Challenges in Evaluation of Parallel Job Schedulers

Dror Feitelson
The Hebrew University of Jerusalem
• Parallel job schedulers
  – Given resource requirements of new jobs
  – Processors
  – Estimated runtime (inaccurate upper bound)
  – Memory?
  – Decide on order of execution and allocation of processors
  – On-line algorithm (don’t know future jobs)
  – Used on clusters, grids, and supercomputers
• Parallel job schedulers
• Evaluation
  – Estimate performance (typically) using simulations
  – Average response time (wait time)
  – Average slowdown (bounded?)
  – Given alternative schemes, which is better
  – Find “optimal” parameter values
  – Depends on workload (the input)
  – Distributions of parameters (job sizes, runtime, ...)
  – Correlations (size-runtime, daily cycle, ...)
Parsing the Title

• Parallel job schedulers
• Evaluation
• Challenges
  – It isn’t easy to do right
  – How to create different load conditions
  – How to incorporate feedback
  – What level of detail to employ
workloads
Workloads

users → jobs → scheduler → parallel system in production use
Workloads

- Users
- Jobs
- Accounting logs
- Scheduler
- Parallel system in production use

Parallel Workloads Archive
www.cs.huji.ac.il/labs/parallel/workload
Using Workloads

log → jobs → simulated scheduler → performance results
Using Workloads

- Log
- Jobs
- Realism
  - Inflexible
  - Performance results
- Statistical model
  - Flexibility
  - Synthetic jobs
  - Lose details
Creating different loads
Work with Netanel Zakay
Why?

• Characterize performance as function of load 
  (Like queueing analysis)
• Find system capacity
  (maximal sustainable load)
• Serve to decouple system and users
  – Users generate load
  – System performance depends on load
  – “Don’t need to know details of user behavior”
Common Approaches

• Use logs with different loads
• Change load by changing job sizes
• Change load by changing runtimes
• Change load by changing interarrivals

• May not be available
• Changes fragmentation, limited resolution
• Causes correlation of load and response time
• Break dependencies, daily cycle
Workload Resampling

• Break log into users
  – Multiple sub-logs with jobs of one user
  – Maintain sessions, locality
  – Create pool of users

• Resample to create new log
  – Select users from the pool (with repetitions)
  – Mix and match in random way
  – Maintain synchronism with daily/weekly cycle
Resampling Details

• Long term users – active for more than 12 weeks
  – Initially all there but start at random week
  – Restart as needed as simulation continues

• Short term users – up to 12 weeks
  – Initially number in average week
  – Find average arrival rate of new users
  – In simulations add new users each week

• Edge users – only within 4 weeks of start/end
  – Don’t use them
Resampling Benefits

- Can change the number of users
  - More users => higher load
  - Less users => lower load
- Create longer log => converging simulation
- Create multiple instances => confidence intervals
- Combine data from many logs => improve representativeness
Resampling Benefits

• Can change the number of users
  – More users => higher load
  – Less users => lower load
• Create longer log => converging simulation
• Create multiple instances => confidence intervals
• Combine data from many logs => improve representativeness
The Feedback Problem

• Users react to load
  – Good performance => submit more jobs
  – Lousy performance => go home

• More users does not necessarily translate to higher load
  – Higher congestion => bad performance => some users reduce their activity

• So resampling with more/less users isn’t really a good solution for changing load
Incorporating feedback
Work with Edi Shmueli
Users are Humans

• They react to system state
  – Good performance => submit more jobs
  – Lousy performance => go home

• They game the system
  – Understand the scheduler
  – Provide false data to cheat it

• They are myopic
  – Personal interest rather than global wellness
Aside: Runtime Estimates

- If runtime estimate is low, job has a better chance to backfill
- If it is too low, job will be killed
- So users are motivated to provide accurate estimates
Performance Feedback

- User behavior leads to **negative feedback**
  - If load is high they reduce submitting of jobs
  - If load is low they submit more jobs
- Captures interaction between users
- **Scheduler performance can affect workload**
- There is no such thing as “the real workload”
- Workload logs reflect the scheduler on the logged system, and its interaction with its users
Implications for Performance Evaluation

• Comparing schedulers “under same conditions” means with same users (not with same log!)

• Performance metrics change
  – Better scheduler => more jobs => higher throughput
  – Better scheduler => more jobs => maybe higher response time (considered worse!)

• Using a user feedback model counteracts efforts to change load
Resampling with Feedback

log
modeling
user behavior
Resampling with Feedback

- Same sequences of jobs
- Timing adjusted based on performance

resampling of users

log

modeling user behavior

feedback

performance results

simulated scheduler

jobs
Too Much Stability

• Consistent use of user feedback model implies stability (feedback is negative)
• But real systems experience large load fluctuations
• Real systems (and users) have more variability and complexity
Complexity and realism
Work with David Krakov
EASY Simulations

• Scheduler has simple algorithm (e.g. EASY backfilling)

• Jobs have simple requirements
  – Number of processors
  – Maybe also requested runtime

• Arrivals from log (possibly modified by feedback)

• Possible to achieve high utilization under high load
Easy to Understand Results

Like in queueing analysis

Offered load

Average response time
Real Scheduling

• Algorithm may be complex (e.g. MAUI with dozens of parameters)

• Jobs have multiple additional requirements
  – Memory
  – Software licenses
  – Hardware and software configurations
  – Fairness at user or group level

• Constraints limit achievable utilization
Heatmaps

- Show detailed performance characteristics
- Analysis at job level
- X is utilization experienced by job
- Y is performance experienced by job
- Shading is number of jobs with given X and Y
No correlation of response time and load.
Metrics
Comparison with Simulation

- CTC-SP2-1996
- CTC-SP2-1996-EASY

Graphs showing response time vs. utilization.
Comparison with Simulation
Comparison with Simulation
Results

• Simulations do not reflect reality
  – Real systems seem to be more constrained

• Averages do not represent variability
  – Variability in load
  – Variability in performance

• No correlation of load and performance at job level
conclusions
Life Is Tough

- Not sure that performance vs. load is meaningful
- Feedback is an important effect
- EASY simulations are over-simplified
- There’s a lot we don’t know or understand
- There’s no single true answer
  - Need to deal with variability
Academia vs. “Real People”

• Academia doesn’t know about all the constraints faced by real schedulers
• Academia doesn’t know about the considerations and goals of real schedulers
• Academia doesn’t contribute real ideas or solutions
What You Can Do

• Be aware of constraints on scheduling
  – Need to be known for relevant evaluations
  – Maybe they can be removed?

• Try to understand users
  – What they want from the system
  – How this can be expressed as a metric
  – How it affects their behavior

• Collect workload data and contribute to the Parallel Workloads Archive
  – What should be added to the standard workload format?

• Write papers for JSSPP workshop
¡Gracias!
¿preguntas?