

**What is needed/possible :
DAMNAG the day after
(del senno di poi son piene le fosse)**

Try to assess needed cpu/disk ratio to define a hardware requirement that makes sense and can be implemented in the immediated future. I.e. try to figure out how much CPU and disk we need and why, leaving to rest of workshop to define the architecture.

A partly failed attempt to define one possible “integrated model”.

From Physics Needs to Computing Needs

- Try to estimate Cpu & Disk needs to fulfill Pierre's wishes
 - Focus on summer 2002
- Need assessment vs. Choice justification
- One way of doing things that
 - Satisfies "some needs"
 - Can be done
- Need approximation and guesses
- Assume some usage pattern
 - Pick one pattern and make it possible

Time frame

- Plan for Summer 2002
 - Is not: do nothing till June, then work on 200pb^{-1}
- Continuous approximation:
 - Start with
 - ☞ Little luminosity
 - ☞ Little selection
 - ☞ Big events
 - Move on to
 - ☞ More beam
 - ☞ Tighter cuts
 - ☞ Reduced event size
- Data size increase faster than Integrated Luminosity, need resources much much sooner than summer (or spring)

Data Set Names (usage patterns)

- Primary = production output
 - Call it DST. Do not care how is written.
 - By definition is such that can be ran through production again. It has as many events as L3 triggers.
 - Number of streams etc. as from “literature”.
- Secondary = user input
 - Call it PAD. Do not care how is written. (could be same format)
 - By definition is such that is easy to go through.
 - The number of event may be much less then L3 (muons, J/Psi) or the same (Jets). It is the “signal sample” (Pierre).
 - Better (?) if can be “updated” without going back to Primary (e.g. new algorithm, but not new selection)
 - ☞ OR WE MAKE THIS A REQUIREMENT ?
- Tertiary = user output
 - ☞ this is the Ntuple to be copied to Russia

The Big picture

- Need to worry also about making the 2nd data set, not only using it
 - The “old plan” was to do that on the CAF
 - Now we somehow look at the CAF as the place to “access” it
- Maybe 2nd “disappears, production output is directly PAD ?
 - How to remake it fast
 - What about routinely accessing only a piece of a stream ?
- Reprocessing = Primary->Secondary “fast”, not at DAQ speed.
 - less CPU (do not re-track) then higher I/O
 - ☞ will not accept a long wait for a “simple” task

Analysys of Disk Resident 2nd Data Set

- Focus on disk resident samples
 - Easier. Likely enough for a lot. Will see later how to handle “disk overflow”.
 - Good to learn what can be done with disk resident data
 - Only way to be fast

- Need:
 1. Time to go through a sample (our goal !)
 2. Cpu/event
 3. Disk/event
 4. #event
 5. #users

1. Time to go through one Data Set

- Take as goal: 1 pass /day
 - No “need” to be faster
 - ☞ look at plots for one day before running again
 - Fast enough to avoid risk of:
 - ☞ moving targets
 - ☞ floor shifting under your feet
 - No problem if wrong by a factor (2day, 3 days, 4 hours...)
 - ☞ as long as not by $O(10)$

2.+3. CPU/event & Disk/event

- Be optimistic: stick to “best of Benchmarks” and “rumored best data compression”
- We only get better, users will have to be smart and efficient
 - Do not plan on present situation, but be very clear about what people will have to do to
- <http://www.webster.com/cgi-bin/dictionary?book=Dictionary&va=leads>
 - ☞ to guide on a way especially by going in advance
 - ☞ to guide someone or something along a way
 - This is the way you should be going
 - If you go this way you will have resources to do such & such

2. Cpu/event

- Assume 1GHz CPU
- Assume I/O as good as Pasha's Stntuple Track Branch
 - 5500ev/11 sec on 500MHz P3 = 1msec/ev on 1GHz
 - ☞ same result for FileInput on bbbar MC (BM report)
 - ☞ JETs only PAD already much faster (Pierre 4 Robert)
 - No matter PAD or ROOT, it has to be 1msec/event
- Add analysis:
 - Pasha's: 0.1 msec 1-10 msec/event 1Khz-100Hz
 - TrkAnal: 1 msec Assume 300Hz as "typical"
 - PhiKAnal 3.6 msec Faster/slower anal will just take
 - Twice.. 10 msec 3x less/more time
- 300 Hz = 1 million event/hour
 - Will meet the one-pass/day goal on "Summer data" (20Mev total)
 - Will need to do better (or be more patient) after Summer

3. Disk/Event

- Rumor : 50KBytes/event
- Start with 200 (Pierre)
- Make all calculations for 100KB/ev
- Not a problem for performance as long as user only reads a part
 - Pasha's tracks: 13KB/ev, 7MB/sec at 1KHz
 - ☞ add analysis: 3x slower = 2MB/sec
- But will want to store full event together
 - Need 100 KB/ev on disk:
 - If have to stage in/out, need 30MB/sec (per user)
 - ☞ that's why disk resident is much better
 - ☞ at least "mostly" disk resident
- 300Hz job "consumes" ~100 GB/hour

4.+5. #Events and #Users

- 6 months at 30 Hz = 500million events
 - Pierre: signal from “literature” is 20 Mev
 - Is the rest all $B \rightarrow h$?
- Assume one data set = 2 million event (add or take 2x)
 - Also for $B \rightarrow p\bar{p}$, $b \rightarrow J/\Psi + \dots$
 - Leave inclusive $B \rightarrow \text{hadron}$ for R&D
 - ☞ do we need that by summer ?
- Assume 200 users
 - Means running data access jobs at the same time
 - ☞ hopefully someone is making plots on final Ntuple, reading, thinking, writing notes/papers

Disk/CPU ratio

- Will not have 200 CPU's by summer
- Assume 5 users per CPU (40 CPU's)
 - 1 user runs for 4 hours = 400GB = 4Mevents
 - Other users share the same data (or run elsewhere)
- Unit is then
 - One 1GHz CPU + 400GB disk
 - ☞ no point in more disk=events, unless add CPU as well
 - ☞ but can easily accommodate larger event size (just \$)
 - Allows 5 users to go through 4 Mev each day
 - ☞ (if their code can do 300Hz)
 - more users ?
 - ☞ split (copy/serve) data to other CPU's
 - ☞ 40 "units" hold 160Mevents

The System

- **40 CPU's.** $40 \times 400 = 16\text{TB}$ disk
 - Maybe too much.
 - Should be able to stay with **10TB = 100Mevents = 1/5 full sample**
 - ☞ allows Pierre's estimate to be off by x5
 - So 250GB/CPU
 - At 300Hz one user stays on one CPU for 2.5 hours
- One “popular” data set
 - 80 people want it the same day
 - 16 cpu accessing the same data
 - ☞ 2MB/sec each job = 30MB/sec needed
 - ☞ assuming can not get more, faster analysis will be “downgraded” by I/O to run at 300Hz no matter what. OK

Proposal

- 10TB are divided among physics groups
- They define their 2nd datasets in such a way that
 - Fit on disk
 - Can be read at 300Hz
 - ☞ $50\text{KB} \times 300\text{Hz} = 15\text{MB/sec}$, in principle ROOT can bring all data to the analysis code that fast, not only a small branch
 - ☞ if 100Hz : 1day → 3days.
- If have more data:
 - Add disk
 - Or (since it wan't run faster) stage in/out
- How to connect the 40CPUs to the 10TB: rest of workshop
 - Of course one full blown 106 CPU SunFire 15K with 0.5TB RAM and 250TB direct attached disk would have been too easy
 - ☞ <http://www.sun.com/servers/highend/sunfire15k/details.html>

An example

- 5 8-ways, 2TB each, 40 user(queues) on each machine.
 - Some load balancing even with no disk sharing
 - Use our “standard 4TB disk units”
 - ☞ can add as much MC as data !
 - Could be local disk, FC, Yocum... Just \$ vs. convenience
 - ☞ In any case around or less then 500K\$
- Each node allows 40 users to go through up to 20Mev each day
- More data → use DIM to keep e.g. 50% of a data set on disk, then all jobs will start on disk files first, and then use staged in ones, but have to add significant tape b/w (of the order of 30MB/sec times the fraction of non-disk data)
 - Non shared disk can be NFS served to the “Kahuna host” (very little traffic there), tape daemon will run locally.
- If need >40 batch queues for one data set, either replicate (hopefully some other DS will be in less demand) or share disks, if share could go for 4-ways...

When ?

- This is needed “now”
- How risky is it anyhow ?
 - Can start with “local” disk, no sharing, even no FC
 - Can even stick to it forever
 - May never need tapes
 - ☞ or it can be spec'd that way.
- Of course the story is not finished, need a complete spec to begin with:
 - /cdf/scratch & /cdf/code
 - Tape drives
 - Data uploading

4 Questions

- Data format:
 - PADS: performance
 - Ntuple: bookkeeping (did my job run on all data ?), hard to handle even temporary disk overflow
- Scaling to Run2
 - >4x the data (at least, cleaner L3, more triggers..): 2000 Mevent
 - more users, more MC (maybe not much MC by next summer)
 - Will want/need one CPU/user. 200~300 CPU's
 - ☞ faster CPU's ? Better code ?
- The other 400million events
 - Not there (log at <30Hz) ? All punch-throughs ? The year later ? A bit at a time on fcdfsgi2 ? Mosix test ?
- PAD (re)creation

PAD (re)creation

- Primary DS are big (1e7-1e8-manye8 events)
- Data format may not be optimal for I/O
- May need pattern reconstruction
- 3 possibilities (at least), must allow for all
 - Slow code: 10Hz ? 1Hz ? → farm
 - ☞ farm I/O node saturated now
 - ☞ can't simply buy more nodes
 - Internal refreshing from info already on disk
 - ☞ Very fast. Do just “as if were one user job”
 - Fast task on bigger sample: $1e8 \text{ events} \times 50\text{Hz} = 10 \text{ CPUxdays}$
 - ☞ too much I/O for farm ? ($50\text{Hz} \times 100\text{KB} = 5\text{MB/sec} \times \text{DS\#}$)
 - ☞ 5 days with 6 fcd fsgi2 CPU's (plus one for staging ?) $\times \text{DS\#}$

Summary

- Need hardware much before summer
- Can do a lot with disk resident data with realistic hw config
- Wan't easily scale to full Run2 though
- To make possibility into reality
 - Need clear definition of strategy and data set sizes
 - Users must know what to expect and prepare for it, not make noise asking for more/different
 - Need fast user analysis code: $>100\text{Hz}$ through user's analysis/skim
 - ☞ I/O must be $O(1\text{msec})/\text{event}$
 - ☞ code must be efficient $(p1+p2).M()$????
 - ☞ Even if we have to write it all in C - -
 - Do not even think of planning on slower stuff