Performance Evaluation:

Workload Characterization Techniques

Hongwei Zhang
http://www.cs.wayne.edu/~hzhang

Acknowledgement: this lecture is partially based on the slides of Dr. Raj Jain.
Outline

- Terminology
- Selection of Workload Components and Parameters
- Workload Characterization Techniques
Outline

- Terminology
- Selection of Workload Components and Parameters
- Workload Characterization Techniques
Terminology

- **Workload**
  - service requests to the system

- **User**
  - entity that makes the service request

**Workload component**

- User is usually called “workload component” in workload characterization
- E.g., applications, sites, and user Sessions (each one of them can be regarded as a component)

- **Workload parameter**
  - measured quantities that are used to model/characterize the workload
    - For example: packet sizes, source-destinations of a packet
Outline

- Terminology

- Selection of Workload Components and Parameters

- Workload Characterization Techniques
Selection of workload components

- The workload component should be at the SUT (i.e., system under test) interface

- Each component should represent as homogeneous a group as possible
  - E.g., combining very different users into a site workload may not be meaningful

- Purpose of study and domain of control also affect the choice
  - E.g., a mail system designer is more interested in a typical mail session than a typical user session involving many applications
Selection of workload parameters

- Do not use parameters that depend upon the system
  - E.g., the elapsed time, CPU time

Instead, only use parameters that depend on workload itself

- E.g., characteristics of service requests:
  - Arrival Time
  - Type of request or the resource demanded
  - Duration of the request
  - Quantity of the resource demanded, for example, buffer space

- Exclude those parameters that have little impact
Outline

- Terminology
- Selection of Workload Components and Parameters
- Workload Characterization Techniques
Workload characterization techniques?

- Averaging
- Specifying dispersion
- Histograms: single-parameter, multi-parameter
- Principal Component Analysis
- Markov Models
- Clustering
Workload characterization techniques

- Averaging
- Specifying dispersion
- Histograms: single-parameter, multi-parameter
- Principal Component Analysis
- Markov Models
- Clustering
Averaging

- (arithmetic) Mean
  \[ \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \]

- Alternatives (to be discussed in detail later)
  - Mode (for categorical variables): Most frequent value
  - Median: 50-percentile

- Geometric mean
- Harmonic mean
Workload characterization techniques

- Averaging
- Specifying dispersion
- Histograms: single-parameter, multi-parameter
- Principal Component Analysis
- Markov Models
- Clustering
Specifying dispersion

- Standard deviation (s): square root of variance $s^2$
  \[ s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \]

- Coefficient of variation (C.O.V.): \( \frac{s}{\bar{x}} \)

- Alternatives (to be discussed in detail later)
  - Range
  - 10- and 90-percentile
  - Semi-interquartile range
  - Mean absolute deviation
What if C.O.V. is high?

- Then “mean” is not sufficient to characterize the workload

- Alternatives
  - Complete histogram
  - Divide users (i.e., workload components) into classes, and specify average for each class
# Case Study: Program Usage in Educational Environments (6 universities)

<table>
<thead>
<tr>
<th>Data</th>
<th>Average</th>
<th>Coef. of Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time ((VAX-11/780^{TM}))</td>
<td>2.19</td>
<td>40.23</td>
</tr>
<tr>
<td>Elapsed time</td>
<td>73.90</td>
<td>8.59</td>
</tr>
<tr>
<td>Number of direct writes</td>
<td>8.20</td>
<td>53.59</td>
</tr>
<tr>
<td>Direct write bytes</td>
<td>10.21</td>
<td>82.41</td>
</tr>
<tr>
<td>Size of direct writes</td>
<td>1.25</td>
<td></td>
</tr>
<tr>
<td>Number of direct reads</td>
<td>22.64</td>
<td>25.65</td>
</tr>
<tr>
<td>Direct read bytes</td>
<td>49.70</td>
<td>21.01</td>
</tr>
</tbody>
</table>

- High Coefficient of Variation
Case study (contd.): only focus on editing sessions

<table>
<thead>
<tr>
<th>Data</th>
<th>Average</th>
<th>Coef. of Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time (VAX-11/780)</td>
<td>2.57 seconds</td>
<td>3.54</td>
</tr>
<tr>
<td>Elapsed time</td>
<td>265.45 seconds</td>
<td>2.34</td>
</tr>
<tr>
<td>Number of direct writes</td>
<td>19.74</td>
<td>4.33</td>
</tr>
<tr>
<td>Direct write bytes</td>
<td>13.46 Kilo-bytes</td>
<td>3.87</td>
</tr>
<tr>
<td>Size of direct writes</td>
<td>0.68 kilo-bytes</td>
<td></td>
</tr>
<tr>
<td>Number of direct reads</td>
<td>37.77</td>
<td>3.73</td>
</tr>
<tr>
<td>Direct read bytes</td>
<td>36.93 Kilo-bytes</td>
<td>3.16</td>
</tr>
</tbody>
</table>

- Much more reasonable C.O.V.
Workload characterization techniques

- Averaging
- Specifying dispersion
- Histograms: single-parameter, multi-parameter
- Principal Component Analysis
- Markov Models
- Clustering
Single-parameter histograms

- A histogram shows the relative frequency of various values of a parameter
  - For continuous-value parameters, need to divide parameter range into subranges called *buckets/cells*

- E.g.
Single-parameter histograms (contd.)

- Could also use tabular/vector representation

- In analytical modeling, histograms can be used to fit a probability distribution, or to verify that the distribution used in the model is similar to what is observed in the histogram

- (-) # of numerical values: \( n \) buckets \(* m \) parameters \(* k \) components
  - May be too much details to be useful
  - => should be used only if variance is high & averages cannot be used

- (-) Key problem: Ignores correlation among parameters
  - => multi-parameter histogram
Multi-parameter histograms

- Difficult to plot joint histograms for more than two parameters
Workload characterization techniques

- Averaging
- Specifying dispersion
- Histograms: single-parameter, multi-parameter
- Principal Component Analysis
- Markov Models
- Clustering
Principal component analysis (PCA)

- **Key Idea**: Use a weighted sum of parameters *to classify workload components*
  - Need to identify what contributes variance

  - For j-th component, Let $x_{ij}$ denote the i-th parameter for j-th component: $y_j = \sum_{i=1}^{n} w_i x_{ij}$

  - PCA assigns weights $w_i$'s such that $y_j$'s provide the maximum discrimination among the components

  - The quantity $y_j$ is called the principal factor (more precisely, "principal component" in statistics)

  - The factors are ordered s.t., the first factor explains the highest percentage of the variance, the second factor explains a lower percentage ...
PCA (contd.)

- Statistically: given $X$,
  - The $y$'s are linear combinations of $x$'s:
    \[ y_i = \sum_{j=1}^{n} a_{ij} x_j \]
    Here, $a_{ij}$ is called the *loading* of variable $x_j$ on factor $y_i$
  - The $y$'s form an orthogonal set (i.e., their pairwise inner product is zero); This is “equivalent” to stating that $y_i$'s are “uncorrelated” to each other (note: not in the precise sense of “uncorrelation”)
    - Two r.v., $X$ and $Y$, $X$ and $Y$ are said to be *orthogonal* if $E[XY] = 0$, and *uncorrelated* if $E[XY] = E[X] \times E[Y]$
  - The $y$'s form an ordered set such that $y_1$ explains the highest percentage of the variance in resource demands
Find principal factors

- Find the correlation matrix of normalized variables

- Find the eigenvalues of the matrix and sort them in the order of decreasing magnitude

- Find corresponding eigenvectors
  - These give the required loadings
Example of PCA:
packets tx & rx for all the workstations of a network

<table>
<thead>
<tr>
<th>Obs. No.</th>
<th>Variables</th>
<th>Normalized Variables</th>
<th>Principal Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x_s$</td>
<td>$x_r$</td>
<td>$x'_s$</td>
</tr>
<tr>
<td>1</td>
<td>7718</td>
<td>7258</td>
<td>1.359</td>
</tr>
<tr>
<td>2</td>
<td>6958</td>
<td>7232</td>
<td>0.922</td>
</tr>
<tr>
<td>3</td>
<td>8551</td>
<td>7062</td>
<td>1.837</td>
</tr>
<tr>
<td>4</td>
<td>6924</td>
<td>6526</td>
<td>0.903</td>
</tr>
<tr>
<td>5</td>
<td>6298</td>
<td>5251</td>
<td>0.543</td>
</tr>
<tr>
<td>6</td>
<td>6120</td>
<td>5158</td>
<td>0.441</td>
</tr>
<tr>
<td>7</td>
<td>6184</td>
<td>5051</td>
<td>0.478</td>
</tr>
<tr>
<td>8</td>
<td>6527</td>
<td>4850</td>
<td>0.675</td>
</tr>
<tr>
<td>9</td>
<td>5081</td>
<td>4825</td>
<td>-0.156</td>
</tr>
<tr>
<td>10</td>
<td>4216</td>
<td>4762</td>
<td>-0.652</td>
</tr>
<tr>
<td>17</td>
<td>3644</td>
<td>3120</td>
<td>-0.981</td>
</tr>
<tr>
<td>18</td>
<td>2020</td>
<td>2946</td>
<td>-1.914</td>
</tr>
<tr>
<td>$\sum x$</td>
<td>96336</td>
<td>88009</td>
<td>0.000</td>
</tr>
<tr>
<td>$\sum x^2$</td>
<td>567119488</td>
<td>462661024</td>
<td>17.000</td>
</tr>
<tr>
<td>Mean</td>
<td>5352.0</td>
<td>4889.4</td>
<td>0.000</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>1741.0</td>
<td>1379.5</td>
<td>1.000</td>
</tr>
</tbody>
</table>
PCA example (contd.)

- Compute the mean and standard deviations of the variables:

  \[
  \bar{x}_s = \frac{1}{n} \sum_{i=1}^{n} x_{si} = \frac{96336}{18} = 5352.0
  \]

  \[
  \bar{x}_r = \frac{1}{n} \sum_{i=1}^{n} x_{ri} = \frac{88009}{18} = 4889.4
  \]

  \[
  s^2_{x_s} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{si} - \bar{x}_s)^2
  \]

  \[
  = \frac{1}{n-1} \left[ \left( \sum_{i=1}^{n} x_{si}^2 \right) - n \times \bar{x}_s^2 \right]
  \]

  \[
  = \frac{567119488 - 18 \times 5353^2}{17} = 1741.0^2
  \]

  \[
  s^2_{x_r} = \frac{462661024 - 18 \times 4889.4^2}{17} = 1379.5^2
  \]
PCA example (contd.)

- Normalize the variables to zero mean and unit standard deviation. The normalized values $x_s'$ and $x_r'$ are given by

$$x_s' = \frac{x_s - \bar{x}_s}{s_{x_s}} = \frac{x_s - 5352}{1741}$$

$$x_r' = \frac{x_r - \bar{x}_r}{s_{x_r}} = \frac{x_r - 4889}{1380}$$
PCA example (contd.)

- Compute the correlation among the variables:

\[ R_{x_s,x_r} = \frac{1}{n-1} \sum_{i=1}^{n} \frac{(x_{si} - \bar{x}_s)(x_{ri} - \bar{x}_r)}{s_{x_s}s_{x_r}} = 0.916 \]

- Prepare the correlation matrix:

\[
C = \begin{bmatrix}
1.000 & 0.916 \\
0.916 & 1.000
\end{bmatrix}
\]
PCA example (contd.)

- Compute the eigenvalues of the correlation matrix:
  
  By solving the *characteristic equation*:

  \[
  |\lambda I - C| = \begin{vmatrix}
  \lambda - 1 & -0.916 \\ 
  -0.916 & \lambda - 1 \\
  \end{vmatrix} = 0
  \]

  \[
  (\lambda - 1)^2 - 0.916^2 = 0
  \]

- The eigenvalues are 1.916 and 0.084.
PCA example (contd.)

- Compute the eigenvectors of the correlation matrix. The eigenvectors $q_1$ corresponding to $\lambda_1 = 1.916$ are defined by the following relationship:

\[
\{ C \} \{ q \}_1 = \lambda_1 \{ q \}_1
\]

or:

\[
\begin{bmatrix}
1.000 & 0.916 \\
0.916 & 1.000
\end{bmatrix}
\times
\begin{bmatrix}
q_{11} \\
q_{21}
\end{bmatrix}
= 1.916
\begin{bmatrix}
q_{11} \\
q_{21}
\end{bmatrix}
\]

or:

$q_{11} = q_{21}$
PCA example (contd.)

- Restricting the length of the eigenvectors to one:
  \[ q_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \quad q_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \]

  - Note: if the eigenvector for an eigenvalue is not unique, any of them can be used in PCA (without much impact on the results of PCA)

- Obtain principal factors by multiplying the eigenvectors by the normalized vectors: \( Y = [q_1; q_2]' \times X^* \)

\[
\begin{bmatrix}
  y_1 \\
  y_2
\end{bmatrix} = \begin{bmatrix}
  \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\
  \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}
\end{bmatrix} \begin{bmatrix}
  x_s - 5352 \\
  1741 \\
  x_r - 4889 \\
  1380
\end{bmatrix}
\]
PCA example (contd.)

- Compute the values of the principal factors

- Compute the sum and sum of squares of the principal factors
  - The sum must be zero
  - The sum of squares give the percentage of variation explained
    - The first factor explains $32.565/(32.565+1.435)$ or 95.7% of the variation
    - The second factor explains only 4.3% of the variation and can, thus, be ignored
PCA example (contd.)

- Plot the values of the principal factors
Workload characterization techniques

- Averaging
- Specifying dispersion
- Histograms: single-parameter, multi-parameter
- Principal Component Analysis
- Markov Models
- Clustering
Markov models

- Why?
  - In addition to the *number* of service requests of each type, we may well want to know the *order* of service requests

- Markov model: the next request depends only on the last request

- Described by a transition matrix:

<table>
<thead>
<tr>
<th>From/To</th>
<th>CPU</th>
<th>Disk</th>
<th>Terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>Disk</td>
<td>0.9</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>Terminal</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Transition matrix/probability

- Transition matrices can also be used
  - for application transitions: e.g., $P(\text{Link} | \text{Compile})$
  - to specify page-reference locality: $P(\text{Reference module } i | \text{Referenced module } j)$

- Given the same relative frequency of requests of different types, it is possible to realize the frequency with several different transition matrices
  - => If order is important, measure the transition probabilities directly on the real system
  - E.g., (next slide)
Example:
Two packet sizes: Small (80%), Large (20%)

- Case #1: An average of four small packets are followed by an average of one large packet, e.g., sssslsssslssss.

<table>
<thead>
<tr>
<th>Current Packet</th>
<th>Next packet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small</td>
</tr>
<tr>
<td>Small</td>
<td>0.75</td>
</tr>
<tr>
<td>Large</td>
<td>1</td>
</tr>
</tbody>
</table>
Example (contd.)

- Case #2: Eight small packets followed by two large packets
  
<table>
<thead>
<tr>
<th>Current Packet</th>
<th>Next packet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small</td>
</tr>
<tr>
<td>Small</td>
<td>0.875</td>
</tr>
<tr>
<td>Large</td>
<td>0.5</td>
</tr>
</tbody>
</table>

- Case #3: Generate a random number $x$ uniformly distributed between 0 and 1
  
  - If $x < 0.8$: generate a small packet;
  
  Otherwise, generate a large packet

<table>
<thead>
<tr>
<th>Current Packet</th>
<th>Next packet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small</td>
</tr>
<tr>
<td>Small</td>
<td>0.8</td>
</tr>
<tr>
<td>Large</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Workload characterization techniques

- Averaging
- Specifying dispersion
- Histograms: single-parameter, multi-parameter
- Principal Component Analysis
- Markov Models
- Clustering
Clustering

- To classify the (potentially) large number of components into a small number of \textit{classes} or \textit{clusters} s.t.
  - The components within a cluster are very similar to each other
  - One member from each class can be selected to represent the class and to study the effect of system design decisions on the entire class
Clustering steps

- Take a sample, that is, a subset of workload components
- Select workload parameters
- *Transform parameters, if necessary*
- *Remove outliers*
- *Scale all observations*
- *Select a distance measure*
- *Perform clustering*
- *Interpret results*
- Change parameters, or number of clusters, and repeat italicized steps
- Select representative components from each cluster
Sampling

- E.g., In one study, 2% of the population was chosen for analysis; later 99% of the population could be assigned to the clusters obtained.

Methods

- Random selection
- Scenario-driven: e.g.,
  - Select top consumers of a resource, when wanting to study the impact of the resource.
Parameter selection

- **Criteria:**
  - Impact on performance
  - Variance of parameters across clusters/components

- **Two candidate methods**
  - Redo clustering with one less parameter, and count the number of components that change cluster membership (note: domain knowledge helps in this case)
    - If changes are small, can remove the parameter
  - Principal component analysis: Identify factors (and hence parameters via weights) with the highest variance
Transformation

- If the distribution is highly skewed, consider a function of the parameter

- E.g., log of CPU time
  - Two programs taking 1 and 2 seconds are almost as different as those taking 1 and 2 milliseconds => the ratio of CPU time, rather than their difference, is more important
Outliers

- Outliers = *data points (of components)* with extreme parameter values

- They affect max/min/mean/variance values, thus affecting normalization (discussed next)

- Can exclude outlying components only if they do not consume a significant portion of system resources
  - Example, disk backup program can be excluded from a workload characterizing sites where backups are done a few times per month, but may not be excluded for cases where backups are done a few times per day
Data scaling

- Objective: to scale parameter values so that their relative values and ranges are approx. equal

- Method #1: Normalize to Zero Mean and Unit Variance: \( \{x_{ik}\} \) for k-th parameter
  \[
  x'_{ik} = \frac{x_{ik} - \bar{x}_k}{s_k}
  \]

- Method #2: Weights:
  \[
  x'_{ik} = w_k \cdot x_{ik}
  \]
  where \( w_k \) is proportional to relative importance or
  \[ w_k = 1/s_k \]
Data scaling (contd.)

- Method #3: Range Normalization:

\[ x'_{ik} = \frac{x_{ik} - x_{\text{min},k}}{x_{\text{max},k} - x_{\text{min},k}} \]

(+) no need to do square/square-root calculation

(-) easily affected by outliers

- Method #4: Percentile Normalization:

\[ x'_{ik} = \frac{x_{ik} - x_{2.5,k}}{x_{97.5,k} - x_{2.5,k}} \]
Distance metric

- Euclidean Distance: Given \( \{x_{i1}, x_{i2}, \ldots, x_{in}\} \) and \( \{x_{j1}, x_{j2}, \ldots, x_{jn}\} \)

\[
d = \left\{ \sum_{k=1}^{n} (x_{ik} - x_{jk})^2 \right\}^{0.5}
\]

- Most commonly used distance metric

- Weighted-Euclidean Distance:

\[
d = \sum_{k=1}^{n} \left\{ a_k (x_{ik} - x_{jk})^2 \right\}^{0.5}
\]

Here \( a_k, k=1,\ldots,n \) are suitably chosen weights for the \( n \) parameters

- Used if the parameters
  - have not been scaled, or
  - have significantly different levels of importance
Distance metric (contd.)

- **Chi-Square Distance:**

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

- Usually used in distribution fitting (e.g., distance between two multinomial distributions)
- Used only if \(x_{.k}\)'s are close to each other (e.g., has been normalized); otherwise, parameters with low values of \(x_{.k}\) get higher weights
Clustering techniques

- Goal: Partition into groups so the members of a group are as similar as possible, and different groups are as dissimilar as possible.

- Statistically, the intragroup variance should be as small as possible, and inter-group variance should be as large as possible.
  - Total Variance = Intra-group Variance “+” Inter-group Variance
    - Only need to worry about either inter- or intra-group variance.
Clustering tech. (contd.)

- **Nonhierarchical techniques**: Start with an arbitrary set of $k$ clusters, move members until the *intra-group* variance is minimum.

- **Hierarchical Techniques**:
  - Agglomerative: Start with $m$ clusters and merge.
  - Divisive: Start with one cluster and divide.

- Two popular techniques:
  - Minimum spanning tree method (agglomerative)
  - Centroid method (Divisive)
Minimum spanning tree (MST) method

- Start with $k = n$ clusters

- Find the centroid of the $i$-th cluster, $i = 1, 2, \ldots, k$

- Compute the inter-cluster distance matrix (based on centroids)

- Merge the nearest clusters

- Repeat italicized steps until all components are part of one cluster
Example of MST method

<table>
<thead>
<tr>
<th>Program</th>
<th>CPU Time</th>
<th>Disk I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

- Step 1: Consider five clusters with i-th cluster consisting solely of i-th program
- Step 2: The centroids are \{2, 4\}, \{3, 5\}, \{1, 6\}, \{4, 3\}, and \{5, 2\}
Example (contd.)

- Step 3: The Euclidean distance is:

<table>
<thead>
<tr>
<th>Program</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>$\sqrt{2}$</td>
<td>$\sqrt{5}$</td>
<td>$\sqrt{5}$</td>
<td>$\sqrt{13}$</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>$\sqrt{5}$</td>
<td>$\sqrt{5}$</td>
<td>$\sqrt{13}$</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td></td>
<td>$\sqrt{18}$</td>
<td></td>
<td>$\sqrt{32}$</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td></td>
<td></td>
<td>$\sqrt{2}$</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Step 4: Minimum inter-cluster distance = $\sqrt{2}$. Merge A+B, D+E
Example (contd.)

- Step 2: The centroid of cluster pair AB is \(\{(2+3) \div 2, (4+5) \div 2\}\), that is, \(\{2.5, 4.5\}\). Similarly, the centroid of pair DE is \(\{4.5, 2.5\}\)
Step 3: The distance matrix is:

<table>
<thead>
<tr>
<th>Program</th>
<th>AB</th>
<th>C</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>0</td>
<td>(\sqrt{4.5})</td>
<td>(\sqrt{8})</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>(\sqrt{24.5})</td>
<td>0</td>
</tr>
<tr>
<td>DE</td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Step 4: Merge AB and C.

Step 2: The centroid of cluster ABC is \(\{(2+3+1) \div 3, (4+5+6) \div 3\}\), that is, \{2, 5\}
Example (contd.)

- Step 3: The distance matrix is:

| Program | Program 
|---------|-------
| ABC     | 0     |
| DE      | √12.5 |

- Step 4: Minimum distance is \((12.5)^{0.5}\). Merge ABC and DE => Single Custer ABCDE
Dendogram

- Dendogram = Spanning Tree

- Purpose: Obtain clusters for any given maximum allowable intra-cluster distance
Centroid method

- Start with \( k = 1 \)
- Find the centroid and intra-cluster variance for the \( i \)-th cluster, \( i = 1, 2, \ldots, k \).
- Find the cluster with the highest variance and arbitrarily divide it into two clusters
  - Find the two components that are farthest apart, assign other components according to their distance from these points; OR
  - Place all components below a hyperplane crossing the centroid in one cluster and all components above the hyperplane in the other
- Adjust the points in the two new clusters until the inter-cluster distance between the two clusters is maximum
- Set \( k = k+1 \). Repeat italicized steps until \( k = m \)
Cluster interpretation

- Assign all measured components to the clusters
- Interpret clusters in functional terms (e.g., a business application), or label clusters by their resource demands (for example, CPU-bound, I/O-bound, and so forth)
- Clusters with very small populations and small total resource demands can be discarded
  - (Don't just discard a small cluster)
- Select one or more representative components from each cluster for use as test workload
  - # of representatives can be proportional to the cluster size, the total resource demands of the cluster, or any combination of the two
Problems with clustering

- The goal of minimizing intracluster variance (or maximizing intercluster variance) many lead to final clusters that are quite different from those visible to eyes.

(How to set the right clustering objective?)
Problems with clustering (contd.)

- The results of clustering are highly variable. No general rules for:
  - Selection of parameters
  - Scaling
  - Distance measure
- Labeling each cluster by functionality may be difficult
  - E.g., in one study, editing programs appeared in 23 different clusters
- May well require many repetitions of the analysis
Summary

- **Workload Characterization** = Models of workloads

- **Methods**
  - Averaging, specifying dispersion
  - Deal with high variance
    - Single parameter histogram, multi-parameter histograms
    - Classification of components based on “principal component analysis”
      (i.e., finding parameter combinations that explain the most variation)
  - Markov model
  - Clustering
    - Divide workloads in groups where each group can be represented by a single benchmark
Homework #1

1. [50 points] The CPU time and disk I/Os of seven programs are shown in Table below. Determine the equation for principal factors.

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Function</th>
<th>CPU Time</th>
<th>Number of I/Os</th>
</tr>
</thead>
<tbody>
<tr>
<td>TKB</td>
<td>Linker</td>
<td>14</td>
<td>2735</td>
</tr>
<tr>
<td>MAC</td>
<td>Assembler</td>
<td>13</td>
<td>253</td>
</tr>
<tr>
<td>COBOL</td>
<td>Compiler</td>
<td>8</td>
<td>27</td>
</tr>
<tr>
<td>BASIC</td>
<td>Compiler</td>
<td>6</td>
<td>27</td>
</tr>
<tr>
<td>PASCAL</td>
<td>Compiler</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>EDT</td>
<td>Text Editor</td>
<td>4</td>
<td>91</td>
</tr>
<tr>
<td>SOS</td>
<td>Text Editor</td>
<td>1</td>
<td>33</td>
</tr>
</tbody>
</table>
2. [50 points] Using a spanning-tree algorithm for cluster analysis, prepare a Dendogram for the data shown in Table below. Interpret your analysis results. (note: no unique solution.)

<table>
<thead>
<tr>
<th>Program Name</th>
<th>Function</th>
<th>CPU Time</th>
<th>Number of I/Os</th>
</tr>
</thead>
<tbody>
<tr>
<td>TKB</td>
<td>Linker</td>
<td>14</td>
<td>2735</td>
</tr>
<tr>
<td>MAC</td>
<td>Assembler</td>
<td>13</td>
<td>253</td>
</tr>
<tr>
<td>COBOL</td>
<td>Compiler</td>
<td>8</td>
<td>27</td>
</tr>
<tr>
<td>BASIC</td>
<td>Compiler</td>
<td>6</td>
<td>27</td>
</tr>
<tr>
<td>PASCAL</td>
<td>Compiler</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>EDT</td>
<td>Text Editor</td>
<td>4</td>
<td>91</td>
</tr>
<tr>
<td>SOS</td>
<td>Text Editor</td>
<td>1</td>
<td>33</td>
</tr>
</tbody>
</table>
Suggestion on solving homeworks

- Use tools such as Matlab whenever appropriate