CS 147: Computer Systems Performance Analysis
Workload Characterization
Overview

Terminology

Specifying Parameters

Identifying Parameters
  Histograms
  Principal-Component Analysis
  Markov Models

Clustering
  Clustering Steps
  Clustering Methods
  Using Clustering
User (maybe nonhuman) requests service
- Also called workload component or workload unit
- Workload parameters or workload features model or characterize the workload
Selecting Workload Components

- Most important: components should be *external*: at interface of SUT
- Components should be homogeneous
- Should characterize activities of interest to the study
Choosing Workload Parameters

- Select parameters that depend only on workload (not on SUT)
- Prefer controllable parameters
- Omit parameters that have no effect on system, even if important in real world
Specifying Parameters

Averaging

- Basic character of a parameter is its average value
- Not just arithmetic mean
- Good for uniform distributions or gross studies
Most parameters are non-uniform

Specifying variance or standard deviation brings major improvement over average

Average and s.d. (or C.O.V.) together allow workloads to be grouped into classes
  - Still ignores exact distribution
Single-Parameter Histograms

- Make histogram or kernel density estimate
- Fit probability distribution to shape of histogram
- Chapter 27 (not covered in course) lists many useful shapes
- Ignores multiple-parameter correlations
Multi-Parameter Histograms

- Use 3-D plotting package to show 2 parameters
  - Or plot each datum as 2-D point and look for “black spots”
- Shows correlations
  - Allows identification of important parameters
- Not practical for 3 or more parameters
How to analyze more than 2 parameters?
- Could plot endless pairs
  - Still might not show complex relationships
- Principal-component analysis solves problem mathematically
  - Rotates parameter set to align with axes
  - Sorts axes by importance
Advantages of PCA

- Handles more than two parameters
- Insensitive to scale of original data
- Detects dispersion
- Combines correlated parameters into single variable
- Identifies variables by importance
Disadvantages of PCA

- Tedious computation (if no software)
- Still requires hand analysis of final plotted results
- Often difficult to relate results back to original parameters
Markov Models

- Sometimes, distribution isn’t enough
- Requests come in sequences
- Sequencing affects performance
- Example: disk bottleneck
  - Suppose jobs need 1 disk access per CPU slice
  - CPU slice is much faster than disk
  - Strict alternation uses CPU better
  - Long disk-access strings slow system
Introduction to Markov Models

- Represent model as state diagram
- Probabilistic transitions between states
- Requests generated on transitions
Creating a Markov Model

- Observe long string of activity
- Use matrix to count pairs of states
- Normalize rows to sum to 1.0

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>Network</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>0.6</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td>0.3</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>Disk</td>
<td>0.8</td>
<td></td>
<td>0.2</td>
</tr>
</tbody>
</table>
Example Markov Model

- Reference string of opens, reads, closes: ORORRCOORCRCRRCC
- Pairwise frequency matrix:

<table>
<thead>
<tr>
<th></th>
<th>Open</th>
<th>Read</th>
<th>Close</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>1</td>
<td>3</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Read</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>Close</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

2015-06-15
Identifying Parameters

Markov Models

Markov Model for I/O String

- Divide each row by its sum to get transition matrix:

<table>
<thead>
<tr>
<th></th>
<th>Open</th>
<th>Read</th>
<th>Close</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>0.25</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>Read</td>
<td>0.13</td>
<td>0.50</td>
<td>0.37</td>
</tr>
<tr>
<td>Close</td>
<td>0.33</td>
<td>0.33</td>
<td>0.34</td>
</tr>
</tbody>
</table>

- Model:
Clustering

Often useful to break workload into categories
“Canonical example” of each category can be used to represent all samples
If many samples, generating categories is difficult
Solution: clustering algorithms
Steps in Clustering

- Select sample
- Choose and transform parameters
- Drop outliers
- Scale observations
- Choose distance measure
- Do clustering
- Use results to adjust parameters, repeat
- Choose representative components
Clustering algorithms are often slow
  - Must use subset of all observations

Can test sample after clustering: does every observation fit into some cluster?

Sampling options
  - Random
  - Heaviest users of component under study
Choosing and Transforming Parameters

- Goal is to limit complexity of problem
- Concentrate on parameters with high impact, high variance
  - Use principal-component analysis
  - Drop a parameter, re-cluster, see if different
- Consider transformations such as Sec. 15.4 (logarithms, etc.)
Dropping Outliers

- Must get rid of observations that would skew results
  - Need great judgment here
  - No firm guidelines
- Drop things that you know are “unusual”
- Keep things that consume major resources
  - E.g., daily backups
Cluster analysis is often sensitive to parameter ranges, so scaling affects results.

Options:
- Scale to zero mean and unit variance
- Weight based on importance or variance
- Normalize range to [0, 1]
- Normalize 95% of data to [0, 1]
Choosing a Distance Measure

- Endless possibilities available
- Represent observations as vectors in \( k \)-space
- Popular measures include:
  - Euclidean distance, weighted or unweighted
  - Chi-square distance
  - Rectangular (“Manhattan”) distance

Chi-square distance is:

\[
d = \sum_{k=1}^{n} \left\{ \frac{(x_{ik} - x_{jk})^2}{x_{jk}} \right\}
\]

and requires \( x_k \) to be close together or low values of \( x_k \) will over-weight parameters. Used primarily in distribution fitting.
Clustering Methods

- Many algorithms available
- Computationally expensive (NP to find optimum)
- Can be simple or hierarchical
- Many require you to specify number of desired clusters
- Minimum Spanning Tree (from book) is not only option!
Types of Clustering

- Agglomerative vs. divisive
- Hierarchical vs. non-hierarchical
Minimum Spanning Tree Clustering

- Start with each point in a cluster
- Repeat until single cluster:
  - Compute centroid of each cluster
  - Compute intercluster (inter-centroid) distances
  - Find smallest distance
  - Merge clusters with smallest distance
- Result is a hierarchy of clusters
- Method produces stable results
  - But not necessarily optimum
K-Means Clustering

- One of most popular methods
- Number of clusters is input parameter, $k$
- First randomly assign points to clusters
- Repeat until no change:
  - Calculate center of each cluster: $(\bar{x}, \bar{y})$
  - Assign each point to cluster with nearest center
- Big problem: How to choose $k$
  - Prior knowledge
  - Trial and error
Jarvis & Patrick’s Method

- Start with each point in own cluster
- For each point, make list of $n$ closest other points
- For each point pair, if $k$ of $n$ nearest neighbors are shared, combine their clusters
- Finds non-globular clusters
- Extremely sensitive, in non-intuitive ways, to $k$ and $n$
Interpreting Clusters

- Art, not science
- Drop small clusters (if little impact on performance)
- Try to find meaningful characterizations
- Choose representative components
  - Number proportional to cluster size or to total resource demands
Drawbacks of Clustering

- Clustering is basically AI problem
- Humans will often see patterns where computer sees none
- Result is extremely sensitive to:
  - Choice of algorithm
  - Parameters of algorithm
  - Minor variations in points clustered
- Results may not have functional meaning