Operational Analysis of Processor Speed Scaling

Kai Shen (University of Rochester)
Alex Zhang (HP Labs)
Terence Kelly (HP Labs)
Christopher Stewart (Univ. of Rochester)

Parallelism and Performance

"Parallel architectures... introduce many new optimization parameters, and so far, no successful autotuners for parallel [applications] exist."

The Landscape of Parallel Computing Research:

A Berkeley View

Motivation

- Server applications will increasingly run on multi-processor machines
  - Not multi-core but many-core (i.e., hundreds of cores)
  - Datacenter-on-chip
- Performance questions
  - Will my application's performance improve with more cores? Frequency scaling? Cache structure?

Performance Models

Architecture parameters
e.g., # cores, per-core speed

$$f(X) = P$$

Performance metric e.g.,
response and execution time, throughput

speed \ # cores			
	9sec	6sec	3sec
	12sec	6sec	4sec
	20sec	15sec	9sec

- Models should be
  1. Applicable to many applications and architectures
  2. Easy to calibrate
  3. Instructional; enhance understanding of parallel systems

Operational Laws

- Performance model parameters reflect only measurable quantities
  - Verified by measurements
  - Robust, easy to understand, accessible for administrators

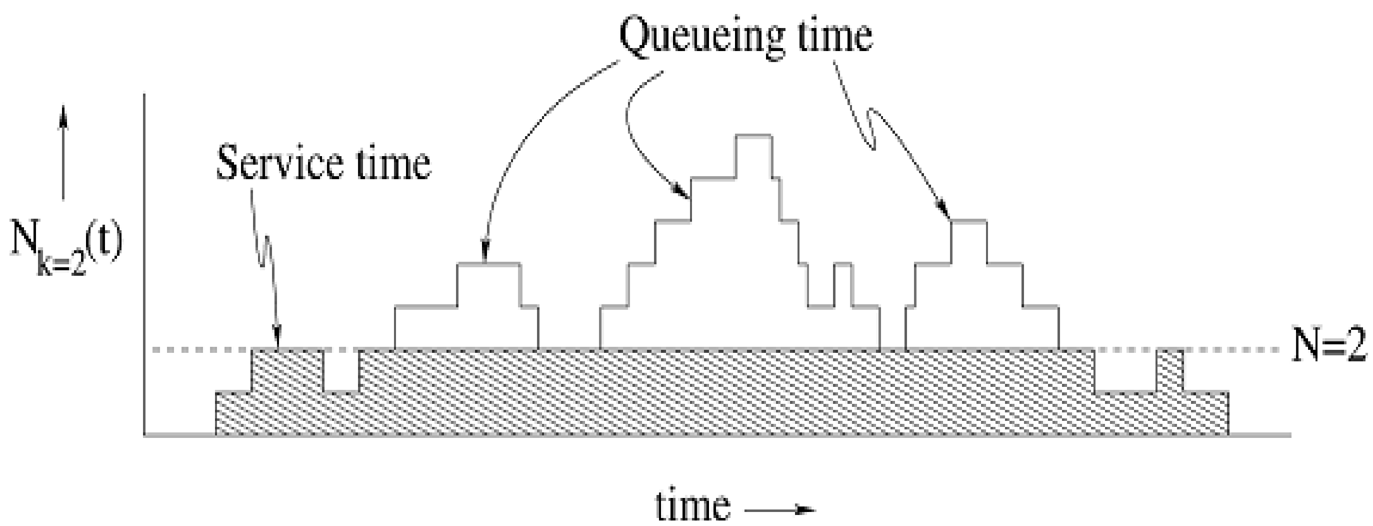
- Well-known operational laws
  - Utilization Law – bottleneck/throughput analysis
  - Little's Law – queuing theory

Revisit Operational Analysis

- Measurable quantities
  - Request arrivals and departures
  - Concept of busy time, response time, and throughput
  - Properties of the parallel architecture
- New approaches for operational analysis
  - Novel analysis of the *occupancy curve*

Occupancy Curve

- Plot of the number of requests in the system versus time
- Horizontal line equal to the number of processors separates requests being serviced from those queued
  - Operational: request arrivals, departures, and number of processors can be measured directly



Processor Speed Scaling

- Processor power can be adjusted by changing parameters or migrating applications
  - May correspond to raw frequencies because of caching

- How does speed scaling affect performance?
  - Deterministic scheduling: Scheduling decisions are based on the set of runnable jobs, their static properties, the amount of remaining service for each job
  - Arrival rate monotonicity: If arrival times of all jobs are multiplied by a constant, aggregate queuing does not increase

Speed Scaling Law

If all processors get a speedup factor of `f`, then aggregate queuing will decrease by at least a factor of `(f -1)/f`.

- Derivation step 1: Increase processor speed by `f`, decrease request arrival times by `1/f`
  - Aggregate queuing delay scales down by `1/f`
  - Rescale x-axis of occupancy curve
- Derivation step 2: Increase request arrival times by `f`
  - No queuing increase
- Operational law that is useful for server management

Takeaway

- Revisit operational laws for future parallel processor architectures
    1. Identified an operational law for server systems that bounds the effect of processor speed scaling

- Ongoing work
    - [MASCOTS 08] Operational laws for capacity planning
    - [USENIX 08] Cross-platform performance model for processor selection and online server management