Panda monitoring of distributed analysis

Sergey Panitkin

BNL

ATLAS

BROOKHAVEN NATIONAL LABORATORY

Outline

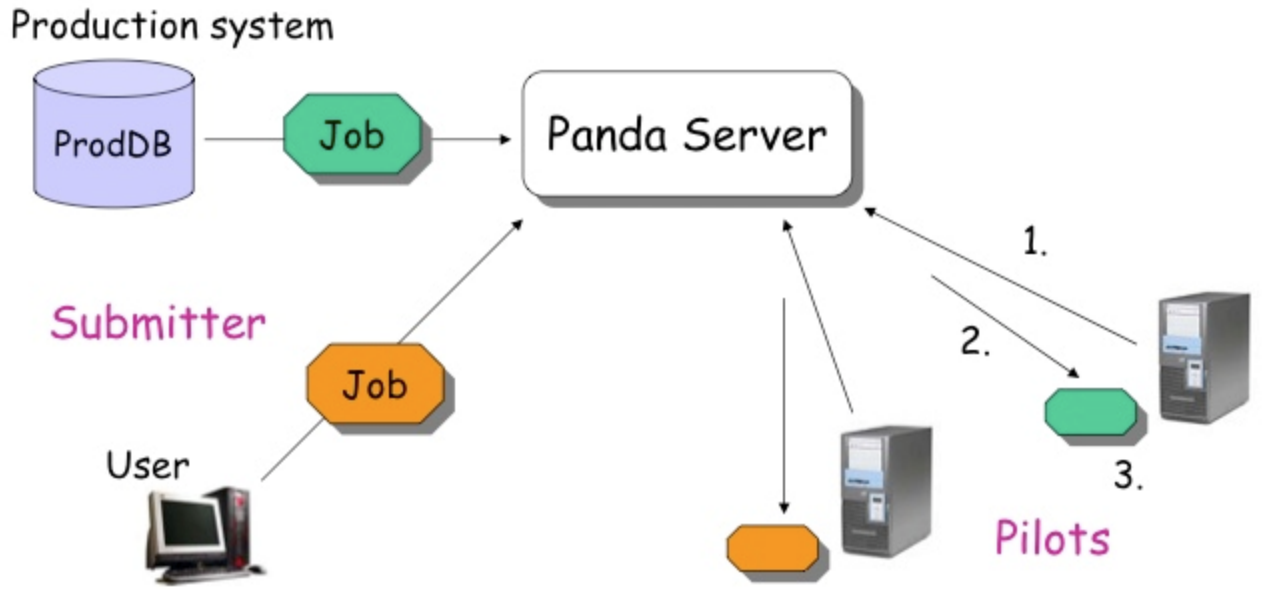
- Introduction
- Pilot timing
- Analysis patterns
- Summary

Many thanks for help, collaboration and encouragement to Kaushik De, Tadashi Maeno, Aaron Thor and Torre Wenaus !

Panda monitoring

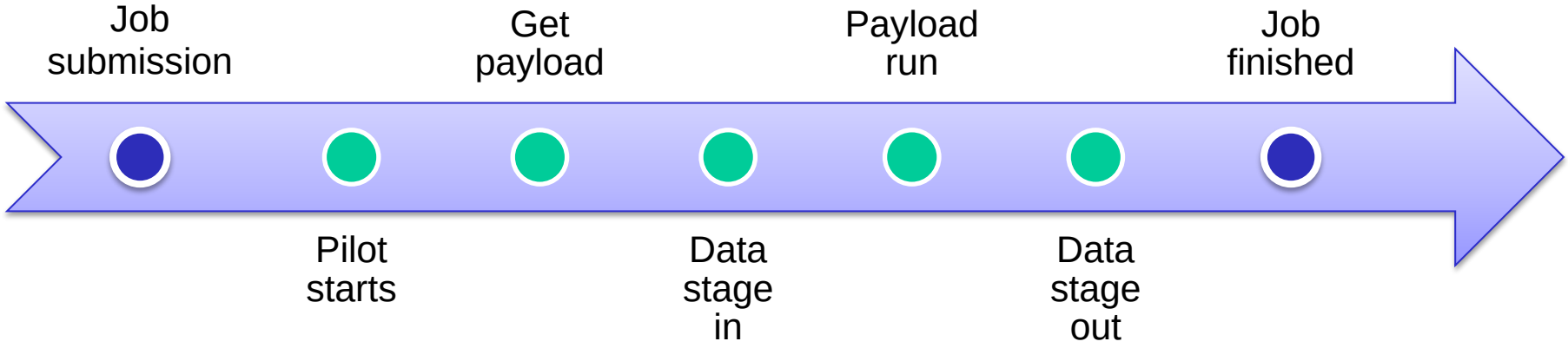
- Panda server collects a variety of information about jobs running on the Atlas grid
- Panda monitor provides a web interface to the Panda system: datablocks, jobs, sites, system component logs, job status logs, production statistics, etc.
- Often, for the sake of brevity, this information is presented in averaged fashion, averaged over time, sites, clouds, etc
- Some relevant information is buried in log files
- Hence, it's often difficult to get detailed understanding of analysis activity
- The goal of the current project was to study Panda pilot timing on a sub-job level in order to understand details of the analysis activities on the grid
- Server side tools that allow to query detailed pilot information were developed by the Panda team (A.Thor).
- We looked at "regular weeks", when no stress tests or Jamborees were taking place. "Non invasive" monitoring.
- What is "normal" analysis pattern?

Job Flow in Panda



Each pilot runs on a worker node
1. send a request
2. receives a job
3. runs the job

Panda job (and pilot) timeline

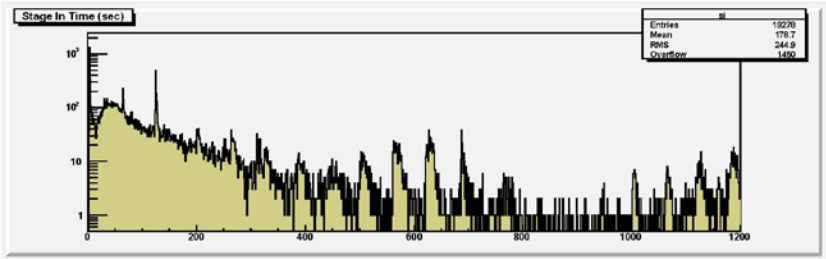


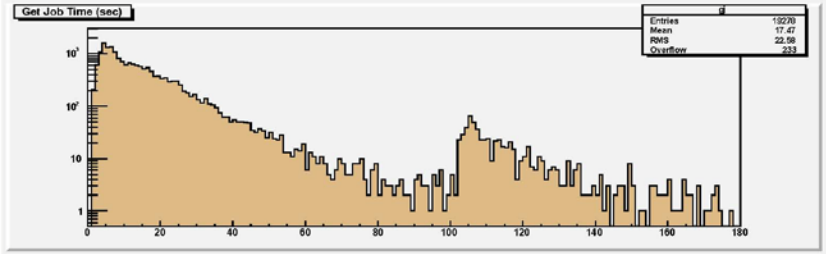
A few definitions:

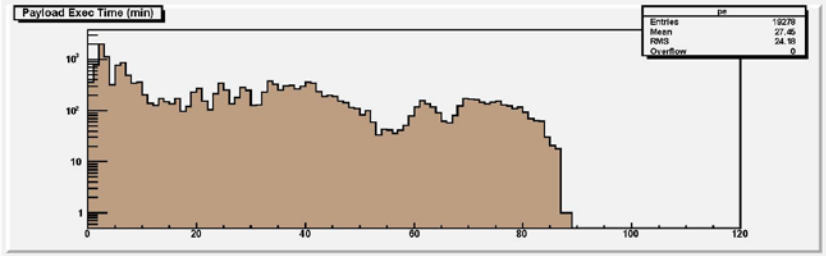
- Wall time = “Job is Finished” time – Job Submission time
- W.T. determines user experience. You job is not finished until the last sub-job is finished.
- Wait time = Pilot start time – Job submission time
- Payload run time = Athena job running time
- Pilot run time = Pilot ends – Pilot starts times

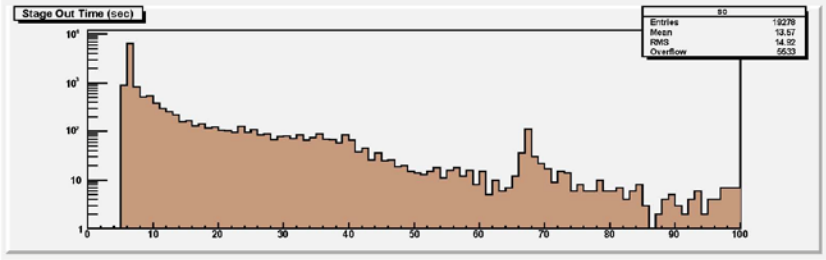
Only finished jobs were considered for analysis

Pilot timing information









Log scale to emphasize tails

Analysis queue BNL, one week, ~19k jobs

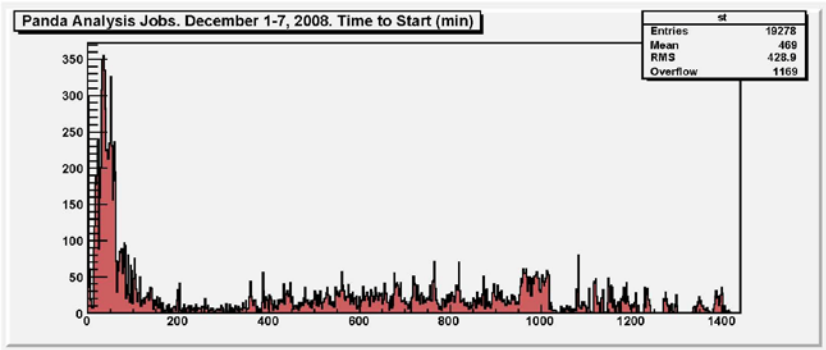
"Data stage in" time: average ~179 s.
- main peaks at 60 and 120s
- occasional retries every minute

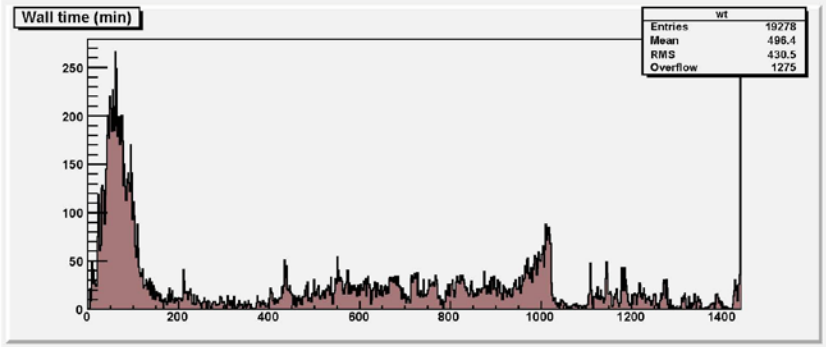
"Get payload" time : average ~17s
- main peak below 10 s
- secondary peak at about 100 s

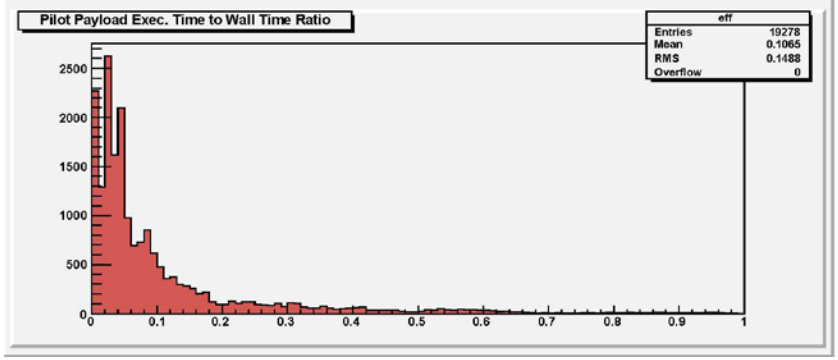
"Payload run" time ~27 min
- main peak below 10 minutes

"Data stage out" time ~ 14 s

Job timing and efficiency







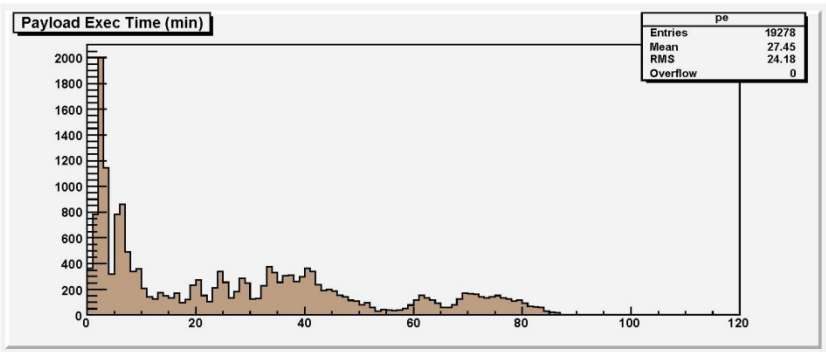
Analysis queue at BNL, one week, ~19k jobs

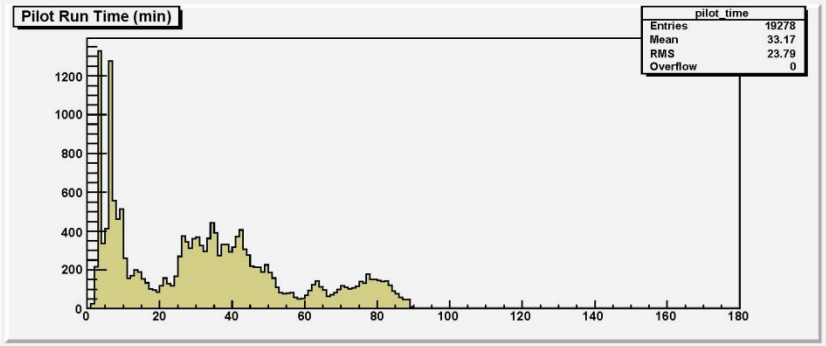
Time to start: avarage ~469 min
peak at ~50 min
long tail

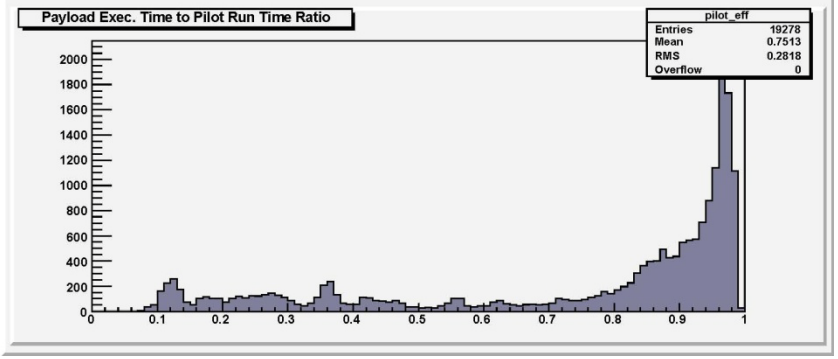
Wall time: average ~ 496 min
peak at ~60
Wall time is dominated by the wait time!

Payload run time to wall time ratio, aka "Job efficiency": average ~10%

Pilot efficiency







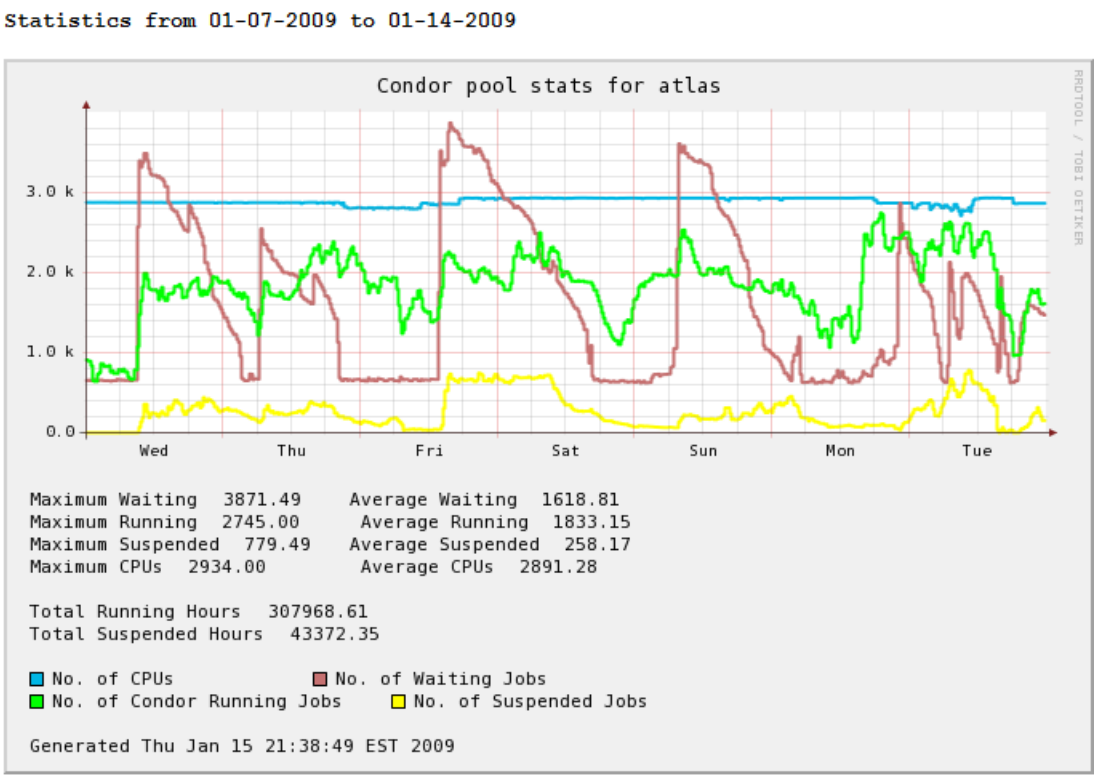
Analysis queue at BNL, one week, ~19k jobs

Payload run time: average ~ 27 min

Pilot run time: average ~33 min
peaks at 5, 35 and 80 minutes

Payload runtime to pilot run time ratio, aka “Pilot efficiency” ~ 75%, peak at 97%
Large tail at low efficiency

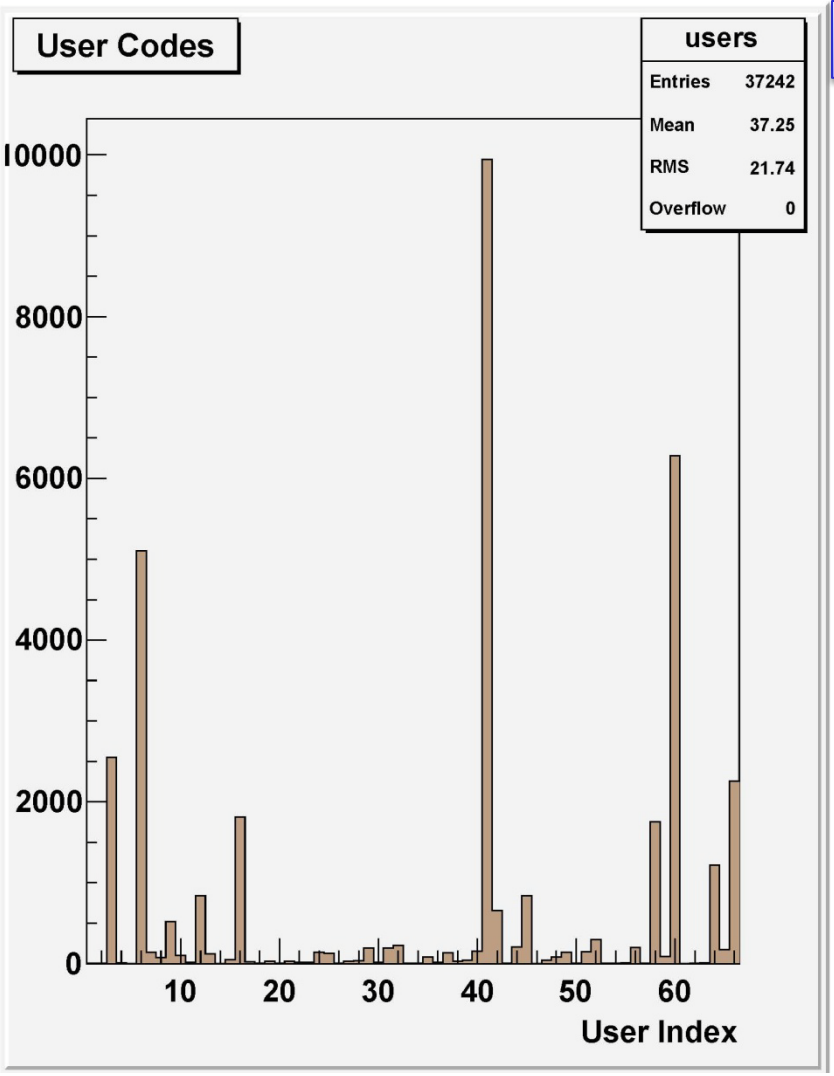
Condor queue usage at BNL



Condor monitoring at BNL

- Job submission pattern at BNL
- Both production and analysis jobs
- Grid analysis queues at BNL ~800 slots
- Short queue ~400 slots

Usage pattern study

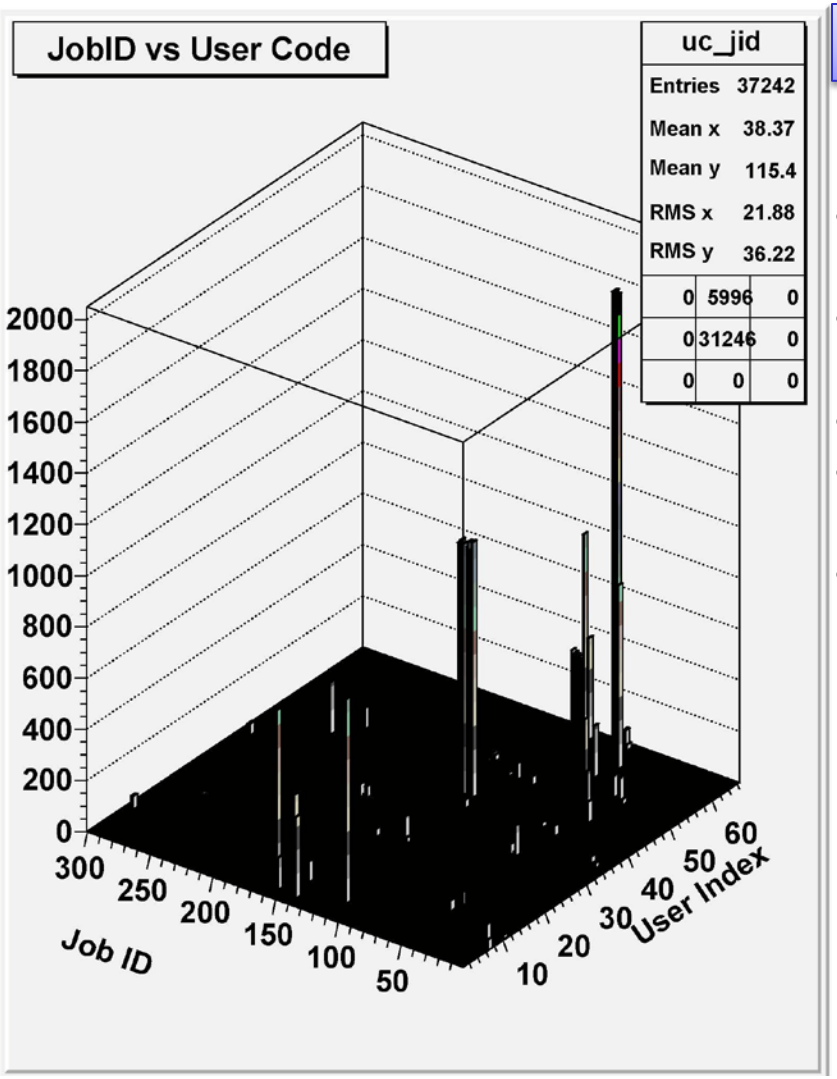


•ANALY_BNL_ATLAS_1 queue, one week, 37k jobs

•68 users during that week
•~7 users submitted jobs with more than 1k sub-jobs
•They submitted ~ 50% of sub-jobs
•User #41 submitted about factor of 2 more sub-jobs
•then the next two “power users” - #6 and #60
•Majority of the users submitted smaller jobs
•User #41 was from France
•Users #6 and #60 were from US

•What was the submission pattern?
•Was it one huge job or several small ones?
•How many sub-jobs per job were submitted?

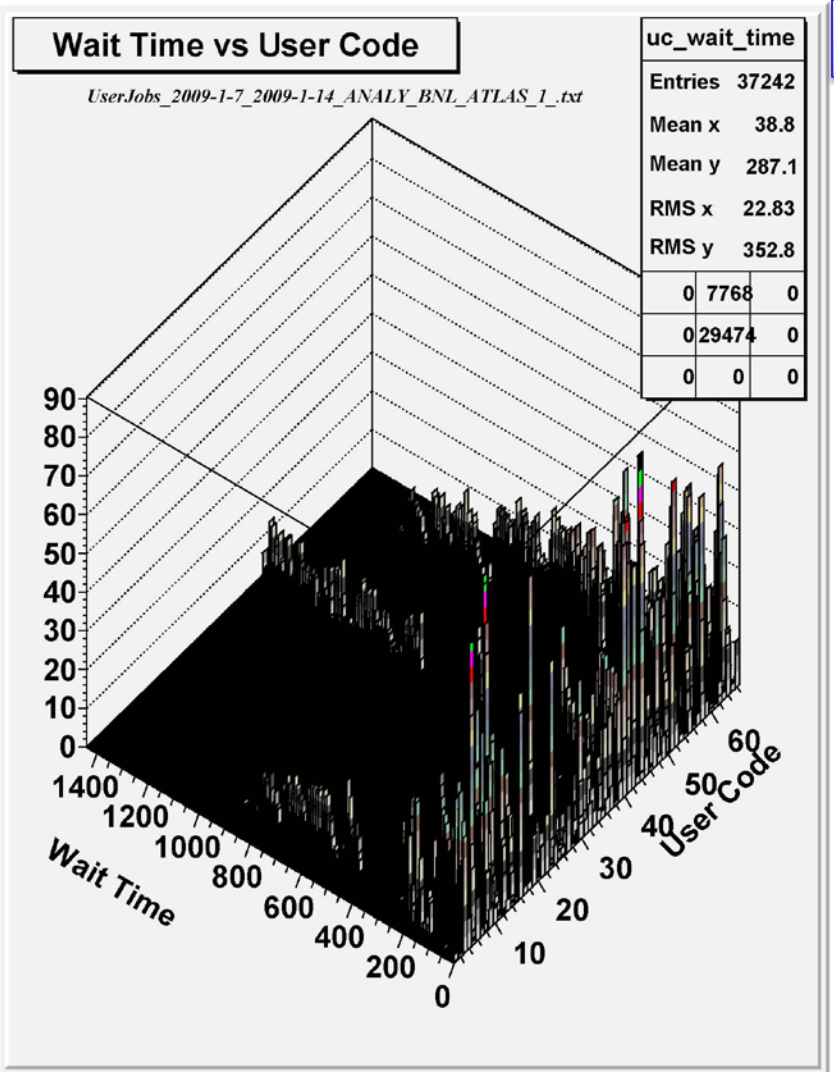
Usage pattern study



- ANALY_BNL_ATLAS_1 queue, one week, 37k jobs

- This plot tries to answer a question about submission pattern.
- The largest single job was submitted by the user #61 ~2k sub-jobs
- User #40 submitted several jobs with 1k sub-jobs
- Most “power users” are expert users, with more than 100 Panda submissions under the belt
- There were also a few novices users running

Usage pattern study



•ANALY_BNL_ATLAS_1 queue, one week, 37k jobs

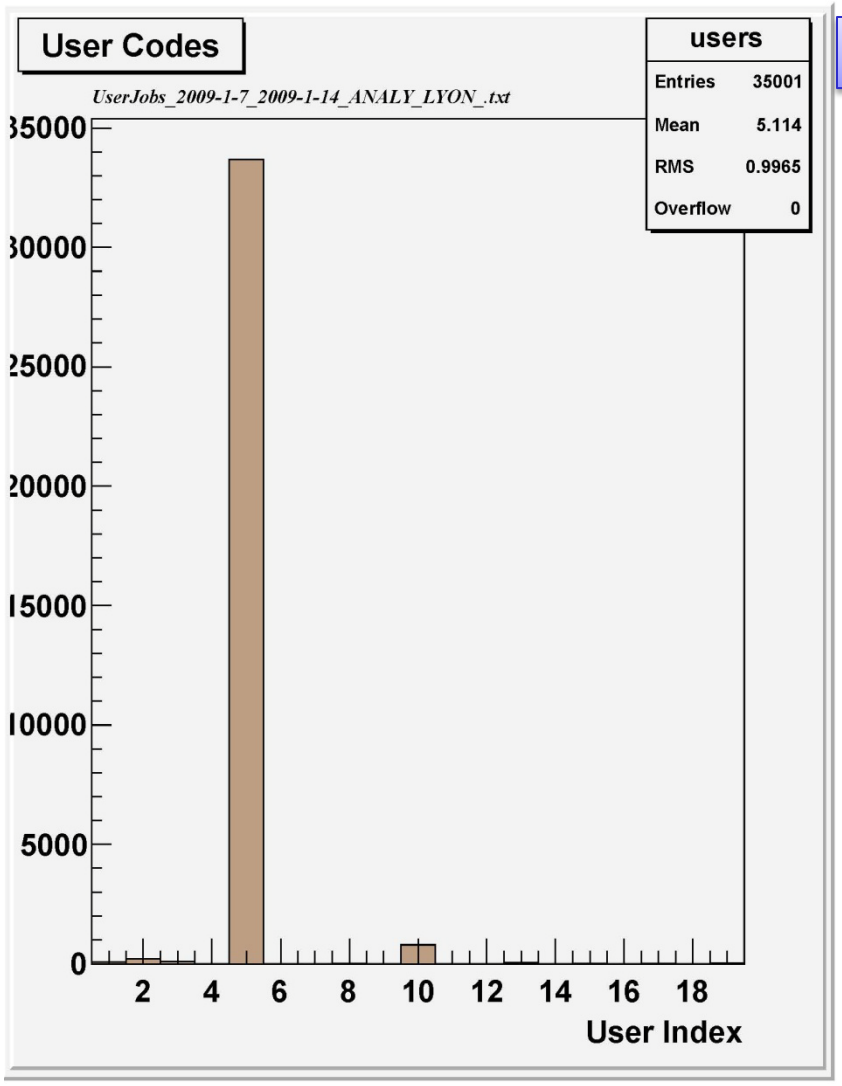
•What happens when you submit a job with 1k sub-jobs to a queue with ~400 slots?
•You'll have to wait!
•Panda has a fair-share mechanism in order to prevent "resource hogging".

$$Priority(n) = 1000 + F - \frac{T+n}{5} - W * (U-Q) * H(x=(U-Q))$$

•Some jobs submitted by the "power users" had to wait for more than 24 hours to begin running.
•Large job submission does not look like an efficient way to analyze data– does it?

• Why do users do it?
•Because they can?
•Because their datasets are big?

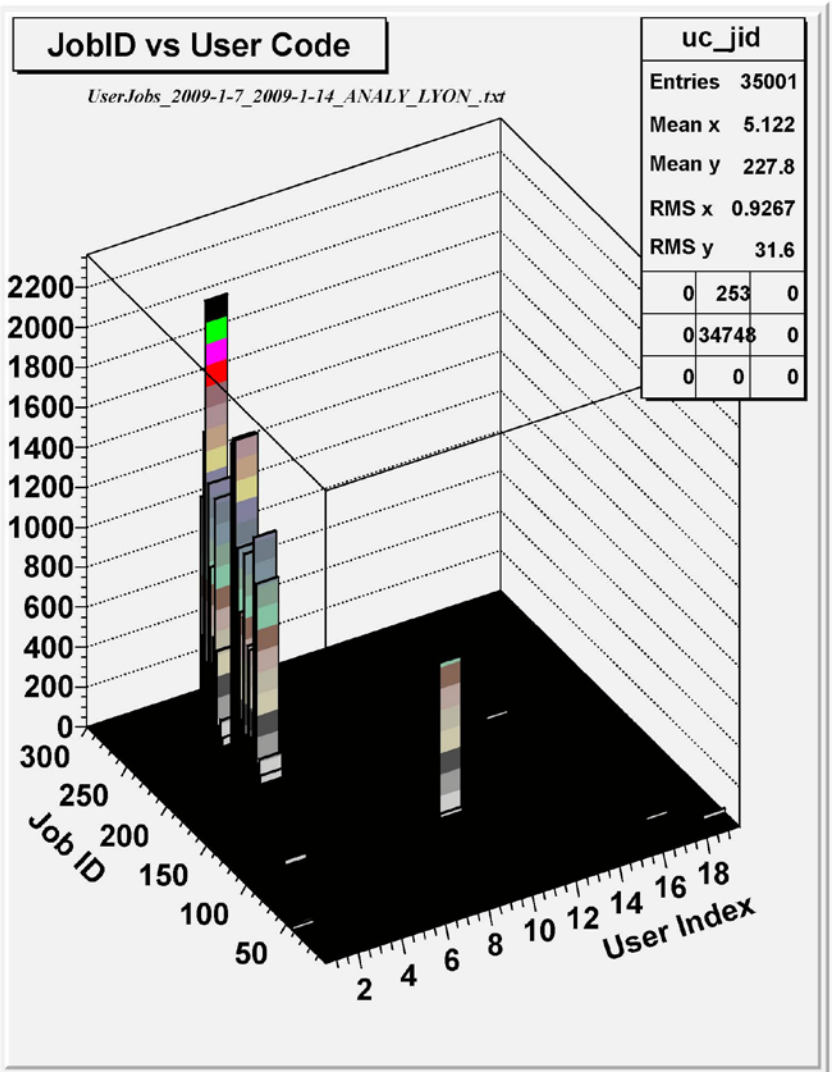
Usage pattern study



ANALY_LYON queue, one week, ~35k jobs

- About 20 users during that week
- User #5 dominated job submission with more than 30k finished.
- What was the submission pattern?

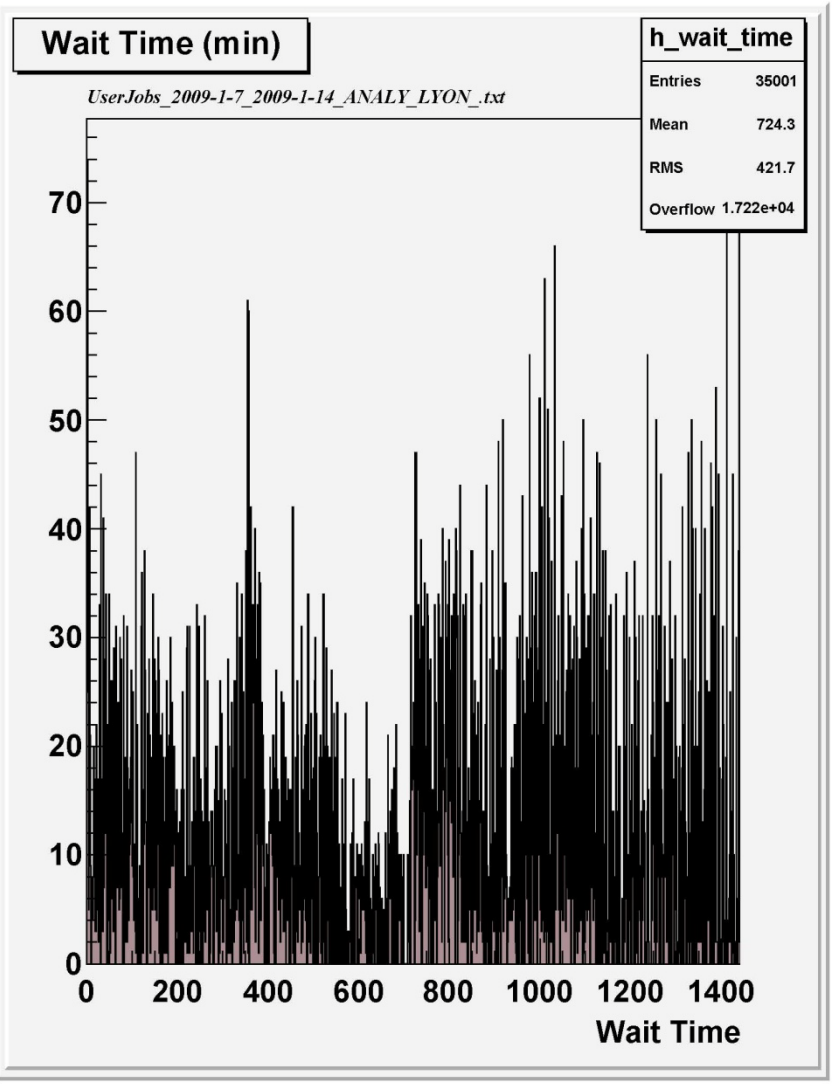
Usage pattern study



ANALY_LYON queue, one week, ~35k jobs

- User #5 submitted several jobs with ~1k sub-jobs each
- User #5 is “experienced”.
- User #5 is from France

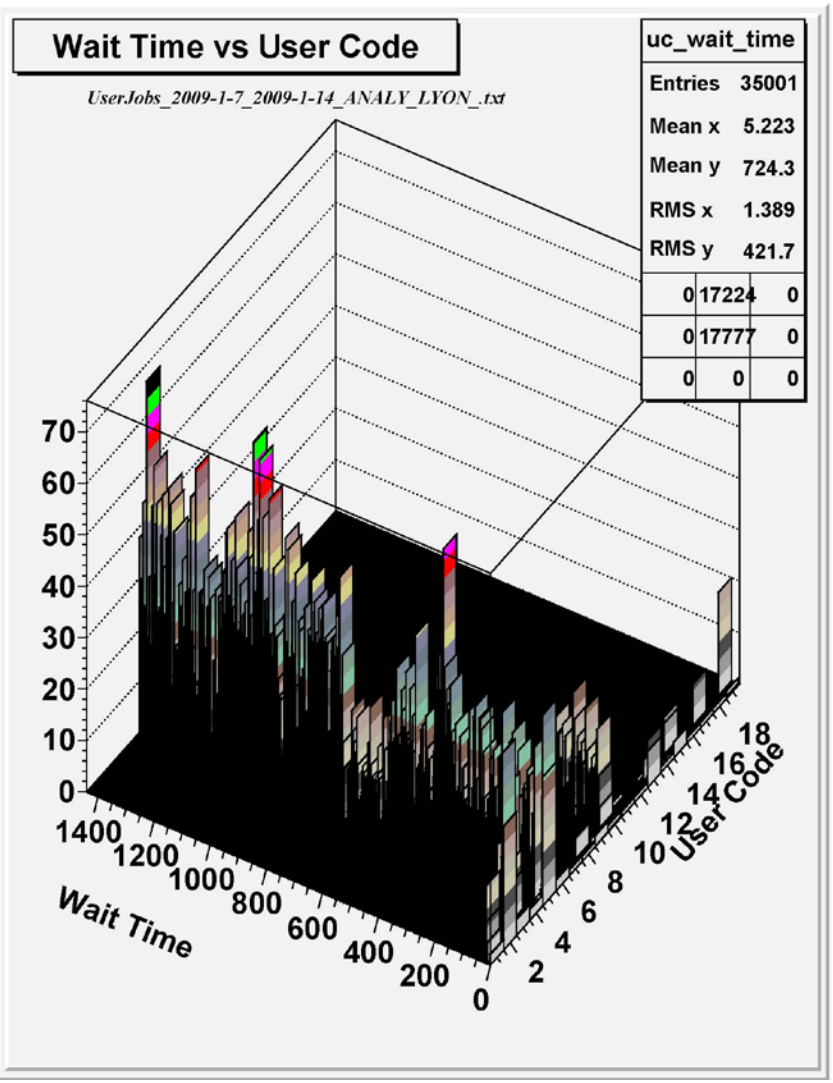
Usage pattern study



ANALY_LYON queue, one week, ~35k jobs

•Wait time for the Leon analysis queue during that week.

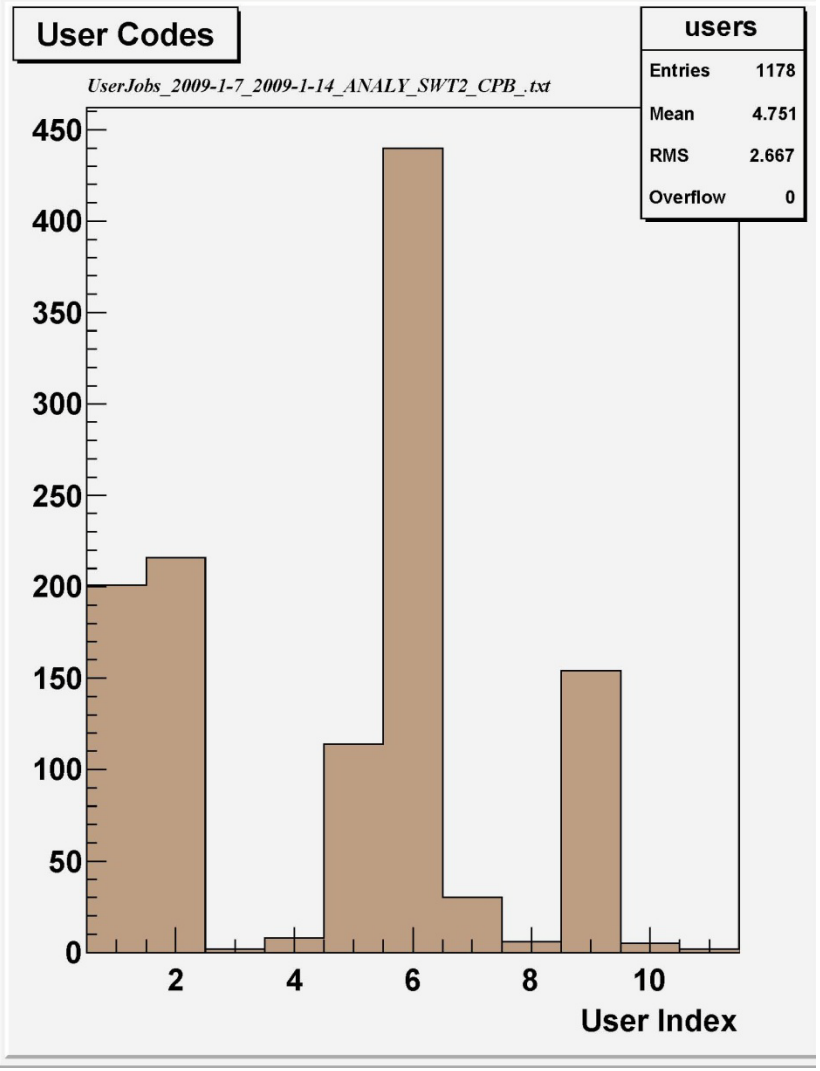
Usage pattern study



ANALY_LYON queue, one week, ~35k jobs

- User #5 jobs were “fighting themselves”
- For some jobs wait time was longer than 24 hours
- Note number of entries in the overflow bin.
- Wait for other users was much shorter

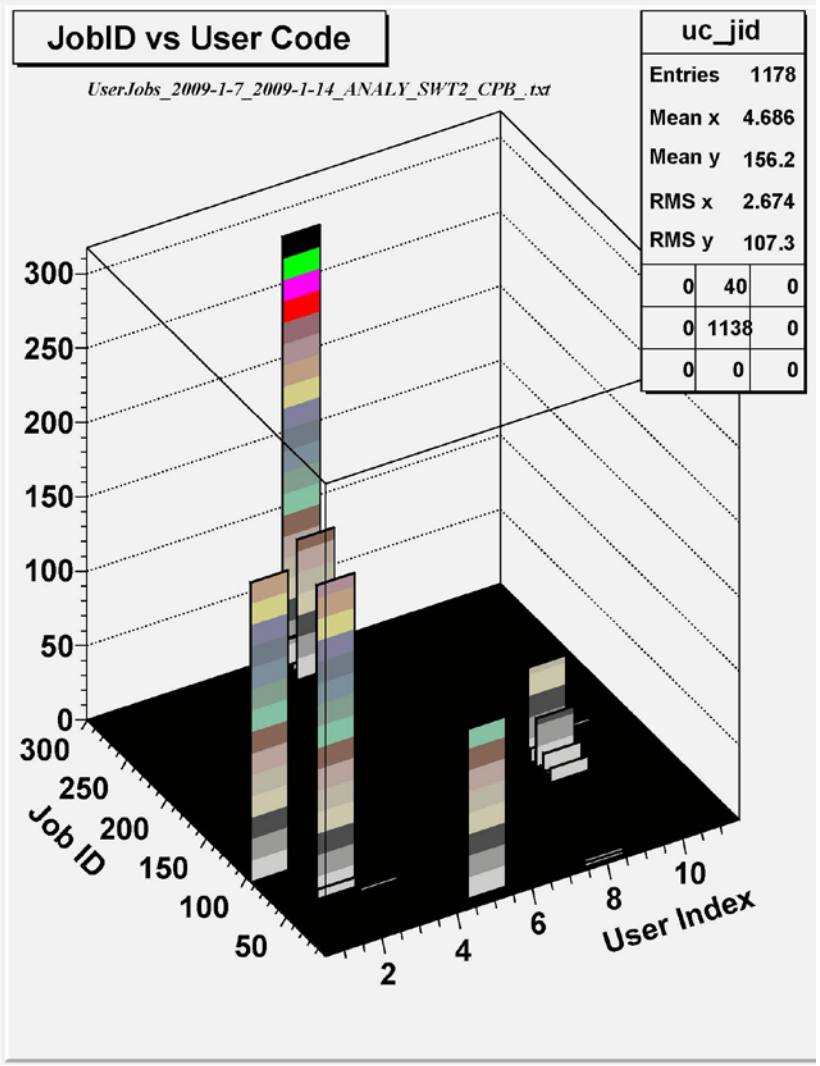
Usage pattern study



ANALY_SWT2_CPB queue, one week, ~1k jobs

- About 12 users during that week
- Most of them from Europe
- User #6 submitted majority of jobs

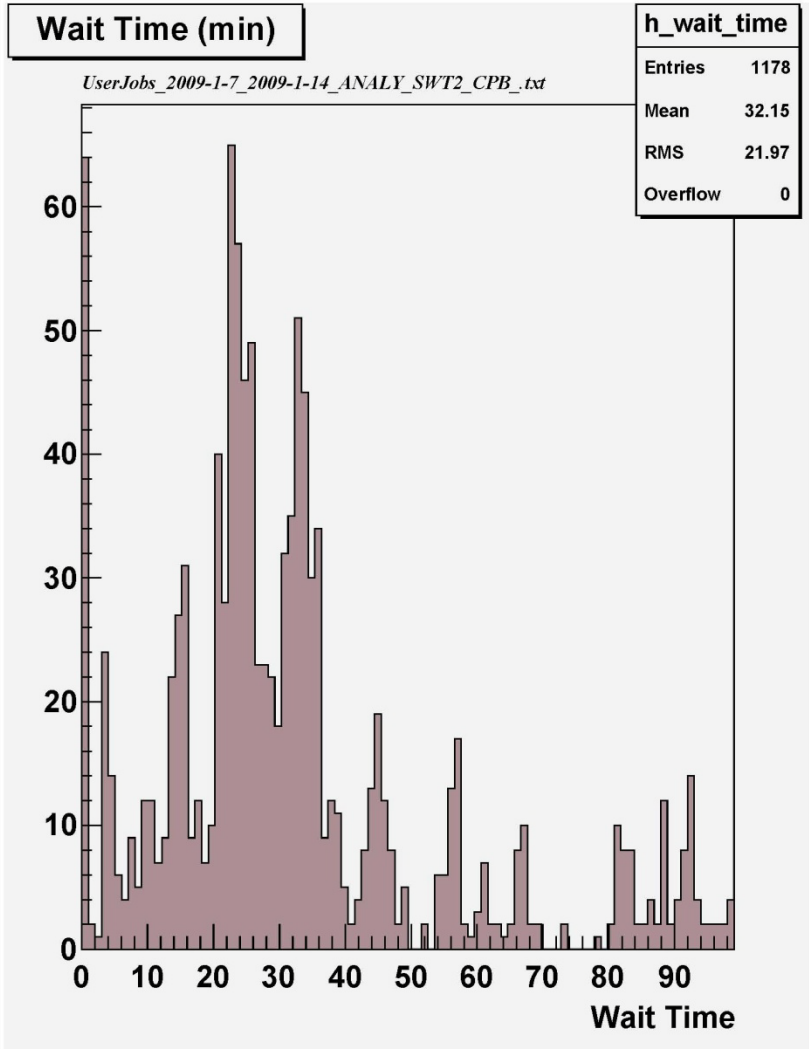
Usage pattern study



ANALY_SWT2_CPB queue, one week, ~1k jobs

- Job submission pattern

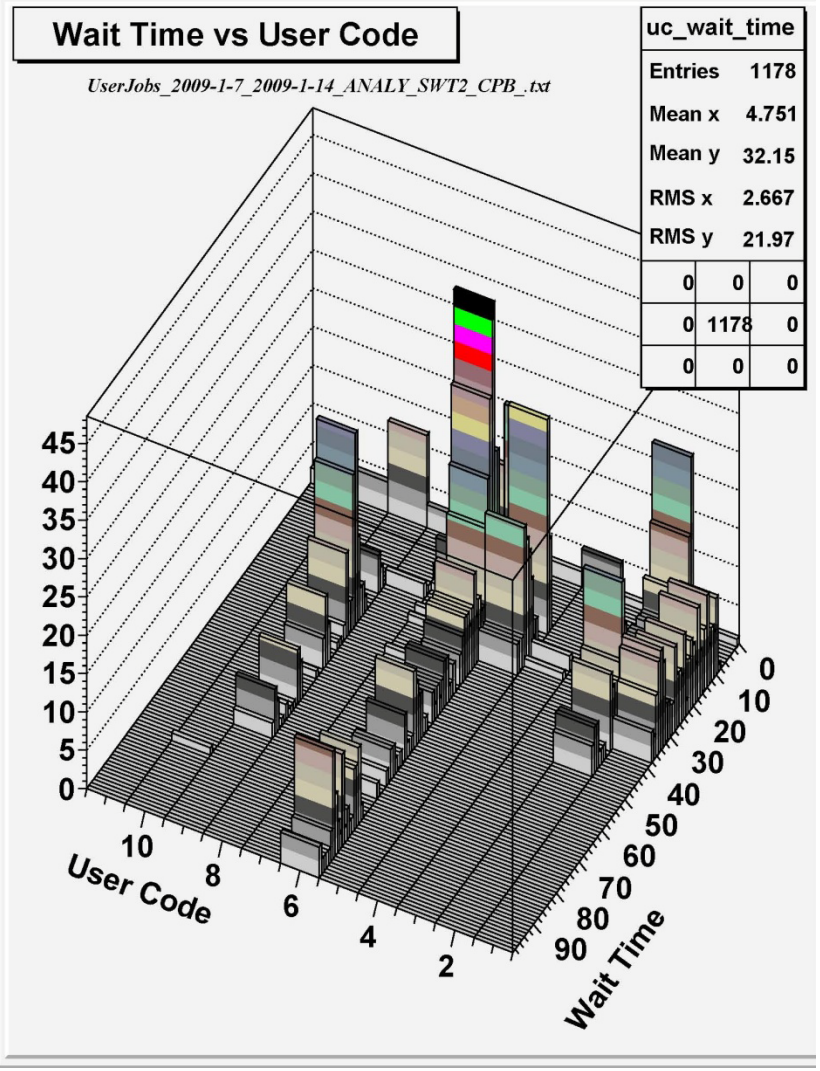
Usage pattern study



ANALY_SWT2_CPB queue, one week, ~1k jobs

- Wait time was on average ~30 minutes

Usage pattern study



ANALY_SWT2_CPB queue, one week, ~1k jobs

- Power user #6 has longest tail in the wait time distribution

Summary

- Panda monitoring collects large amount of information about analysis jobs
- We attempted to look at everyday analysis activity using Panda database
- Caveats –
  - We looked only at a few Atlas analysis sites – far from a complete study!
  - User analysis pattern may change in the future
  - Panda may be retuned, queues will be bigger, etc
- Nevertheless a few qualitative observations can be made:
  - Analysis efficiency determined as a ratio of payload run time to wall time was ~10%
  - It is determined by a sub-job wait time
  - Pilot efficiency defined as a ratio of payload run time to pilot run time is ~70%, with a peak at 95%.
  - Analysis activity is dominated by a few "power users" who submit a majority of jobs
  - Due to the Panda "fair-share" mechanism these users suffered the longest wait time.
  - Submission pattern is dominated by large round numbers of sub-jobs – like 500 or 1000.
  - Why?
- What is an optimal size of the job for a site with a given queue depth?
  - Is it determined by the dataset size? Analysis speed? Size of the sub-job sandbox? Queue time limit? A combination of the above?
  - Is there an optimal job size for any fixed conditions?
- The type of information described in this talk will be incorporated into Panda monitoring