Computing Usage Task Force (CUTF) appointed by spokespeople in May 2004

Stefano Belforte (chair)

Patrizia Azzi (top)

Song Min Wang (exotics)

Ray Culbertson (exotics)

Ashutosh Kotwal (EWK and offline)

Matt Herndon (bottom and tracking)

Konstantin Anikeev (bottom)

Elliot Lipeles (CAF)

Igor Sfiligoi (CAF)

Charge (shortened)

- This TF is charged with reporting on the current degree of optimization in using CDF computing resources for analysis and on how to improve it.
  - Needed monitoring first
- The TF will take into account resources both at FNAL and in the CDF-Grid and will develop and deploy monitoring and accounting tools as needed to study the situation.
  - We will see results today
- An additional product of this TF will be the accounting of usage of computing resources off site, as requested by CDF International Finance Committee.
  - Tool and strategy identified, implementation in progress

CUTF work schedule

- June = review of status talking with physics groups
- July = setup monitor of CAF to verify/quantify
  - Modify CAF framework w/o disrupt operations
- August = collect data and ideas
  - More ideas about how/what to monitor came while monitoring
- September = rethink critically June's understanding in light of monitor information and write report
  - Delayed by "back-to-school" workload and Stefano's health troubles
  - Have most report sections and materials in hand, needs one solid week of good editing

- Final report will not have significant news with respect to today's talk, only more detail and easier reading

CUTF situation assessment

- Thanks to delayed DAQ upgrade, CPU was not missed

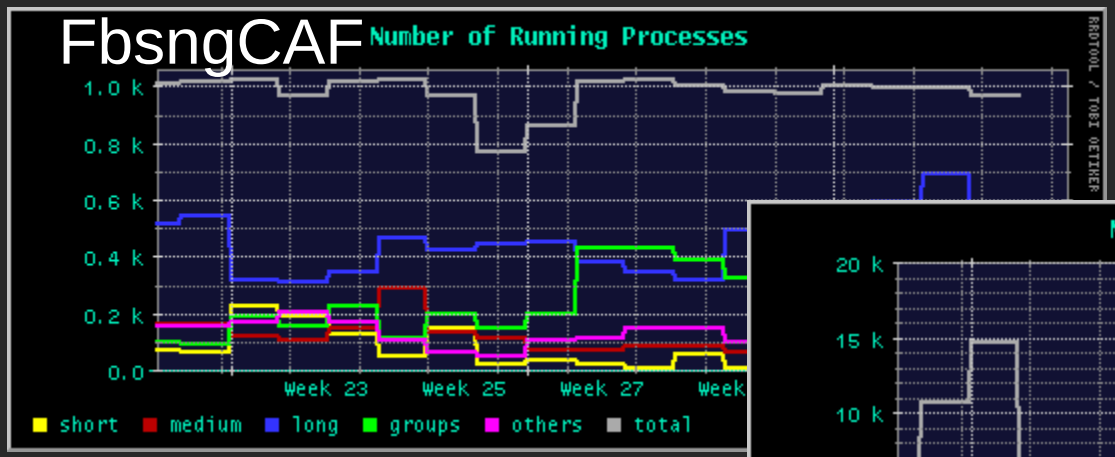
- Much more user MC done/needed than planned
- QCD/EWK already doing almost all computing on ntuple
  - A lot of work done on private systems, e.g. all W-mass on Duke's trailers mini-cluster
- Top/Exotics do similarly, but also do a lot of MC and run a lot on data to develop good algorithms (e.g. jet b-tag)
- B analysis are enormously CPU consuming due to large variety of analysis channels and topics and combinatorial load in secondary/tertiary vertex finding (4-track vertexes e.g.)

- Many users doing many different things. No way to isolate typical cases and attack them. **Very difficult to optimize**.

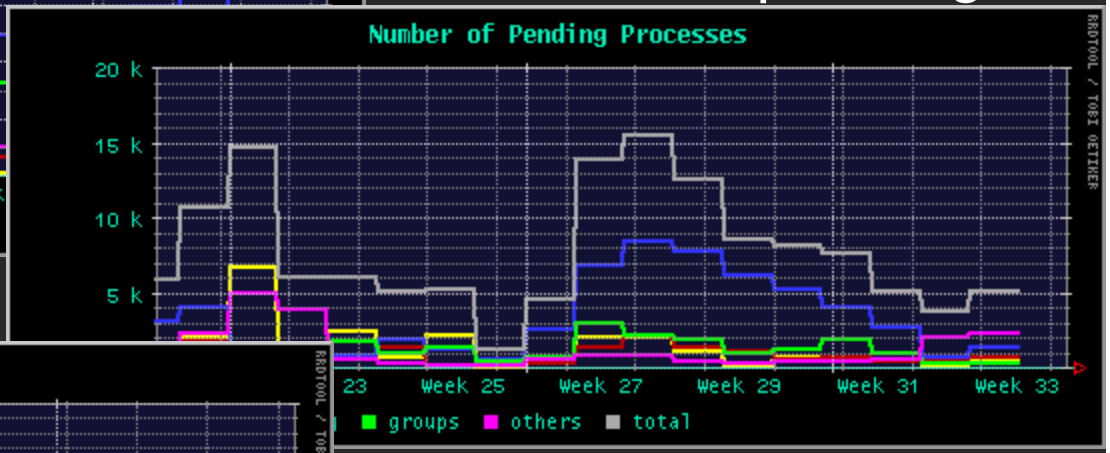
- Users are using CPU to try, test, explore, learn, do physics

FNAL resources always saturated (May-Aug)

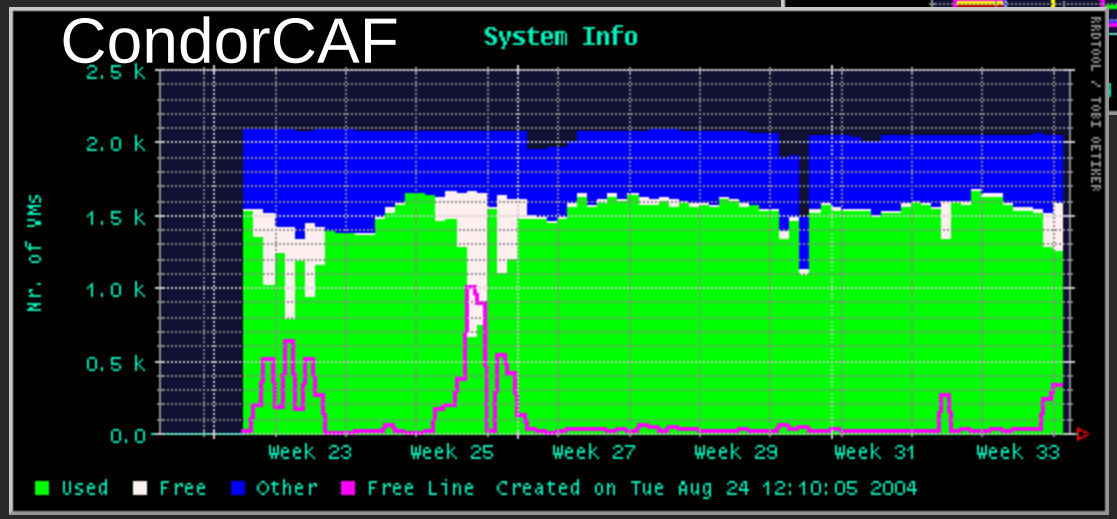
FbsngCAF

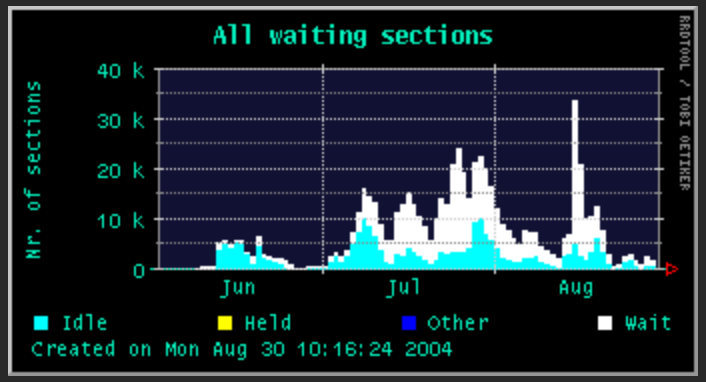


CAF:
Always running
Thousands pending



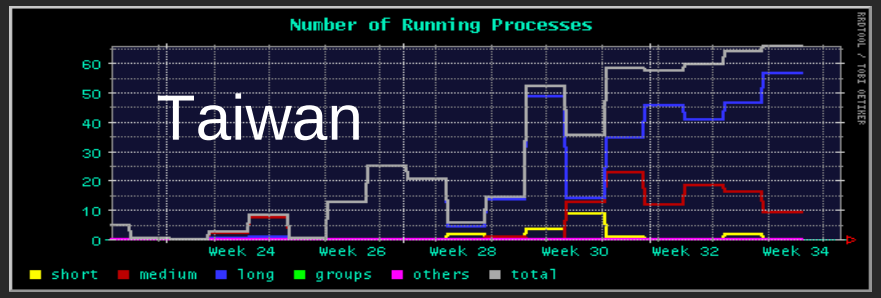
CondorCAF

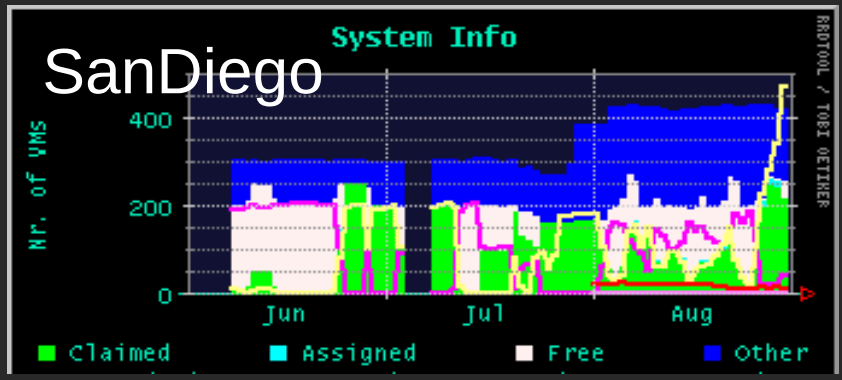


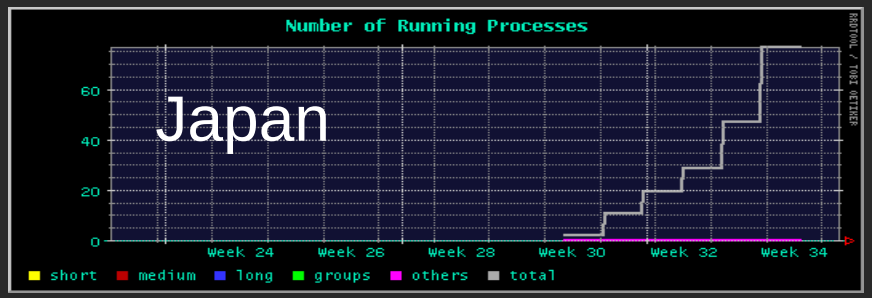


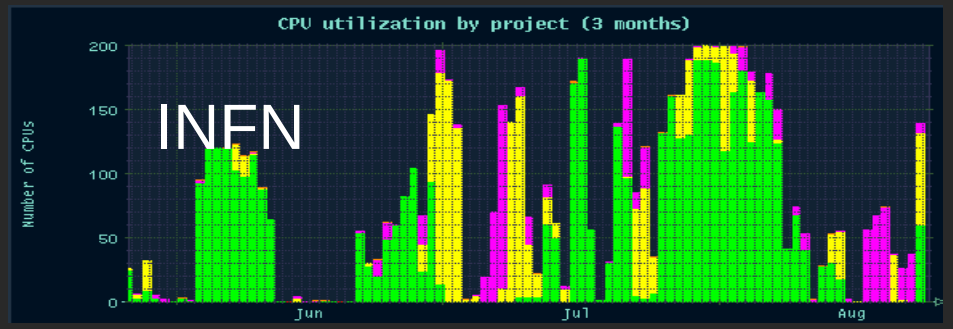
Offsite usage ramping up (May-Aug)

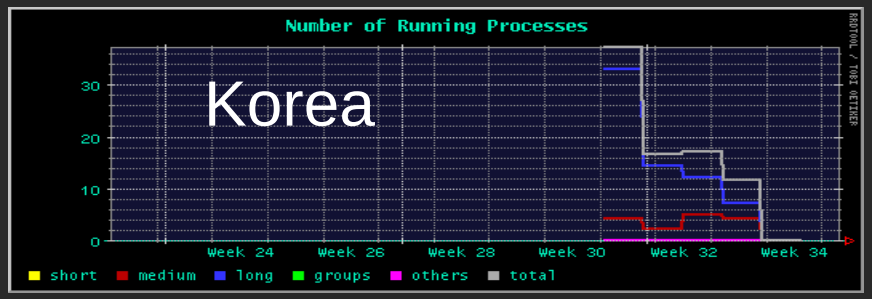
- Yes, we need a uniform, better, monitoring ! So limited the study to FNAL's CAF

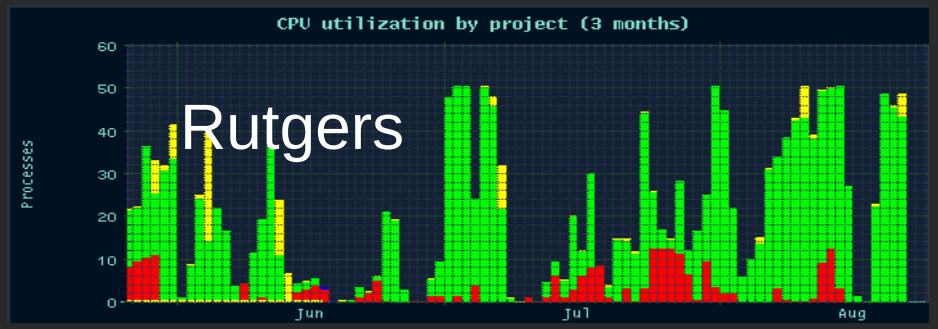












CUTF hard task

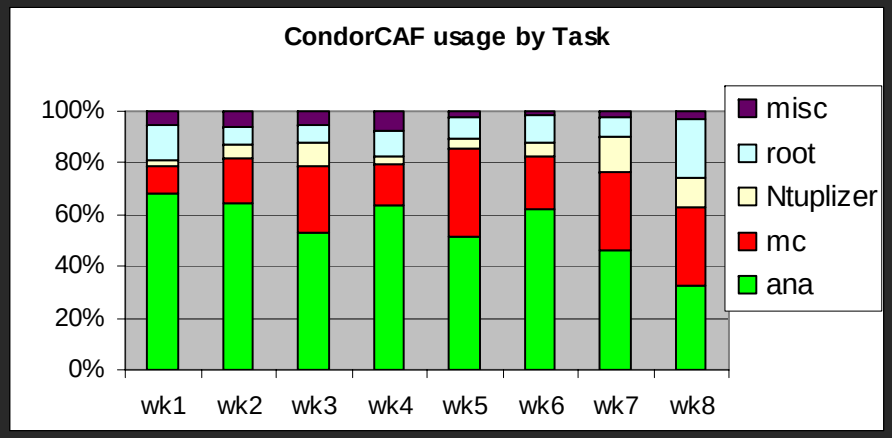
- Try to learn from monitor about analysis jobs
- What really people use the CAF's for ?
- What datasets do they use ?

- How fast are our jobs ?
- Could we make them faster ?
- How ?

- It proved to be very difficult
  - 800 users requested a CAF queue
  - 50~60 running at any time
  - But not the same ones !!

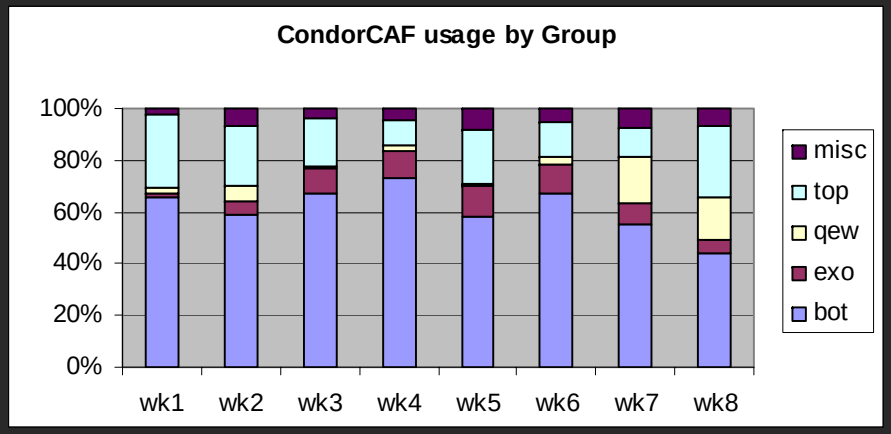
Who is using that CPU ? Jul/Aug on CondorCAF

Lots of MC, Lots of B



- Up to 50% is MC (root=MC)
  - this is a lot of MC !! All organized MC production uses less

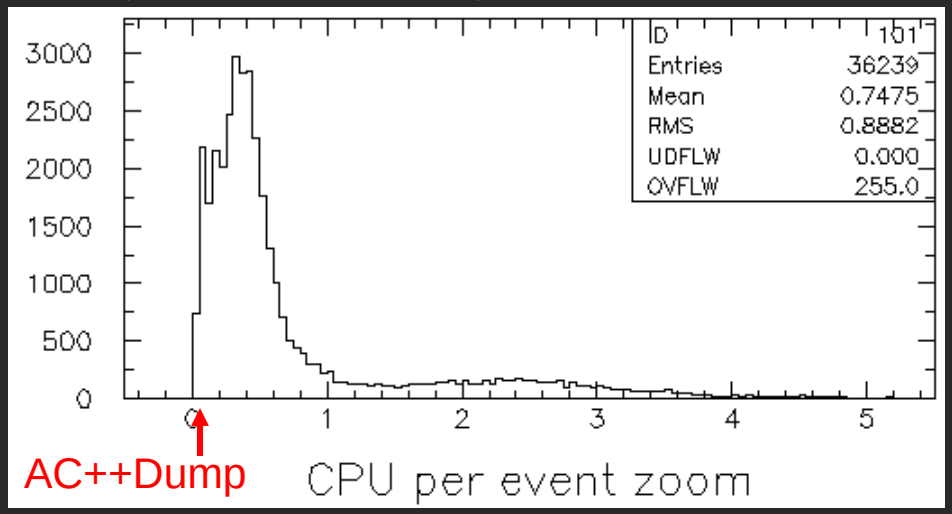
Now → → → 2 months ago



- B rules
- qew = QCD + EWK is a minor customer
- what is all that activity by top/Exo ?
- it is not making common ntuples
- turns out to be MC and developing top b-tag

Last 8 month of CAF (non Condor)

- 240K cpu-days available, 80K jobs "logged"
- Have input-dataset/cpu-time info for 36K jobs: 56K cpu-days, ~25%

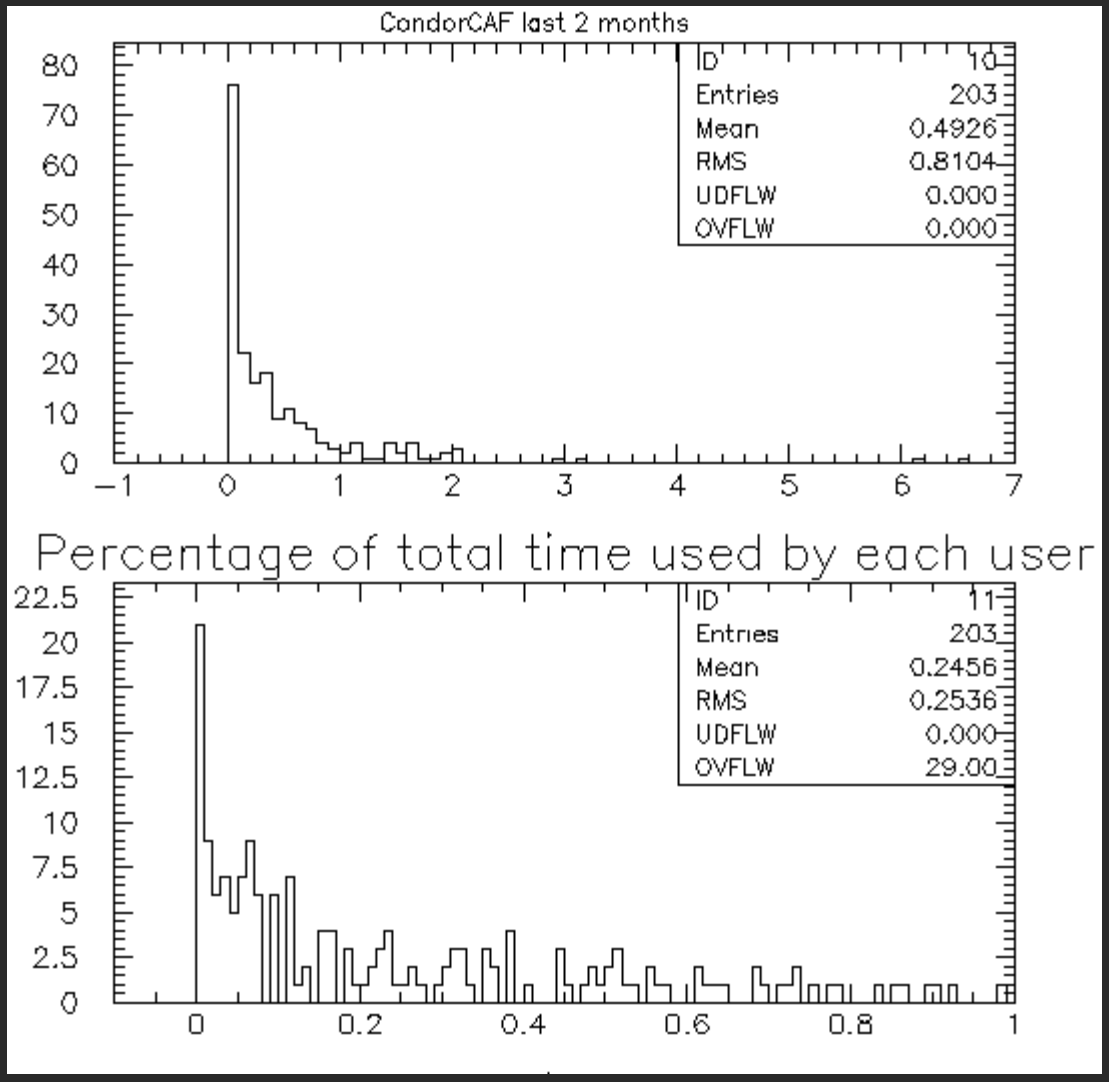


AC++Dump sits at ~0.06 sec/event

CPU usage dominated by users code, not framework

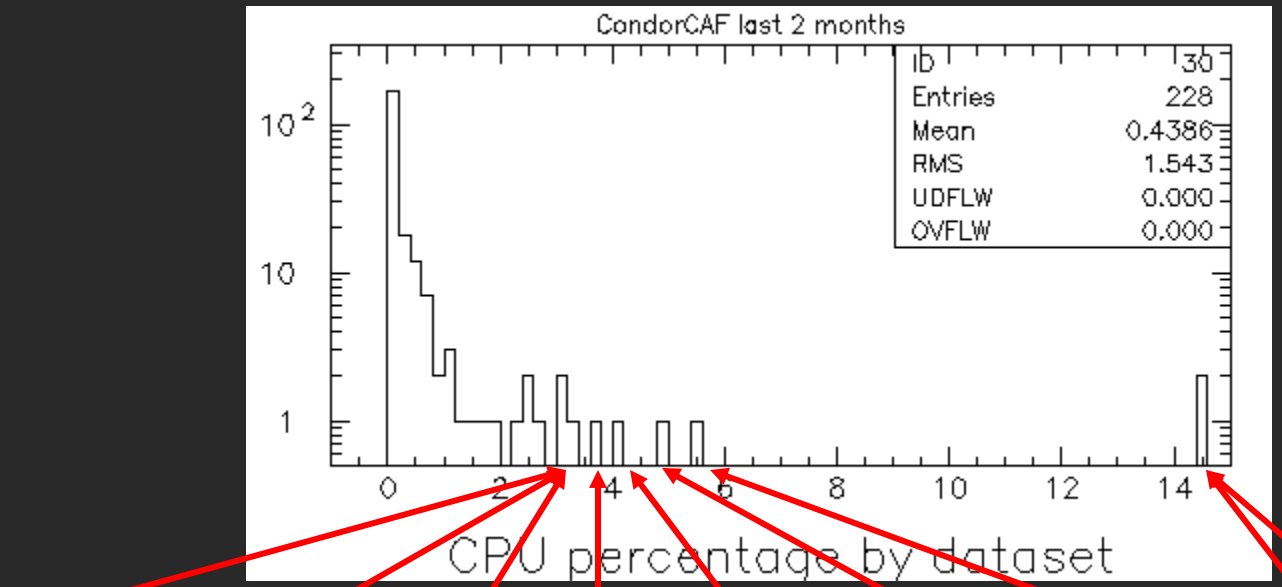
- 7K jobs use >1 sec/event
  - 400 of them account for 40% of total cpu: 23K cpu-days. 18K (30%) hadronic B + J/Psi (CVTMFT?). Can help here but saving O(10% total)
- 30K jobs use <1sec/event, each job minuscule fraction of total
  - Account for ~60% of "analysis CPU"
  - Not I/O limited, user code is the "culprit"
  - could not find a way to summarize/break-down them
  - No idea how to tackle them "C++-wise"

CPU By user



- 203 users on CondorCAF in last 2 months
- Not as simple as a few users using all CPU (we could learn a lot about patterns and needs talking to them)

CPU by dataset



xpmm0c xbhd0e jpmm0d gjt10d xbmu0d xbel0d xpmm0d xbhdod xbhdoc

- 228 cdf dataset names "hit" in last 2 months on CondorCAF
- Usual suspects show up, red names makes up ~60% of total
- 30% of usage for the big inclusive B-Charm xbhd0d + xbhd0c
- 40% is scattered in >200 dataset names

One clear thing

- The model we have been using to estimate CPU needs does not fit well the patterns we see
  - There is much more MC
  - There is much more usage of common Ntuple
  - There is much more slow analysis of large datasets

- Nevertheless, the number of computers we build fit nicely the need, we use them all

How to optimize ?

- Most usage is by many single users doing many different things and spending long time at them
- Anecdotal evidence for a lot of learning by trial and error
  - impossible to quantify
  - difficult to cure "from the top"

- So it is a complex scenario, how do we put order and method into it in order to "optimize" it ?

  - Put pressure from the top (offline management) and let the system optimize itself
    "market economy": make waste cost

CUTF suggested path to efficiency improvement

- Move more work from scattered users effort to planned effort
  - More validation, more commonality, more accessibility, more efficient usage of remote farms, more planning capability.\
  - Give a lot of resources to planned jobs
  - Squeeze random user work to force optimization by necessity:
    - make waste expensive for user

- Move "tasks" from user's executable to production (cosmic ray, dedx, beam constraint): need validation
- Need time and effort from physicists not computing professionals

- A few places where code can be sped up have been singled out, and addressed, but even a factor 2 here buys ~10% of total

For example…

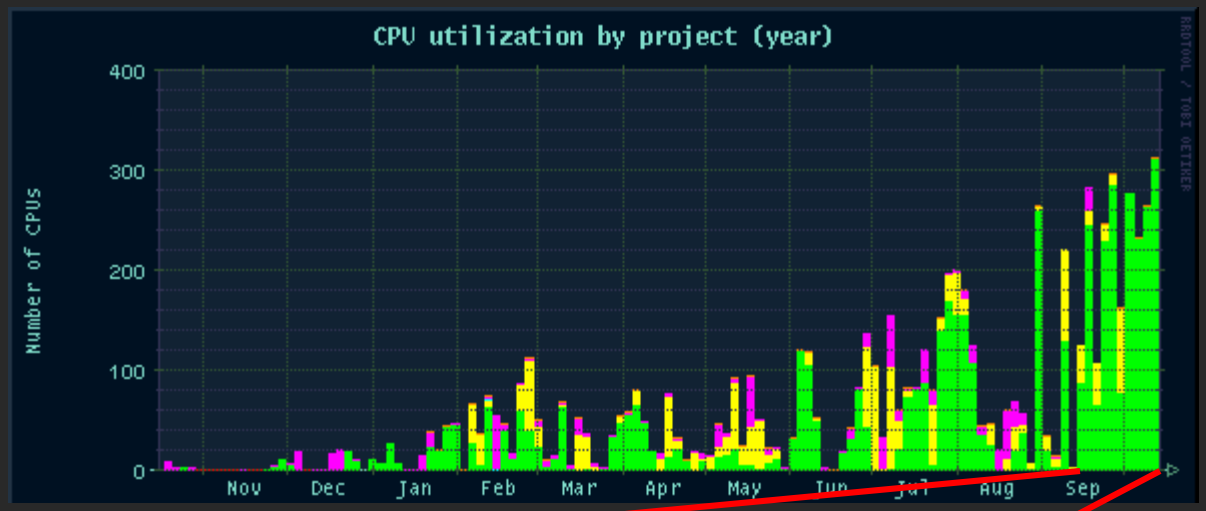
- Do more at production times (cosmic filter, beam constrain…)
- Make official ntuples early in production (encourage sharing)
- Speed up CVTMFT (constrained vertex fitting)
- These are all being tackled now, cosmic filter in production already, recipies for faster CVTMFT and for using complier optimization are on the web

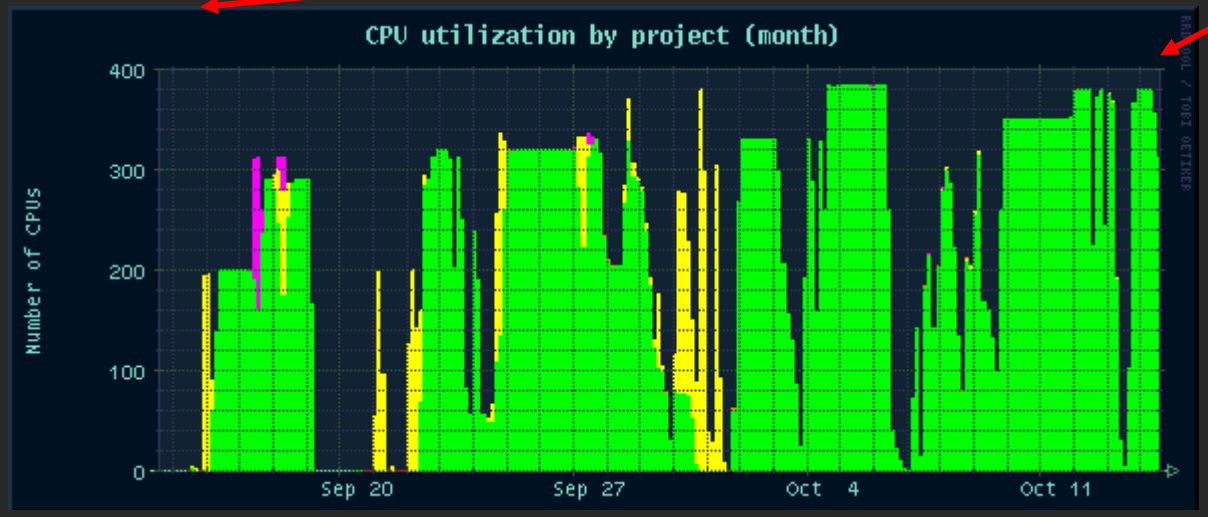
- B group has to reach "organization" level of Top/Exo/EWK/QCD
  - Ntuples, Common MC, Smaller samples (skimming large B samples is a gigantic work that has scared most away, INFN now developing tools to make it "easy").
  - All requires lots of physicist time and effort, but are in progress
  - Also B group has more people, so more usage and more difficulty in reaching internal agreement and organization. Imposing that from top would not work, it has to grow from experience and B group does not have "the Run 1a example" to build upon

Spares

Italian analysis farm (CNAF) during 2004

- Hardware grew from 50 to 500 GHz from January to September
- Users "noticed"
- Last month:
  - ~80%
- >70% for MC
  - Data 6TB only





CDF GRID October 2004

http://cdfkits.fnal.gov/grid/

Total cpu. Locals get priority in some sites ➔ "CDF" gets less

Farm	Site	GHz	TB
CAF	FNAL	1200	300
CondorCAF	FNAL	2000	Shared with CAF
CNAFCAF	Bologna - I	500	7.5
KORCAF	Korea	120	0.6
ASCAF	Taiwan	134	3.0
SDSC	San Diego	280	4.0
HEXCAF	Rutgers	100	4.0
TORCAF	Toronto	576	10
MITCAF	Boston	110	2
JPCAF	Japan	152	5.0
CANCAF	Cantabria	52	1.5
TOTAL AT FNAL		2200	300
TOTAL OFFSITE		2024	37.5