An Empirical Characterization of Several Server Workloads

Kai Shen
Dept. of Computer Science, University of Rochester

Workload Characterization and Benchmarking

- Workload characterization
  - understand realistic workload characteristics/patterns
  - generalize these characteristics/patterns
- Benchmarking
  - compare across systems
  - evaluate the effectiveness of system-level techniques

Empirical Server Workloads

- TPC-C
  - database-driven standard benchmark
  - five types of transactions
- SPECweb
  - web server standard benchmark
  - GET/POST reqs, with or without cookie, varying file sizes
- RUBiS
  - J2EE multi-component service
  - several types of requests in a typical auction service
- WeBWorK
  - online homework distribution and solution checking
  - driven by real user traces with thousands of different teacher-created problem sets

Distribution of Request CPU Usage

- X-axis
  - 1 to 99-percentile range
- Y-axis
  - normalize to even distribution
- Discussions
  - clustering
  - regularity
### Inter-property Correlation

- **X/Y-axes**: 1 to 99th percentile range
- **Discussion**: correlation, implication

### Re-evaluate Past Results:
**Event Chain-based Request Classification**

- Classify requests into groups based on event chains [Barham et al. 2004]
  - there are multiple types of events
  - event chain distance measured as string-edit distance
- A simple re-evaluation
  - just system call events (each system call number is a type)
  - ignore the event order to save computation time
- Use a representative to represent all group members
- **Coefficient of determination**, $R^2$ accuracy $\left(1 - \frac{E_{\text{model}}}{E_{\text{mean}}}\right)$:
  - TPC-C: 0.98
  - SPECweb: 0.94
  - RUBiS: 0.89
  - WebWorK: -0.24

### Re-evaluate Past Results:
**Type-based Resource Usage Prediction**

- Weighted linear model of request CPU utilization on request types [Stewart et al. 2007]
  - $U = c_0 + c_1 T_1 + c_2 T_2 + \ldots + c_n T_n$
  - $T_1, T_2, \ldots, T_n$ are the proportions of each request type in the workload
- RUBiS: several auction service types
- WebWorK: submitting/viewing problems, submitting solutions

### WeBWorK: An Collaborative Web Application

- **WeBWorK**
  - weaker clustering, less behavior regularity, weaker inter-property correlation, compared to standard benchmarks like TPC-C, SPECweb, and RUBiS
  - due to independent content creation from large number of end users in collaborative web applications
  - impacts on workload modeling and performance prediction
- Emerging collaborative web applications
  - Wiki-based web sites
  - social networking sites
  - others?