King Fahd University of Petroleum & Minerals
Computer Engineering Dept

COE 587 – Performance Evaluation And Analysis

Term 152

Dr. Ashraf S. Hasan Mahmoud

Rm 22-420

Ext. 1724

Email: ashraf AT kfupm DOT edu DOT sa

Slides are based on the textbook:

R. Jain, "Art of Computer Systems Performance Analysis," Wiley, 1991, ISBN:0471503363

Book website:
http://www.cse.wustl.edu/~jain/books/perfbook.htm

Measurement Techniques And Tools – Topics

- What are the different types of workloads?
- Which workloads are commonly used by other analysts?
- How are the appropriate workload types selected?
- How is the measured workload data summarized?
- How is the system performance monitored?
- How can the desired workload be placed on the system in a controlled manner?
- How are the results of the evaluation presented?

Chapter 4: Types of Workloads

- **Skipped**
  - Made specific for CPU/Instruction set performance evaluation and benchmarking
  - Subsequent chapter (Chapter 5) handles networking related material.

Chapter 5: The Art of Workload Selection

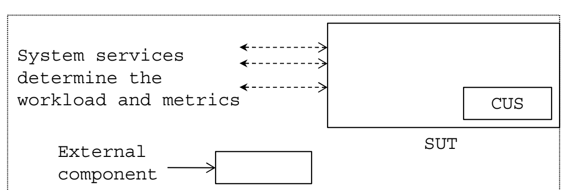
- **Workload selection is the MOST crucial step in any performance evaluation project**
- **Considerations:**
  - **Services exercised**
  - **Level of detail**
  - **Representativeness**
  - **Timeliness**
- **Minor considerations: Loading level, impact of other components, and repeatability.**

Services Exercised

- **View the system as a service provider**
- **System under test (SUT) – complete set of components that are being purchased or designed**
- **Component under study (CUS) – specific component in the SUT whose alternative are being considered**
- **Example – SUT = CPU, CUS = ALU**
- **SUT → System; CUS → component**

- **The workloads are primarily specified by the SUT**
- **The metrics chosen should reflect the performance of services provided at the system level and not at the component level**

- **Example: Two CPUs – use MIPS**
- **Example: Two time-sharing systems – use transactions/sec**



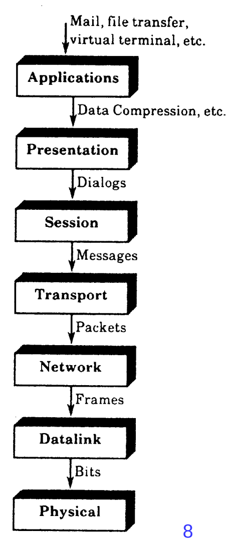
The SUT and CUS

Services Exercised – Cont'd

- **Summary**
  - **Requests at the service-interface level of the SUT should be used to specify or measure the workload**
  - **Clear distinction between SUT and CUS**

Example 5.1

- **Compare two networks**
- **ISO/OSI 7 layers model**
- **Different workloads for different layers (services)**
  - **Physical – bits transmitted**
  - **Data link – frames**
  - **Network – packets**
  - **Transport – messages**
  - **Application – mail, file transfer, etc.**



++++

Level of Detail

- List of possible levels of detail:
  1. Most frequently used request
  2. Frequency of request types
  3. Time-stamped sequence of requests (e.g trance)
  4. Average resource demand
  5. Distribution of resource demands
- The least detailed are (1) – may be as an initial step
- In (4), the request "presents" load to the system – e.g. a user required an average CPU time of 50 milliseconds.
  - Typical for analytical studies
- Sometimes the average demand of a request may not be sufficient – the actual distribution is needed as in (5)

Representativeness

- Test workload should be representative of the real application.
- Match workload (requests) to actual application in terms of
  - Arrival rate
  - Resource demands
  - Resource usage profile

Timeliness

- Workload should follow changes in usage pattern in a timely fashion
  - Telephone network (old) – symmetric traffic
  - Internet (new) – asymmetric traffic
- Real users behavior is a moving and fuzzy target
  - Users tend to focus on services where the system response is optimal
- Interdepedence of system design and workload – specially for systems under design
  - A system optimized for one or more workloads can not be guaranteed to operate efficiently in other environments

Chapter 6: Workload Characterization

- Workload component or workload unit → user
- Workload parameters (features): measured quantities, service requests, or resource demands that are used to model or characterize the workload
- Example of workload parameters: transactions types, instructions, packet sizes, source/destinations of a packet, etc.
- Techniques to characterize workloads
  1. Averaging
  2. Specifying dispersion
  3. Single-parameter histograms
  4. Multiparameter histograms
  5. Principle component analysis
  6. Markov models
  7. Clustering

Averaging

- Let $\{x_1, x_2, \ldots, x_n\}$ be n observed values of a workload parameter, the arithmetic mean $\bar{x}$ is given by

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- There are cases where the use of the mean is inappropriate and the median, mode, geometric mean, or harmonic mean should be used – More on this in Chapter 12
- E.g. For categorical parameters, then the most frequent value (the mode) should be used – Packet destinations are A, B, and C → average has no meaning, while the mode (most frequently used address) has real meaning.

Specifying Dispersion

- The average does not reflect variability in the data
- Variability is specified by the variance, s^2, which is given by

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2$$

- The standard deviation, s, is the square root of the variance.
- Coefficient of variation (COV) is the radio of standard deviation to the mean, i.e.

$$\text{C.O.V} = s/\bar{x}$$

Specifying Dispersion – cont'd

- Other alternative for specifying variability (discussed in Chapter 12):
  - Range (min and max)
  - 10- and 90- percentiles,
  - Semi-interquartile, and
  - Mean absolute deviation
- Zero C.O.V. → variance is zero or parameter is constant
- High C.O.V. → high variance, i.e. the mean alone is not sufficient
  - Maybe you should consider classifying the data into different classes (histogram)

Study Case 6.1

- Resource demands for various programs on six university sites were measured for 6 months.

- Table 6.1 – shows results for all programs (applications) – note the high COV
- Table 6.2 – shows results for all editors in the same data - note the COV is much lower
- Therefore, perhaps is not a good approach to lump all applications data together!!

TABLE 6.1 Workload Characterization Using Average Values

Data	Average	Coefficient of Variation
CPU time (VAX-11/780)	2.19 seconds	40.23
Number of direct writes	8.20	53.59
Direct-write bytes	10.21 kbytes	82.41
Number of direct reads	22.64	25.65
Direct-read bytes	49.70 kbytes	21.01

TABLE 6.2 Characteristics of an Average Editing Session

Data	Average	Coefficient of Variation
CPU time (VAX-11/780)	2.57 seconds	3.54
Number of direct writes	19.74	4.33
Direct-write bytes	13.46 kbytes	3.87
Number of direct reads	37.77	3.73
Direct-read bytes	36.93 kbytes	3.16

Single-Parameter Histograms

- **Histogram – shows the relative frequencies of various values of a parameter.**
  - **Divide the parameter range into subranges (buckets or cells)**
  - **Count observations that fall within each subrange**
- **Usage in measurement or simulation – to generate test workload**
- **Usage in analysis – to fit a probability distribution and to verify/validate distributions.**
- **Key shortcoming – correlation between parameters is not accounted for.**

Single-Parameter Histograms – cont'd

- **Example – short job require less CPU and have typical low I/O activity**
- **If one designs a workload based on single parameter (CPU) histogram, one produce short jobs with high I/O activity, a workload which is not realistic**
- **Solution: multiparameter histograms**

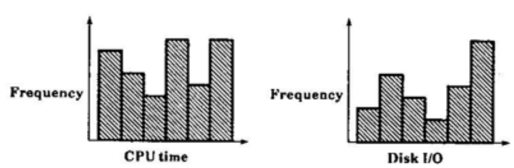


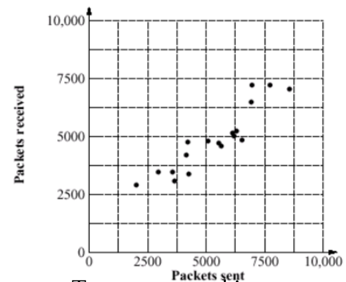
FIGURE 6.1 Single-parameter histograms of CPU time and disk I/O.

TABLE 6.3 Tabular Representation of a Single-Parameter Histogram

	CPU Time (milliseconds)				Number of Disk I/O			
Program	0–5	6–10	11–15	15+	0–20	21–40	41–60	60+
DOVERSEND	...	...	...	...	...	...	...	...
EMACS	...	...	...	...	...	...	...	...
MAIL	...	...	...	...	...	...	...	...
SCRIBE	...	...	...	...	...	...	...	...
PRESSIFY	...	...	...	...	...	...	...	...
DIRECTORY	...	...	...	...	...	...	...	...
TELNET	...	...	...	...	...	...	...	...

Multiparameter Histograms

- **Used when there is (significant) correlation between different workload parameters**
- ***n*-dimensional matrix (or histogram) is used to describe the distribution of n workload parameters**
- **It is difficult to plot joint histograms for more than two parameters.**
- **Too detailed ➔ Rarely used!!**



Two parameter histogram

Principle-Component Analysis (PCA)

- Goal: Use weighted sum of parameters to classify components
- Often a weighted sum such as $y_j = \sum_{j=1}^{n} w_i x_{ij}$ is used to this purpose

  where w_i is the weight for the i^{th} parameter for the j^{th} component
- But how to decide on the weights?
- The PCA procedure finding the weights w_i's such that y_j's provides maximum discrimination among the components.

Principle-Component Analysis (PCA)

- Let the n parameters be $\{x_1, x_2, \ldots, x_n\}$
- The PCA produces a set of FACTORS $\{y_1, y_2, \ldots, y_n\}$ such that
  - 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. inner product is zero)

$$\langle y_i, y_j \rangle = \sum_k a_{ik} a_{kj} = 0$$

This is equivalent to stating that the y's are uncorrelated

  - The y's form an ordered set such that y_1 explain the highest percentage of the variance, y_2 explains a lower percentage of the variance, and so forth.

Principle-Component Analysis (PCA)

- How to find the principle factors?
  - Find the parameters correlation matrix, C.
  - Find the eigen values, λ's, of the matrix and sort them in the order of decreasing magnitude.
  - Find corresponding eigen vectors (q's).
    - These give the required loadings (a_{ij}'s).

- For the set of n parameters $\{x_1, x_2, \ldots, x_n\}$, the correlation matrix C is an n by n matrix whose sr^{th} element is given by $R_{xs,xr}$

$$R_{x_s,x_r} = \frac{(1/n)\sum_{i=1}^{n}(x_{si} - \bar{x}_s)(x_{ri} - \bar{x}_r)}{S_{x_s} S_{x_r}}$$

where S_{xs} and S_{xr} are the standard deviations for the parameter x_s and x_r, respectively.

Principle-Component Analysis (PCA) – cont'd

- **Example 6.1:** The number of packets sent and received, denoted by xs and xr, respectively, by various stations on a local-area network were measured. The observed numbers are as follows:

```
xs = [7718   6958   8551   6924   6298
      6120   6184   6527   5081   4216
      5532   5638   4147   3562   2955
      4261   3644   2020];
xr = [7258   7232   7062   6526   5251
      5158   5051   4850   4825   4762
      4750   4620   4229   3497   3480
      3392   3120   2946];
```

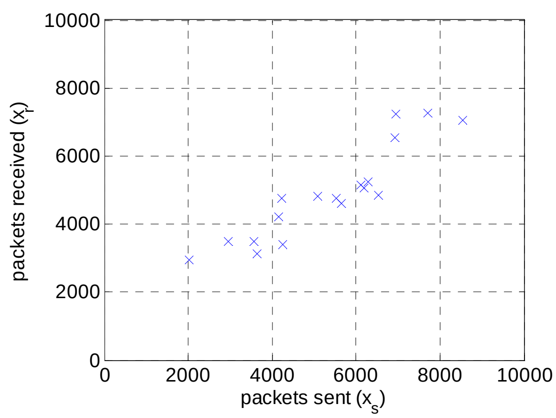
1) Generate a scatter plot from the data – Comment on the correlation between the two sequences
2) Carry on the PCA procedure to produce the principle factors
3) Plot the new (transformed) data – Comment on the correlation between the two new sequences

Principle-Component Analysis (PCA) – cont'd

Solution:

1) Scatter plot of original data – as shown in figure

It can be observed that the data is highly correlated. There is almost a linear relationship between Xs and Xr



Principle-Component Analysis (PCA) – cont'd

Solution:

2) The following are the steps to carry on the PCA procedure.

a) Compute the mean and standard deviation for Xs and for Xr

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

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

Principle-Component Analysis (PCA) – cont'd

Solution:

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

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

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

Similarly for Xr:

$$s_{x_r}^2 = \frac{462661024 - 18 \times 4889.4^2}{17} = 1379.5^2$$

Principle-Component Analysis (PCA) – cont'd

Solution:

The corresponding standard deviations are

$$s_{x_s} = 1741.0 \qquad s_{x_r} = 1379.5$$

b) Normalize the variables Xs and Xr to zero mean unit standard deviation

Define and

$$x'_s = \frac{x_s - \bar{x}_s}{s_{x_s}} \qquad x'_r = \frac{x_r - \bar{x}_r}{s_{x_r}}$$

Using Matlab Xss can be computed as in `(Xs-mean(Xs))/std(Xs)`

same for Xrr – refer to source code.

Principle-Component Analysis (PCA) – cont'd

Solution:

This should be 1/(*n*-

c) Compute the correlation matrix:

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

Note that $R_{x_s,x_s} = 1$ and $R_{x_r,x_r} = 1$

The correlation matrix is given by

$$\mathbf{C} = \begin{bmatrix} 1.000 & 0.916 \\ 0.916 & 1.000 \end{bmatrix}$$

Using matlab, one can produce the correlation matrix using the command "`C = corrcoef(Xs, Xr)` "

Principle-Component Analysis (PCA) – cont'd

Solution:

d) Compute the eigenvalues of the correlation matrix C by solving the characteristic equation for the matrix C.

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

$$(\lambda - 1)^2 - 0.916^2 = 0$$

This means the eigenvalues are: λ1 = 1.916 and λ2 = 0.084.

Using Matlab, the characteristic equation for the matrix C can be computed using: "`poly(C)`" – the returned result is a vector corresponding to the coefficients of the characteristic equation. i.e. `[1.0000 -2.0000 0.1617]`

Note that using Matlab one can obtain the eigenvalues directly without explicitly obtaining the characteristic equation. The command "[V, D] = eign(C)" returns a matrix V whose columns are the eigenvectors and a diagonal matrix D with the eigenvalues as the diagonal elements are in an **ascending** order. Refer to source code.

⌘ Finally, it should be observed that since the solution in the textbook obtains the eigenvectors in a **descending** order, then the matlab code needs to reverse order of the eigenvectors to obtain the same order for the principle factors in the textbook.

Principle-Component Analysis (PCA) – cont'd

Solution:

d) Compute the eigenvectors of the matrix C: q1 and q2.

Let q1 correspond to λ1, then C q1 = λ1 q1,

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

Or q11 = q21

Now if the vector q1 has length equal to 1, then $\mathbf{q}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

Similarly, the vector q2 is given by $\mathbf{q}_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$

Principle-Component Analysis (PCA) – cont'd

Solution:

e) The principle factors are obtained as follows:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{x_s - 5352}{1741} \\ \frac{x_r - 4889}{1380} \end{bmatrix}$$

f) Compute the values by substituting the in the formula above. The values are as shown in the table.

Principle-Component Analysis (PCA) – cont'd

Solution:

g) Compute the sum and sum of squares of the principle factors.

- The sum of squares give the percentage of variation explained.

- Therefore, y1 explains 32.565/(32.565+1.435) = 95.7% of the variation, while y2 explains only 4.3% of the variation.

i	x_s	x_r	x_s'	x_r'	y_1	y_2
1	7718	7258	+1.359	+1.717	+2.175	+0.253
2	6958	7232	+0.922	+1.698	+1.853	+0.549
3	8551	7062	+1.837	+1.575	+2.413	-0.186
4	6924	6526	+0.903	+1.186	+1.477	+0.200
5	6298	5251	+0.543	+0.262	+0.570	-0.199
6	6120	5158	+0.441	+0.195	+0.450	-0.174
7	6184	5051	+0.478	+0.117	+0.421	-0.255
8	6527	4850	+0.675	-0.029	+0.457	-0.497
9	5081	4825	-0.156	-0.047	-0.143	+0.077
10	4216	4762	-0.652	-0.092	-0.527	+0.396
11	5532	4750	+0.103	-0.101	+0.002	-0.145
12	5638	4620	+0.164	-0.195	-0.022	-0.254
13	4147	4229	-0.692	-0.479	-0.828	+0.151
14	3562	3497	-1.028	-1.009	-1.441	+0.013
15	2955	3480	-1.377	-1.022	-1.696	+0.251
16	4261	3392	-0.627	-1.085	-1.211	-0.324
17	3644	3120	-0.981	-1.283	-1.601	-0.213
18	2020	2946	-1.914	-1.409	-2.349	+0.357
Sum x	96336	88009	+0.0	+0.000	+0.000	+0.000
Sum x2	567119474	462660973	+17.0	+17.000	+32.565	+1.435
mean	+5352.0	+4889.4	+0.000	+0.000	+0.000	+0.000
std	+1741.0	+1379.5	+1.000	+1.000	+1.384	+0.290

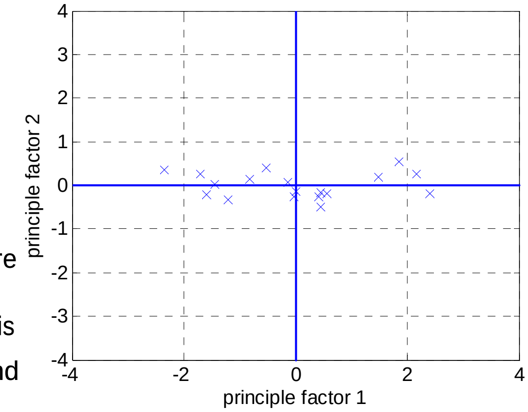
Principle-Component Analysis (PCA) – cont'd

Solution:

Using matlab – the previous procedure can be done as follows:

```
C     = corrcoef(Xs, Xr);
[V,L] = eig(C);
Y     = V*[Xss; Xrr]; Y = Y';
Y1    = Y(:,2); Y2 = Y(:,1);
```

3) Scatter plot of the transformed data – as shown in figure



It can be observed that there is little correlation between PF1 and PF2 (i.e. PF1 does not depend on PF2). There is almost a zero slope linear relationship between PF1 and PF2

Principle-Component Analysis (PCA) – cont'd

```
%
% Example 6.1 - textbook page 77-80
clear all
FontSize = 14; MarkerSize = 9; LineWidth = 2;
Xs = [7718 6958 8551 6924 6298 6120 6184 6527 5081 4216 5532 5638 4147 3562
2955 4261 3644 2020];
Xr = [7258 7232 7062 6526 5251 5158 5051 4850 4825 4762 4750 4620 4229 3497
3480 3392 3120 2946];

n = length(Xs);

XsSum = sum(Xs); Xs2Sum = sum(Xs.*Xs);
XrSum = sum(Xr); Xr2Sum = sum(Xr.*Xr);

Xsbar = mean(Xs); XsVar = var(Xs); XsStd = std(Xs);
Xrbar = mean(Xr); XrVar = var(Xr); XrStd = std(Xr);

Xss = (Xs - Xsbar)/XsStd;
Xrr = (Xr - Xrbar)/XrStd;

XssSum = sum(Xss); Xss2Sum = sum(Xss.*Xss);
XrrSum = sum(Xrr); Xrr2Sum = sum(Xrr.*Xrr);
Xssbar = mean(Xss); XssStd = std(Xss);
Xrrbar = mean(Xrr); XrrStd = std(Xrr);
%
% form the correlation matrix and get the eignvalue
C = corrcoef(Xs, Xr);
P = poly(C); % get the coefficients for the characteristic equation
L = roots(P); % this gets the roots for the characteristic equation or the
              % eignvalues - one can simply do L = eig(C);
[V,L] = eig(C);
%
% Compute the principle factors - To get the same vectors y1 and y2 as in
% the example, the eignvalues/vectors must be sorted from max to min
Y = V*[Xss; Xrr]; Y = Y';
Y1 = Y(:,2); Y2 = Y(:,1); % y1 now corresponds to the largest eign value
Y1Sum = sum(Y1); Y12Sum = sum(Y1.*Y1);
Y2Sum = sum(Y2); Y22Sum = sum(Y2.*Y2);
% Get stats for the principle factors
Y1bar = mean(Y1); Y1Std = std(Y1);
Y2bar = mean(Y2); Y2Std = std(Y2);
%
% Produce the table in the textbook
fprintf('-------------------------------------------------------------------\n');
fprintf(' i          x_s          x_r     x_s''     x_r''     y_1      y_2\n');
fprintf('-------------------------------------------------------------------\n');
for i=1:n
    fprintf('%2d     %9d   %9d   %+5.3f   %+5.3f   %+5.3f   %+5.3f\n', ...
        i, Xs(i), Xr(i), Xss(i), Xrr(i), Y1(i), Y2(i));
end
fprintf('-------------------------------------------------------------------\n');
fprintf('Sum x   %9d  %9d  %+5.1f     %+5.3f  %+5.3f    %+5.3f\n', XsSum, XrSum,
XssSum, XrrSum, Y1Sum, Y2Sum);
fprintf('Sum x2  %9d  %9d  %+5.1f    %+5.3f  %+5.3f    %+5.3f\n', Xs2Sum,
Xr2Sum, Xss2Sum, Xrr2Sum, Y12Sum, Y22Sum);
fprintf('mean     %+5.1f     %+5.1f  %+5.3f    %+5.3f  %+5.3f    %+5.3f\n',
Xsbar, Xrbar, Xssbar, Xrrbar, Y1bar, Y2bar);
fprintf('std      %+5.1f     %+5.1f  %+5.3f    %+5.3f  %+5.3f    %+5.3f\n',
XsStd, XrStd, XssStd, XrrStd, Y1Std, Y2Std);
fprintf('-------------------------------------------------------------------\n');
%
% Plot the old data
figure(1); set(gca, 'FontSize', FontSize);
h = plot(Xs, Xr,'x');
xlabel('packets sent (x_s)'); ylabel('packets received (x_r)');
set(h,'MarkerSize', MarkerSize);
axis([0 10000 0 10000]); grid;
%
% Plot the new (transformed) data
figure(2); set(gca, 'FontSize', FontSize);
h = plot(Y1, Y2,'x');
xlabel('principle factor 1'); ylabel('principle factor 2');
set(h,'MarkerSize', MarkerSize);
line([0 0],[-4 4],'LineWidth', LineWidth, 'LineStyle', '--');
line([-4 4],[0 0],'LineWidth', LineWidth, 'LineStyle', '--');
axis([-4 4 -4 4]); grid;
```

reversing the order of the principle factors

Markov Models

- If the next system state depends only on the current state ➔ Markov model
  - i.e. order of requests is as important as their intensity
- Typically used in queueing analysis
- Characterized by a probability transition matrix
- **Example:** The table below shows the transition probability matrix for a job moving between the CPU, the disk and the terminal.
  - After each visit to the CPU, the job moves to the disk with probability 0.3 or to the terminal with probability equal to 0.1.

From/To	CPU	Disk	Terminal
CPU	0.6	0.3	0.1
Disk	0.9	0	0.1
Terminal	1	0	0