Approximate Mean Value Analysis for Multi-core Systems

Lei Zhang and Douglas G. Down

Computing and Software
McMaster University
Hamilton, ON, Canada

Motivation

Challenges for performance modeling are brought by multi-core systems:

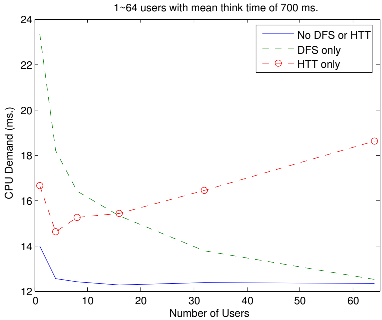
- Non-product-form
- Hyper-Threading Technology (HTT) and Dynamic-Frequency Scaling (DFS)
- Numerical instability of Mean Value Analysis (MVA) [1]

Contributions

We propose a novel performance evaluation method for multi-core systems with the following techniques:

- Flow-equivalent aggregation
- Service demand estimation for the effects of HTT and DFS
- Numerically stable MVA algorithms

HTT and DFS



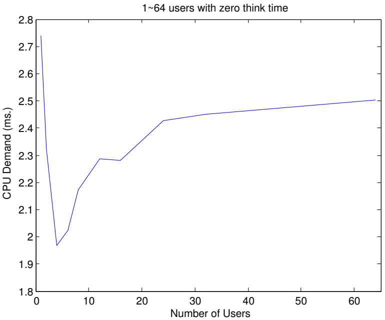


Figure: HTT and DFS impact on CPU demand

Problem and Reason

MVA for load-dependent queues can potentially raise numerical issue:

- This problem is caused by the calculation of the probability of a resource being idle in every iteration of MVA:

$$P_m(0|n) = 1 - \sum_{i=1}^{n} P_m(i|n), \tag{1}$$

  where $P_m(i|n)$ is the probability that i jobs present at mth resource when n jobs in the system
- It may exhibit unreasonable results, such as negative throughputs and response times under heavy load

What's APEM?

- Approximate Performance Evaluation method for Multi-core computer systems
- Basic Ideas:
  1. Use a queueing network with a Flow-Equivalent Server (FES) to model the multi-core system
  2. Use a service rate curve approach to estimate input parameters for the FES
  3. Adjust the service demand due to the effect of DFS
  4. Adopt a numerically stable MVA algorithm to solve the queueing network

Flow-Equivalent Aggregation

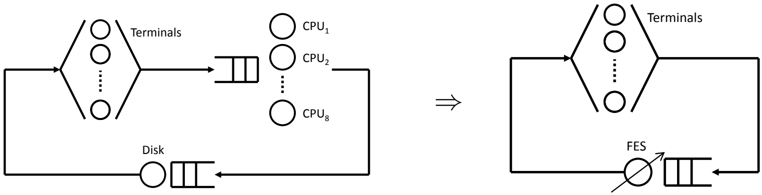


Figure: Approximating network

Observation

There exists a probability that an upcoming job will be (partially) processed at a slow rate, because the core frequency has been scaled down to the lowest value

- The probability that we described above is given by $P(\mathcal{A} > s)$, where s is the sampling interval, and $\mathcal{A}$ is an inter-arrival time

Estimation 1

We assume arrivals follow an exponential distribution with mean Z, and we can calculate $P_{slow}(n)$, which is the probability that a job arrives to a slow server given n jobs in the system, as:

$$P_{slow}(n) = P(\mathcal{A} > s|n) \approx e^{-\frac{sn}{ZK}}, \quad (2)$$

where K is the number of cores in the CPU (note that (2) is an equality only if all the users are "thinking")

Estimation 2

Now, we have the adjustment of the CPU service demand as follows:

$$D'(i) = D(i) \cdot (1 - P_{slow}(n)) + D_{slow}(i) \cdot P_{slow}(n), \qquad (3)$$

where $D(i)$ is the CPU demand with i jobs at the CPU, and $D_{slow}(i)$ is the service demand of the CPU with one core running at the minimum frequency

Estimation 3

To calculate $D_{slow}(i)$, we compute its inverse - $\mu_{slow}(i)$ - using the following equation:

$$\mu_{slow}(i) = \mu_{slow}(1) + \mu(i-1), \tag{4}$$

where $\mu_{slow}(1)$ is measured as one job in the system and the CPU is running at the minimum frequency

Assumptions

- The inter-arrival times follow an exponential distribution
- The average think time is sufficiently large
- If an arrival enters a "slow" server, then the entire job will be processed at the slowest possible rate
- Each logical core works independently and has the same arrival rate

CMVA

Conditional MVA (proposed by Casale [2]):

- The queue length formula avoids the computations of the state probabilities at each node
- It relates the average queue length of a load-dependent server to the conditional queue length as seen by a job during its residence time at that server:

$$R_m(n, t) = D_m(n, t) \times (1 + Q_m(n_m \geq 1, n, t)) \tag{5}$$

- It is proved that:

$$Q_m(n_m \geq 1, n, t) = Q_m(n - 1, t + 1), \tag{6}$$

  with the service rates at this queue shifted up by one more job at the queue

Experimental Settings

- Dell desktop computer equipped with an Intel i7-2600 quad-core processor (octo-core logically with HTT), CPU frequency is dynamically scaled between 1.6 GHz and 3.4 GHz.
- JBoss + MySQL + TPC-W [3]
- CMVA-DFS vs. CMVA vs. Multi-server MVA

Service Rate Curves

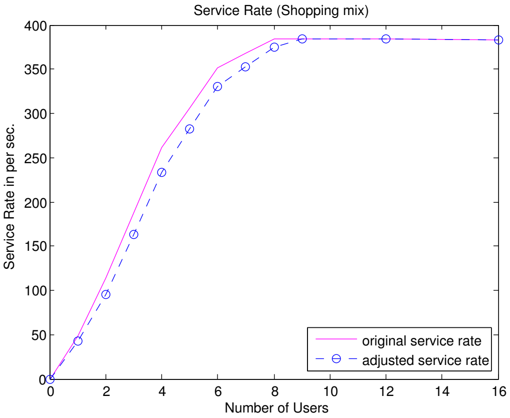


Figure: Service rate of FES

Mean Response Times

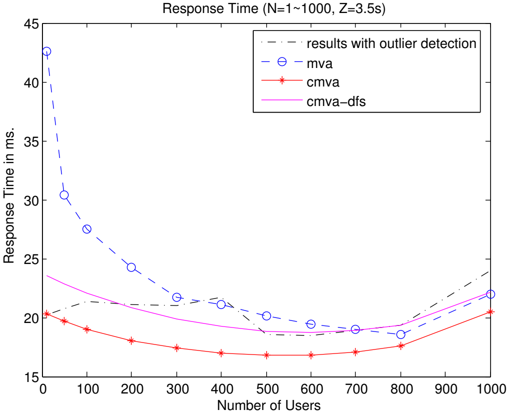


Figure: Predicted and measured R

Conditions for APEM

- Modeled systems must be measurable
- The accuracy of APEM may be affected by the estimation of service rate curves
- Assumptions for the service demand adjustment mentioned before should hold in APEM

Conclusion and Future Work

Performance modeling for multi-core systems

- Approximate non-product-form networks by flow-equivalent aggregation techniques
- Address the effects of DFS with a numerically stable MVA algorithm CMVA-DFS

Potential directions for future work:

- Adapt APEM for open and mixed queueing networks
- Find a numerically stable MVA algorithm with reasonable time and space complexities

References I

[1] M. Reiser and S. S. Lavenberg.
Mean-value analysis of closed multichain queuing networks.
Journal of the ACM, vol. 27, pages 313-322, 1980.

[2] G. Casale.
A note on stable flow-equivalent aggregation in closed networks.
Queueing Systems, vol. 60, pages 193-202, 2008.

[3] T. Horvath.
TPC-W J2EE Client Implementation.
http://www.cs.virginia.edu/ th8k/downloads/, last accessed May 2013.