

# Batch Queue Predictions

Dan N

# Outline

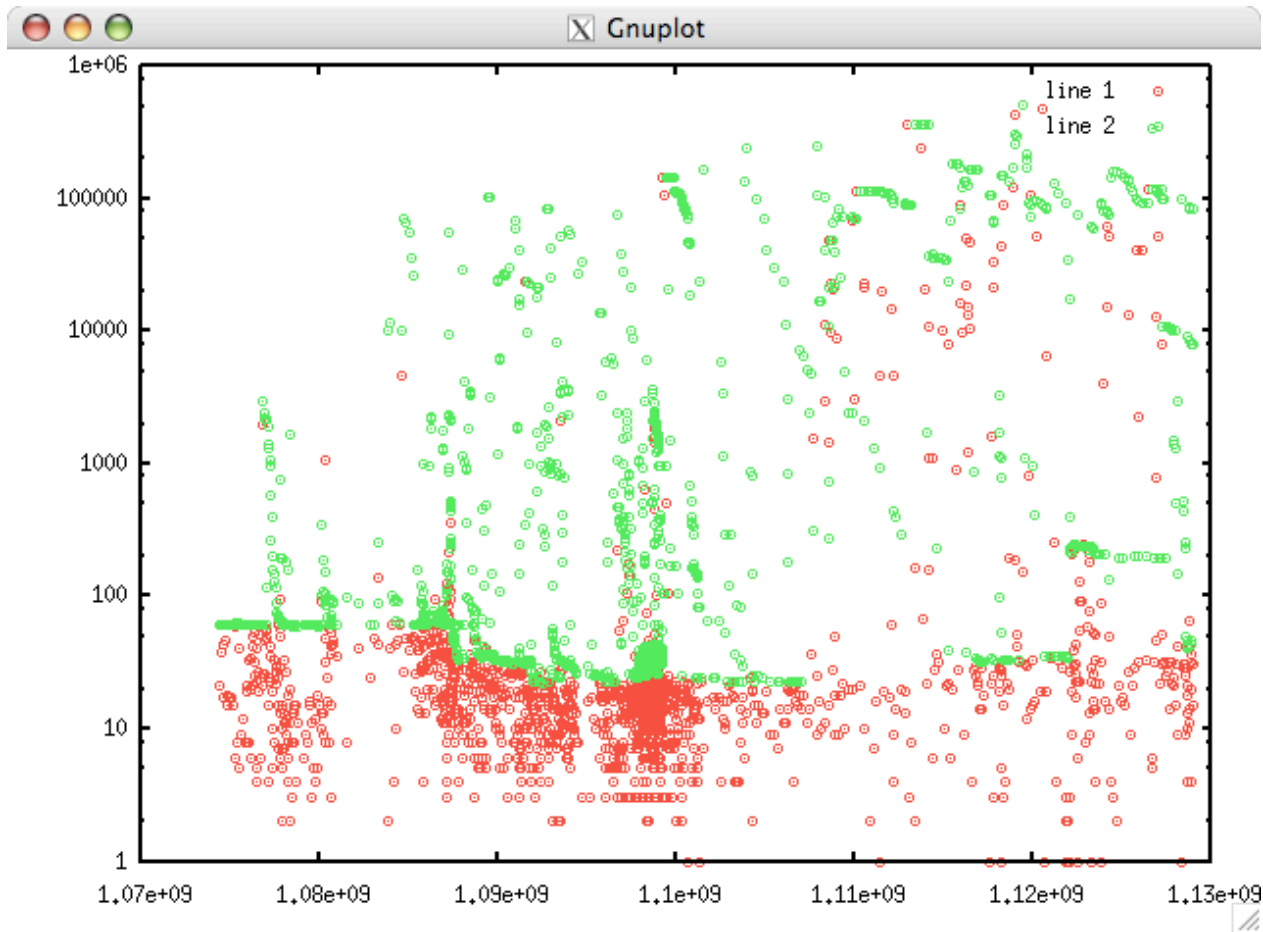
- What we have today
- What we are currently working on
- What we need for LEAD-VGrADS DEMO integration

# What we have today

- Start with *historical traces* of job wait time data
  - 1145444710 27365 4 43200
  - 1145444710 27397 4 43200
  - 1145469362 1 8 600
  - 1145473477 12 17 3600
- End with predictions of *quantiles*, with specified confidence
- Predictions made using trace based simulation, one instance of simulation per machine/queue tuple

# Simulation Trace Output

## .95 quantile, .95 confidence



# Today

- Can do any quantile, but confidence currently discrete (90%, 95%, 99%)
- Rigorous verification of .95, .75 and .5 quantiles
- Correct and accurate for most data we've encountered (*54/55 queues*)
- We confidently predict bounds, we cannot assume *actual wait times!*

# Invert It!

- Predictions made by inferring quantile from sorted empirical data wait times
- Answers the ‘bound’ question, “What is the maximum number of seconds (Y) my job will wait, P% of the time?”
  - Give us P, we give you Y
- Can invert the data and method
  - Give us Y, we give you P

# Current Work

- Using median quantile predictions to rank resources for makespan scheduling (Rice/UCSB paper hopefully to appear at sc06!)
- Use inverted percentage tool, plus batch queue tricks, to ‘guarantee’ resource availability at a certain time

# Inverted Batch Queue Trick

- User says, “I need  $N$  nodes,  $T$  seconds from now, for  $Y$  seconds, with  $P\%$  chance please.”
- We ask ‘ $P_{est}$  = submit a request for  $N$  nodes for  $T+Y$  seconds, right now’
- If  $P_{est} > P\%$ , wait and ask again later
- When  $P_{est}$  is first  $\leq P\%$ , submit job



# Cases

- IF our job executes early, we run the loop again. If we have a new  $\geq P\%$  certain results, we cancel the job and resubmit.
- IF once our job arrives and we can no longer find a  $P\%$  inverted result, we hold the reservation
- IF our jobs arrive late, we must live with the consequences
- **CONSERVATIVE:** when job arrives, hold it no matter what

# Other Inverted Trick

- Assume we will get the resources immediately
- Request a job of duration which lasts from now until the end of required slot
- If we get resource, ask if probability of getting new allocation high enough, if so delete allocation and try again

# LEAD-VGrADS DEMO

## integration

- Vgrads scheduler will be creating vgdI requests, with slot information
- Vges can use inverted trick to select which machine to submit jobs to immediately, vg nodes enter new 'pending state'
- Vges detects/in told when resources become available, vg nodes go to 'Bound' state
- Requires Vges to be able to call external program periodically during the 'pending' state

# Questions

- How do we effect our own predictions?
  - Predictions are only updated when jobs make it through the queue
- Backup plan: use ‘shortest’ queue to immediately request nodes on same resource the inverted method is being attempted on.
- Explore possibility of asking ‘job of N nodes for Y seconds, no sooner than T seconds from now with P% probability’
  - We don’t know how MUCH later!
  - Can make both upper and lower bound predictions, try to create a window of opportunity

