Program Behavior Prediction Using a Statistical Metric Model

Ruhi Sarikaya, Canturk Isci and Alper Buyuktosunoglu

**Workload Prediction and Adaptive Management**

- Dynamically-varying Workloads
- Runtime Monitoring
- Workload Characterization, Classification and Prediction
- Dynamic Adaptations for Resource/Power Management

**Predicting Workload Behavior**

- Define workload features
- Determine future characteristics

**Existing Techniques:**
- Last Value (Strawman/Reactive)
- History Based (Statistical)
- Table Based (Patterns)

**Our Approach: Statistical Metric Modeling**
- Inspired from natural language modeling
- Workload features ⊗ words in language
- Workload patterns ⊗ grammar structure
- Model workload structure at runtime
- Build metric probability distributions
- Predict future characteristics

**Statistical Metric Model (SMM)**

**SMM Overview:**

- Probability distribution $P(s)$ over sequences $s$:
  
  \[ s = (s_1, s_2, ..., s_l) \]

- Ex: $P(\text{"How are you doing"}) = 0.001$

- Difficult to compute $P(s) = P(s_1, s_2, ..., s_l)$

- Decompose the probability instead:
  
  \[ P(s) = P(s_1) \times P(s_2 | s_1) \times P(s_3 | s_2, s_1) \times ... \times P(s_l | s_{l-1}, ..., s_1) \]

- Ex: $P(\text{"How are you doing"}) = P(\text{"how"}) \times P(\text{"are\"} | \text{"how"}) \times P(\text{"you\"} | \text{"are\"}) \times P(\text{"doing\"} | \text{"you\"})$

- Use $n$-gram Approximation:
  
  Assume each word depends only on the previous $n$ words

  \[ P(s) = \prod_{i=1}^{l} P(s_i | s_{i-1}, ..., s_{i-n+1}) \]

- Apply model smoothing to conditional distributions to compensate for data sparsity

**SMM for Workload Behavior Prediction:**

- **Global metric modeling**:
  
  \[ P_{global}(s_{l-1}) \]

- **Temporal metric modeling**:
  
  \[ P_{temporal}(s_l | s_{l-1}, ..., s_{l-n+1}) \]

- Overall model:
  
  \[ P_{final} = \beta_1 \cdot P_{global} + \beta_2 \cdot P_{temporal} \]

**Experimental Results**

- More Important for SMM:
  
  - More variability
  - Harder to predict

- Prediction Accuracy:
  
  Across all workloads: 20% improvement
  
  Variable workloads: 40% improvement
  
  Repetitive runs (x2)
  
  Additional 15% improvement across all workloads

**Summary**

**Primary Contributions**

- New workload behavior prediction approach Inspired by language modeling
- Evaluation with a comprehensive set of benchmarks and datasets
- Significant improvement in accuracy over prior approaches

**4 Main SMM Strengths**

- Models long-term global patterns in application behavior
- Can track and predict variable-length patterns
- Resilient to small fluctuations in workload behavior
- Adapts and improves over time, as it learns more it predicts better

**Evaluation Highlights**

- Improve prediction accuracy by 40% for variable workloads
- Average improvement of 20% across all benchmarks
- Additional 15% improvement with recurring workloads

© 2010 IBM Corporation