Workload Characterization Techniques

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

These slides are available on-line at:
http://www.cse.wustl.edu/~jain/cse567-06/
Overview

- Terminology
- Components and Parameter Selection
- Workload Characterization Techniques: Averaging, Single Parameter Histograms, Multi-parameter Histograms, Principal Component Analysis, Markov Models, Clustering
- Clustering Method: Minimum Spanning Tree, Nearest Centroid
- Problems with Clustering
Terminology

- User = Entity that makes the service request
  = Workload Unit or Workload component

- Workload components:
  - Applications
  - Sites
  - User Sessions

- Workload parameters or Workload features: Measured quantities, service requests, or resource demands. For example: transaction types, instructions, packet sizes, source-destinations of a packet, and page reference pattern.
Components and Parameter Selection

- The workload component should be at the SUT interface.
- Each component should represent as homogeneous a group as possible. Combining very different users into a site workload may not be meaningful.
- Domain of the control affects the component: Example: mail system designer are more interested in determining a typical mail session than a typical user session.
- Do not use parameters that depend upon the system, e.g., the elapsed time, CPU time.
Components (Cont)

- Characteristics of service requests:
  - Arrival Time
  - Type of request or the resource demanded
  - Duration of the request
  - Quantity of the resource demanded, for example, pages of memory
- Exclude those parameters that have little impact.
Workload Characterization Techniques

1. Averaging
2. Single-Parameter Histograms
3. Multi-parameter Histograms
4. Principal Component Analysis
5. Markov Models
6. Clustering
Averaging

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

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

- Coefficient Of Variation: \[ s / \bar{x} \]

- Mode (for categorical variables): Most frequent value

- Median: 50-percentile
## Case Study: Program Usage in Educational Environments

<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>2.19 seconds</td>
<td>40.23</td>
</tr>
<tr>
<td>Elapsed time</td>
<td>73.90 seconds</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 kilobytes</td>
<td>82.41</td>
</tr>
<tr>
<td>Size of direct writes</td>
<td>1.25 kilobytes</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 kilobytes</td>
<td>21.01</td>
</tr>
<tr>
<td>Size of direct reads</td>
<td>2.20 kilobytes</td>
<td></td>
</tr>
<tr>
<td>Number of buffered writes</td>
<td>52.84</td>
<td>11.80</td>
</tr>
<tr>
<td>Buffered write bytes</td>
<td>978.04 bytes</td>
<td>9.98</td>
</tr>
<tr>
<td>Size of buffered writes</td>
<td>18.51 bytes</td>
<td></td>
</tr>
</tbody>
</table>

- **High Coefficient of Variation**
## Characteristics of an Average Editing Session

<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>
<tr>
<td>Size of direct reads</td>
<td>0.98 kilo-bytes</td>
<td></td>
</tr>
<tr>
<td>Number of buffered writes</td>
<td>199.06</td>
<td>4.30</td>
</tr>
<tr>
<td>Buffered write bytes</td>
<td>3314.95 bytes</td>
<td>3.04</td>
</tr>
<tr>
<td>Size of buffered writes</td>
<td>16.65 bytes</td>
<td></td>
</tr>
</tbody>
</table>

- Reasonable variation
Single Parameter Histograms

- \( n \) buckets \( \times \) \( m \) parameters \( \times \) \( k \) components values.
- Use only if the variance is high.
- Ignores correlation among parameters.
Difficult to plot joint histograms for more than two parameters.
**Principal Component Analysis**

- **Key Idea**: Use a weighted sum of parameters to classify the components.

- Let $x_{ij}$ denote the $i$th parameter for $j$th component.

  $$y_j = \sum_{i=1}^{n} w_i x_{ij}$$

- Principal component analysis 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.

- The factors are ordered. First factor explains the highest percentage of the variance.
Principal Component Analysis (Cont)

- Statistically:
  - 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, that is, their inner product is zero:
    \[ <y_i, y_j> = \sum_k a_{ik} a_{kj} = 0 \]
    This is equivalent to stating that \( y_i \)'s are uncorrelated to each other.
  - The y's form an ordered set such that \( y_1 \) explains the highest percentage of the variance in resource demands.
Finding Principal Factors

- Find the correlation matrix.
- Find the eigen values of the matrix and sort them in the order of decreasing magnitude.
- Find corresponding eigen vectors. These give the required loadings.
## Principal Component Example

<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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Summations</th>
<th>Values</th>
<th>Mean</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sum x$</td>
<td>96336</td>
<td>610.5</td>
<td>1741.0</td>
</tr>
<tr>
<td>$\sum x'^2$</td>
<td>567119488</td>
<td>47104.19</td>
<td>1379.5</td>
</tr>
<tr>
<td>$\sum x^2$</td>
<td>462661024</td>
<td>2883.9</td>
<td>0.000</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>0.000</td>
</tr>
</tbody>
</table>
Principal Component Example

- Compute the mean and standard deviations of the variables:

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

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

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

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

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

Similarly:

\[
S_{x_r}^2 = \frac{462661024 - 18 \times 4889.4^2}{17} = 1379.5^2
\]
Principal Component (Cont)

- Normalize the variables to zero mean and unit standard deviation. The normalized values $x'_s$ and $x'_r$ are given by:

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

$$
x'_r = \frac{x_r - \bar{x}_r}{s_{x_r}} = \frac{x_r - 4889}{1380}
$$
Principal Component (Cont)

- Compute the correlation among the variables:

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

- Prepare the correlation matrix:

\[
C = \begin{bmatrix}
1.000 & 0.916 \\
0.916 & 1.000
\end{bmatrix}
\]
Principal Component (Cont)

- Compute the eigen values of the correlation matrix: By solving the characteristic equation:

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

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

- The eigen values are 1.916 and 0.084.
Principal Component (Cont)

- Compute the eigen vectors of the correlation matrix. The eigen vectors $q_1$ 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:

$$q_{11}=q_{21}$$
Principal Component (Cont)

- Restricting the length of the eigen vectors to one:

\[
\begin{align*}
q_1 &= \begin{bmatrix}
\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}}
\end{bmatrix} \\
q_2 &= \begin{bmatrix}
\frac{1}{\sqrt{2}} \\
-\frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}}
\end{bmatrix}
\end{align*}
\]

- Obtain principal factors by multiplying the eigen vectors by the normalized vectors:

\[
\begin{bmatrix}
y_1 \\
y_2
\end{bmatrix} = \begin{bmatrix}
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\
\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}}
\end{bmatrix} \begin{bmatrix}
x_s - 5352 \\
x_r - 1741
\end{bmatrix}
\]

\[
\begin{bmatrix}
x_r - 4889 \\
x - 1380
\end{bmatrix}
\]

- Compute the values of the principal factors.
- Compute the sum and sum of squares of the principal factors.
Principal Component (Cont)

- 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.
Markov Models

- **Markov**
  - 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 matrices can be used also for application transitions. E.g., $P(\text{Link} | \text{Compile})$

- Used to specify page-reference locality.
  - $P(\text{Reference module } i | \text{Referenced module } j)$
Transition Probability

- 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.

- Example: Two packet sizes: Small (80%), Large (20%)
  - An average of four small packets are followed by an average of one big packet, e.g., sssbsbsbsbssss.

<table>
<thead>
<tr>
<th>Current Packet</th>
<th>Next packet</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Small</td>
<td>Large</td>
</tr>
<tr>
<td>Small</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>Large</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Transition Probability (Cont)

- Eight small packets followed by two big 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>

- Generate a random number $x$.  
  $x \leq 0.8 \Rightarrow$ generate a small packet;  
  otherwise generate a large packet.
Clustering

Disk I/O’s vs CPU time

- Clusters of data points indicate similar behavior of disk I/O's and CPU time.
Clustering Steps

1. Take a sample, that is, a subset of workload components.
2. Select workload parameters.
3. Select a distance measure.
4. Remove outliers.
5. Scale all observations.
6. Perform clustering.
7. Interpret results.
8. Change parameters, or number of clusters, and repeat steps 3-7.
9. Select representative components from each cluster.
1. Sampling

- In one study, 2% of the population was chosen for analysis; later 99% of the population could be assigned to the clusters obtained.
- Random selection
- Select top consumers of a resource.
2. Parameter Selection

- Criteria:
  - Impact on performance
  - Variance

- Method: Redo clustering with one less parameter

- Principal component analysis: Identify parameters with the highest variance.
3. Transformation

- If the distribution is highly skewed, consider a function of the parameter, e.g., log of CPU time
4. Outliers

- Outliers = data points with extreme parameter values
- Affect normalization
- Can exclude only if they do not consume a significant portion of the system resources. Example, backup.
5. Data Scaling

1. Normalize to Zero Mean and Unit Variance:

\[ x'_{ik} = \frac{x_{ik} - \bar{x}_k}{s_k} \]

2. Weights:

\[ x'_{ik} = w_k \cdot x_{ik} \]

\[ w_k \propto \text{relative importance} \quad \text{or} \quad w_k = \frac{1}{s_k} \]

3. Range Normalization:

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

Affected by outliers.
Data Scaling (Cont)

- Percentile Normalization:

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

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

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

2. Weighted-Euclidean Distance:

\[
d = \sum_{k=1}^{n} \left\{ a_k (x_{ik} - x_{jk})^2 \right\}^{0.5}
\]
Here \( a_k \), \( k = 1, 2, \ldots, n \) are suitably chosen weights for the \( n \) parameters.

3. Chi-Square Distance:

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

- The Euclidean distance is the most commonly used distance metric.

- The weighted Euclidean is used if the parameters have not been scaled or if the parameters have significantly different levels of importance.

- Use Chi-Square distance only if $x_k$'s are close to each other. 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
Clustering Techniques (Cont)

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

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

- Two popular techniques:
  - Minimum spanning tree method (agglomerative)
  - Centroid method (Divisive)
Minimum Spanning Tree-Clustering Method

1. Start with \( k = n \) clusters.
2. Find the centroid of the \( i^{th} \) cluster, \( i=1, 2, \ldots, k \).
3. Compute the inter-cluster distance matrix.
4. Merge the the nearest clusters.
5. Repeat steps 2 through 4 until all components are part of one cluster.
Minimum Spanning Tree Example

<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 ith cluster consisting solely of ith program.
- Step 2: The centroids are \{2, 4\}, \{3, 5\}, \{1, 6\}, \{4, 3\}, and \{5, 2\}.
Spanning Tree Example (Cont)

- 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>$\sqrt{18}$</td>
<td>$\sqrt{32}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>$\sqrt{2}$</td>
<td></td>
<td></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.
Step 2: The centroid of cluster pair AB is 
\[\{(2+3)/2, (4+5)/2\}\], that is, \{2.5, 4.5\}. 
Similarly, the centroid of pair DE is \{4.5, 2.5\}.
Spanning Tree Example (Cont)

- 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>$\sqrt{10.25}$</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>$\sqrt{24.4}$</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, $\{2, 5\}$. 
Spanning Tree Example (Cont)

- **Step 3:** The distance matrix is:

<table>
<thead>
<tr>
<th>Program</th>
<th>ABC</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABC</td>
<td>0</td>
<td>√12.5</td>
</tr>
<tr>
<td>DE</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Step 4:** Minimum distance is 12.5.
  Merge ABC and DE ⇒ Single Custer ABCDE
- Dendogram = Spanning Tree
- Purpose: Obtain clusters for any given maximum allowable intra-cluster distance.
Nearest Centroid Method

- Start with $k = 1$.
- Find the centroid and intra-cluster variance for $i^{th}$ 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.
- Place all components below the centroid in one cluster and all components above this hyper plane 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 steps 2 through 4 until $k = n$. 
Cluster Interpretation

- Assign all measured components to the clusters.
- Clusters with very small populations and small total resource demands can be discarded.
  (Don't just discard a small cluster)
- 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.
- Select one or more representative components from each cluster for use as test workload.
Problems with Clustering
Goal: Minimize variance.

The results of clustering are highly variable. No rules for:
- Selection of parameters
- Distance measure
- Scaling

Labeling each cluster by functionality is difficult.
- In one study, editing programs appeared in 23 different clusters.

Requires many repetitions of the analysis.
Summary

- Workload Characterization = Models of workloads
- Averaging, Single parameter histogram, multi-parameter histograms, …
- Principal component analysis consists of finding parameter combinations that explain the most variation
- Clustering: divide workloads in groups that can be represented by a single benchmark
Exercise 6.1

- 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>

Table 1: Data for Principal Component Exercise
Exercise 6.2

Using a spanning-tree algorithm for cluster analysis, prepare a Dendogram for the data shown in Table below. Interpret the result of your analysis.

Table 1: Data for Principal Component Exercise

<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>
Homework

- Read chapter 6
- Submit answers to exercises 6.1 and 6.2