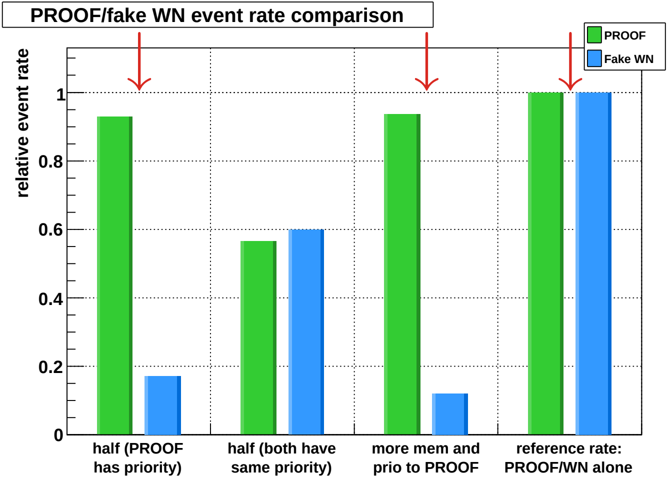
Event rates with different resources: fake load on WN



- ~ 90% RAM to PROOF (~ 7 GiB), cap=800%, same priority
- Same results as with less RAM ⇒ our tasks don't use much memory

Prototype benchmarks

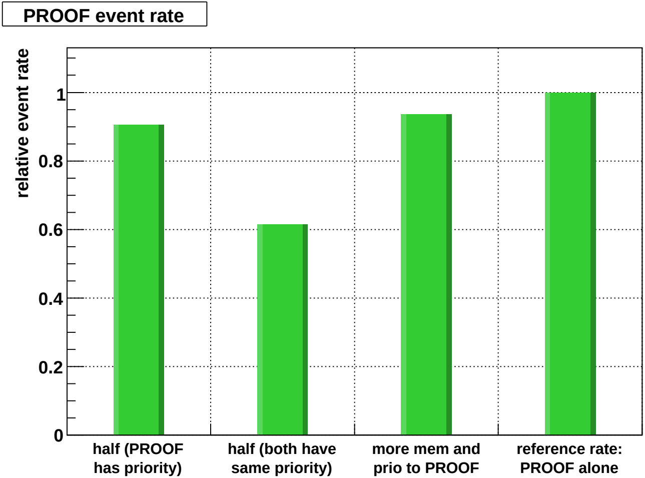
Event rates with different resources: fake load on WN



- With load on WN ⇒ event rate slightly less than maximum!
- Xen scheduler works fine

Prototype benchmarks

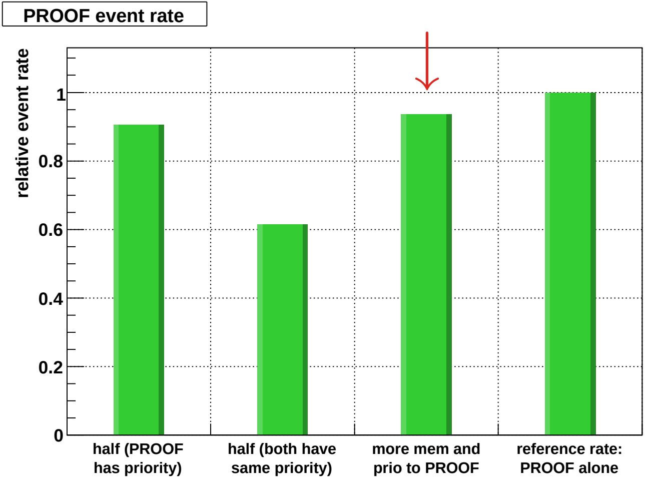
Event rates with different resources: real Grid jobs on WN



- Same PROOF rates as before, even if WN heavily swaps
- Xen and different physical disks guarantee perfect domUs isolation!

Prototype benchmarks

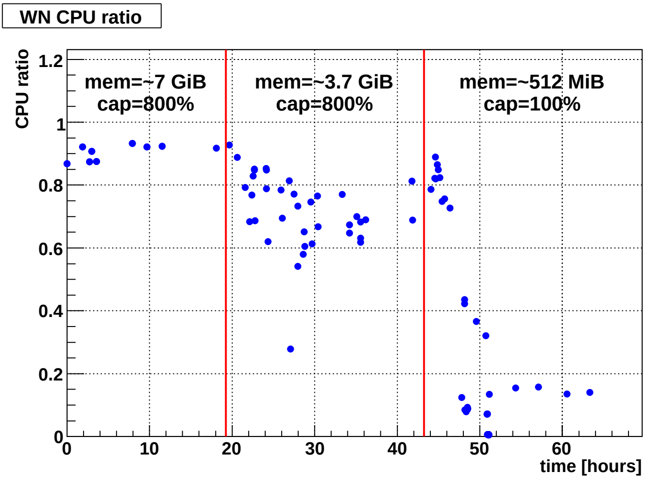
Event rates with different resources: real Grid jobs on WN



- Same PROOF rates as before, even if WN heavily swaps
- Xen and different physical disks guarantee perfect domUs isolation!

Prototype benchmarks

Grid jobs CPU usage with different resources



- As swap usage increases, jobs turn from CPU-bound into I/O-bound
- No more job failures wrt average ⇒ they slow down but don't crash

Outlook

GSI authentication

It's standard ⇒ Grid users are already accustomed to it

Auth to AliEn is not propagated to PROOF workers in current Analysis Framework ⇒ easy to fix

Better resource allocation

Night/day cronjob is too much coarse-grained!

Dynamically when `TProof::Open()` *called ⇒ transparent for the user and load-dependent*

LCG accounting

Integration into LCG accounting system through DGAS

Monitoring and management

A colorful web interface is currently under development

Thanks!

Questions, suggestions, ideas, criticism, praise?