Methodology

easy but important
1. What is performance evaluation about?
   2. Factors
   3. The Scientific Method
   4. Patterns
Two Examples

1. Compare Windows versus Linux
2. Evaluate the power consumption of a telecom «box»

How would you tackle these problems?
What do you need to be careful about?
Example 2.5: Windows versus Linux. Assume you want to compare Windows versus Linux. Chen and co-authors did it in [7]. They use as metric: number of cycles, instructions, data read/write operations. The load was generated by various benchmarks: “syscall” generates elementary operations (system calls); “memory read” generates references to an array; an application benchmark runs a popular application (ghostview).
Example 2

---

**Example 1.3: Power Consumption.** The electrical power consumed by a computer or telecom equipment depends on how efficiently the equipment can take advantage of low activity periods to save energy. One operator proposes the following metric as a measure of power consumption [2]:

\[ P_{\text{Total}} = 0.35P_{\text{max}} + 0.4P_{50} + 0.25P_{\text{sleep}} \]

where \( P_{\text{Total}} \) is the power consumption when the equipment is running at full load, \( P_{50} \) when it is submitted to a load equal to 50% of its capacity and \( P_{\text{sleep}} \) when it is idle. The weights (0.35, 0.4 and 0.25) mean for example that we assume that the full load condition occurs during 35% of the time.
Metric

Define a metric; examples

- Response time
- Power consumption
- Throughput

Define operational conditions under which metric is measured (« Viewpoint », see Chapter 11)

---

**Example 2.5: Windows versus Linux.** Assume you want to compare Windows versus Linux. Chen and co-authors did it in [7]. They use as metric: number of cycles, instructions, data read/write operations. The load was generated by various benchmarks: “syscall” generates elementary operations (system calls); “memory read” generates references to an array; an application benchmark runs a popular application (ghostview).
**Example 1.3: Power Consumption.** The electrical power consumed by a computer or telecom equipment depends on how efficiently the equipment can take advantage of low activity periods to save energy. One operator proposes the following metric as a measure of power consumption [2]:

\[
P_{\text{Total}} = 0.35P_{\text{max}} + 0.4P_{50} + 0.25P_{\text{sleep}}
\]

where \(P_{\text{Total}}\) is the power consumption when the equipment is running at full load, \(P_{50}\) when it is submitted to a load equal to 50% of its capacity and \(P_{\text{sleep}}\) when it is idle. The weights (0.35, 0.4 and 0.25) mean for example that we assume that the full load condition occurs during 35% of the time.
Example 1.3: Multi-dimensional Metric and Kiviat Diagram. We measure the performance of a web server submitted to the load of a standard workbench. We compare 5 different configurations, and obtain the results below.

<table>
<thead>
<tr>
<th>Config</th>
<th>Power (W)</th>
<th>Response (ms)</th>
<th>Throughput (tps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>23.5</td>
<td>3.78</td>
<td>42.2</td>
</tr>
<tr>
<td>B</td>
<td>40.8</td>
<td>5.30</td>
<td>29.1</td>
</tr>
<tr>
<td>C</td>
<td>92.7</td>
<td>4.03</td>
<td>22.6</td>
</tr>
<tr>
<td>D</td>
<td>53.1</td>
<td>2.19</td>
<td>73.1</td>
</tr>
<tr>
<td>E</td>
<td>54.7</td>
<td>5.92</td>
<td>24.3</td>
</tr>
</tbody>
</table>

Which configuration is best?

A. A
B. B
C. C
D. D
E. E
F. None of the above
G. I don’t know
Multi-dimensional metrics are commonplace

In such cases use Kiviat diagrams (spider plots)

Key concept is: A is *non dominated* i.e. there is no other configuration that is better in all dimensions

D is also non-dominated

A and D are the only configurations of interest
You need to define the load under which your system operates

Example 2.5: Windows versus Linux. Assume you want to compare Windows versus Linux. Chen and co-authors did it in [7]. They use as metric: number of cycles, instructions, data read/write operations. The load was generated by various benchmarks: “syscall” generates elementary operations (system calls); “memory read” generates references to an array; an application benchmark runs a popular application (ghostview).
Load

You need to define the load under which your system operates

---

**Example 1.3: Power Consumption.** The electrical power consumed by a computer or telecom equipment depends on how efficiently the equipment can take advantage of low activity periods to save energy. One operator proposes the following metric as a measure of power consumption [2]:

\[
P_{\text{Total}} = 0.35P_{\text{max}} + 0.4P_{50} + 0.25P_{\text{sleep}}
\]

where \( P_{\text{Total}} \) is the power consumption when the equipment is running at full load, \( P_{50} \) when it is submitted to a load equal to 50% of its capacity and \( P_{\text{sleep}} \) when it is idle. The weights (0.35, 0.4 and 0.25) mean for example that we assume that the full load condition occurs during 35% of the time.
Load Metric

System under study

Where is the metric?
Where is the load?
Know your goals

**Example 2.5: Windows versus Linux.** Assume you want to compare Windows versus Linux. Chen and co-authors did it in [7]. They use as metric: number of cycles, instructions, data read/write operations. The load was generated by various benchmarks: “syscall” generates elementary operations (system calls); “memory read” generates references to an array; an application benchmark runs a popular application (ghostview).

Goal in Example 2.5 is to make a comparison.

**Example 1.3: Power Consumption.** The electrical power consumed by a computer or telecom equipment depends on how efficiently the equipment can take advantage of low activity periods to save energy. One operator proposes the following metric as a measure of power consumption [2]:

\[
P_{\text{Total}} = 0.35P_{\text{max}} + 0.4P_{50} + 0.25P_{\text{sleep}}
\]

Goal in Example 1.2 is to provide an engineering rule.
Putting Things Together
A Performance Evaluation Study...

System under study

Goals

Load → Metric

System under study
3. Factors
TCP Throughput Increases with Mobility

Example 1.6: TCP Throughput. Figure 1.1, left, plots the throughput achieved by a mobile during a file transfer as a function of its speed. It suggests that throughput increases with mobility.
Does mobility increase throughput?

A. Yes, it is proven by this experiment
B. It is true but perhaps only for a very specific system
C. No it is not true
D. I don’t know

**EXAMPLE 1.6: TCP THROUGHPUT.** Figure 1.1, left, plots the throughput achieved by a mobile during a file transfer as a function of its speed. It suggests that throughput increases with mobility. The right plot shows the same data, but now the mobiles are...
**TCP Throughput Decreases with Mobility**

**EXAMPLE 1.6: TCP Throughput.** Figure 1.1, left, plots the throughput achieved by a mobile during a file transfer as a function of its speed. It suggests that throughput increases with mobility. The right plot shows the same data, but now the mobiles are separated in two groups: one group (‘s’) is using a small socket buffer (4K Bytes), whereas the second (‘L’) uses a larger socket buffer (16 K Bytes). The conclusion is now inverted: throughput decreases with mobility. The hidden factor influences the final result: all experiments with low speed are for small socket buffer sizes. The socket buffer size is a hidden factor.
Why were we fooled?

Hidden factor had a more important role than the factor we were interested in.

Factor: an element that may impact the performance
- (desired factors): intensity of load, number of servers
- (nuisance factors): time of the day, presence of denial of service attack

We interpreted correlation as causality.
Need to be aware of all factors and incorporate in the analysis.
Or randomize experiment to reduce impact of hidden factors.
Simpson’s Paradox

A well known phenomenon -- Special case of Hidden Factor paradox when metric is success rate and factors are discrete

We classify the mobiles as slow (speed $\leq 2$ m/s) or fast (speed $> 2$ m/s). We obtain the following result. We say that a mobile is successful if its throughput is $\geq 1.5$ Mb/s

<table>
<thead>
<tr>
<th></th>
<th>failure</th>
<th>success</th>
<th>$\mathbb{P}$(success)</th>
</tr>
</thead>
<tbody>
<tr>
<td>slow</td>
<td>11</td>
<td>3</td>
<td>0.214</td>
</tr>
<tr>
<td>fast</td>
<td>5</td>
<td>4</td>
<td>0.444</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>7</td>
<td>23</td>
</tr>
</tbody>
</table>

from where we conclude that fast mobiles have a higher success probability than slow

Now introduce the nuisance parameter “socket buffer size”:

<table>
<thead>
<tr>
<th>“S” mobiles</th>
<th>failure</th>
<th>success</th>
<th>$\mathbb{P}$(success)</th>
</tr>
</thead>
<tbody>
<tr>
<td>slow</td>
<td>10</td>
<td>1</td>
<td>0.091</td>
</tr>
<tr>
<td>fast</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>“L” mobiles</th>
<th>failure</th>
<th>success</th>
<th>$\mathbb{P}$(success)</th>
</tr>
</thead>
<tbody>
<tr>
<td>slow</td>
<td>1</td>
<td>2</td>
<td>0.667</td>
</tr>
<tr>
<td>fast</td>
<td>4</td>
<td>4</td>
<td>0.500</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>11</td>
</tr>
</tbody>
</table>
However when examining the individual departments, it was found that no department was significantly biased against women; in fact, most departments had a small bias against men.

<table>
<thead>
<tr>
<th>Major</th>
<th>Men</th>
<th>Women</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Applicants</td>
<td>% admitted</td>
</tr>
<tr>
<td>A</td>
<td>825</td>
<td>62%</td>
</tr>
<tr>
<td>B</td>
<td>560</td>
<td>63%</td>
</tr>
<tr>
<td>C</td>
<td>325</td>
<td>37%</td>
</tr>
<tr>
<td>D</td>
<td>417</td>
<td>33%</td>
</tr>
</tbody>
</table>
Simpson’s paradox explained

\[ P(\text{good} \mid \text{slow}) = \sum_i P(\text{good} \mid \text{slow}, hf = i)P(hf = i \mid \text{slow}) \leq P(\text{good} \mid \text{fast}) = \sum_i P(\text{good} \mid \text{fast}, hf = i)P(hf = i \mid \text{fast}) \geq \]

weights are different

\[ \text{good} = \text{high thruput} \]
\[ hf = \text{hidden factor} \]
\[ hf \in \{\text{Large socket buffer, Small socket buffer}\} \]

\[ P(hf = \text{Small socket buffer} \mid \text{slow}) = \frac{11}{14} \approx 79\% \]
\[ P(hf = \text{Small socket buffer} \mid \text{fast}) = \frac{1}{9} \approx 11\% \]
Avoiding Simpson’s Paradox

\[
P( \text{good} \,|\, \text{slow} ) = \sum_i P( \text{good} \,|\, \text{slow}, hf = i) P(hf = i \,|\, \text{slow})
\]

\[
P( \text{good} \,|\, \text{fast} ) = \sum_i P( \text{good} \,|\, \text{fast}, hf = i) P(hf = i \,|\, \text{fast})
\]

Make the weights equal!

\[P(hf = i \,|\, \text{slow}) = P(hf = i \,|\, \text{fast}), \quad \forall i\]
\[ P(hf = i|\text{slow}) = P(hf = i|\text{fast}), \quad \forall i \text{ means ...} \]

A. The hidden factor \( hf \) and the desired factor slow/fast are independent
B. The hidden factor \( hf \) is distributed uniformly across all experimental conditions
C. A and B
D. None of the above
E. I don’t know
Solution

\[ X \text{ and } Y \text{ are independent } \iff \]
\[ P(X = i \text{ and } Y = j) = P(X = i)P(Y = j) \forall i, j \iff \]
\[ P(X = i|Y = j) \text{ does not depend on } j \]

Answer A is correct

Answer B means \( P(hf = i_1|\text{slow}) = P(hf = i_2|\text{slow}) \forall i_1, i_2 \) and is not necessary
Avoiding Simpson’s Paradox

\[ P(\text{good} \mid \text{slow}) = \sum_i P(\text{good} \mid \text{slow}, hf = i)P(hf = i \mid \text{slow}) \]

\[ P(\text{good} \mid \text{fast}) = \sum_i P(\text{good} \mid \text{fast}, hf = i)P(hf = i \mid \text{fast}) \]

Make the weights equal!

\[ P(hf = i \mid \text{slow}) = P(hf = i \mid \text{fast}), \quad \forall i \]

This means that the hidden factor \( hf \) and the desired factor (slow/fast) are independent

⇒ randomize the experiment so that the hidden factor is independent of the desired factor
Here: when choosing a slow mobile, pick at random between the S (small socket buffer) and L (large socket buffer) mobiles so that the probability of choosing S for a slow is the same as for a fast (instead of 79% and 11%)
Take-Home Message

Identify hidden factors – make them disappear if you can

**Question 1.4.1.** Consider again comparing Windows versus Linux. Can you imagine what factors might play an important role in the analysis? What external factors have to be taken care of during the evaluation? ²

²From [1]: External factors are: background activity; multiple users; network activity. These were reduced to a minimum by shutting the network down and allowing one single user. The different ways of handling idle periods in Windows NT and NetBSD also need to be accounted for, because they affect the interpretation of measurements. Cycle counts in idle periods of NetBSD have to be removed.

else make them appear explicitly in the analysis,

or randomize the experiments to neutralize their impact (i.e. make the nuisance factors independent of the desired factors)
Take Home Message

Performance evaluation uses the language of probabilities
In this course we will exercise how to use probability theory in practice to do scientific work
These are the same fundamentals as data science
Putting Things Together
A Performance Evaluation Study...
4. The scientific method

Joe measures performance of his Wireless Shop:
What do you recommend to Joe?

A. Buy more access points
B. Change your server
C. Call IBM
D. Live with it
E. I don’t know
Joe’s solution: buy 2 more access points
Example 2: Is ATM-ABR better than ATM-UBR?

ABSTRACT. We compare the performance of ABR and UBR for providing high-speed network interconnection services for TCP traffic. We test the hypothesis that UBR with adequate buffering in the ATM switches results in better overall goodput for TCP traffic than explicit rate ABR for LAN interconnection. This is shown to be true in a wide selection of scenarios. Four phenomena that may lead to bad ABR performance are identified and we test whether each of these has a significant impact on TCP goodput. This reveals that the extra delay incurred in the ABR end-systems and the overhead of RM cells account for the difference in performance. We test
Scientific Method

A conclusion can only be proven to be wrong.
Do not draw conclusions unless the experiment was designed to test the statement.

- Measurement 1 suggested that the wireless network was congested, but the experiment was not designed to test this statement.

Joe jumped into a conclusion instead of trying to validate:
H1: “the wireless network is the bottleneck”
- for example: measure the number of collisions / packet loss
- result: collision $\approx 1\%$; conclusion: H1 is not valid
- hypothesis H2: the server is saturated
- experiment: measure memory utilization: result $\approx 100\%$
Performance After Doubling Server Memory

![Graphs showing performance after doubling server memory.](image)
Take Home Message

You should not conclude from an experiment without trying to invalidate the conclusion.

(Popper, 1934): you should alternate between the roles of:
- Proponent
- Adversary
5. Patterns

These are common traits found in different situations. Knowing some of them may save a lot of time.
**Example 1.11: Bottlenecks.** You are asked to evaluate the performance of an information system. An application server can be compiled with two options, A and B. An experiment was done: ten test users (remote or local) measured the time to complete a complex transaction on four days. On day 1, option A is used; on day 2, option B is used. The results are in the table below.

<table>
<thead>
<tr>
<th></th>
<th>remote</th>
<th>local</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>123</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>189</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>99</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>167</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>177</td>
<td>44</td>
</tr>
<tr>
<td>B</td>
<td>107</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>179</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>199</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>103</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>178</td>
<td>71</td>
</tr>
</tbody>
</table>

The expert concluded that the performance for remote users is independent of the choice of an information system. We can criticize this finding and instead do a bottleneck analysis. For remote users, the bottleneck is the network access; the compiler option has little impact. When the bottleneck is removed, i.e., for local users, option A is slightly better.
Bottlenecks may be your enemy

Bottlenecks are like non invited people at a party – they may impose their agenda

Previous example: what we are measuring is the bottleneck, not the intended factor
How do you proceed?

Example 1.12: CPU model. A detailed screening of a transaction system shows that one transaction costs in average: 1’238’400 CPU instructions; 102.3 disk accesses and 4 packets sent on the network. The processor can handle $10^9$ instructions per second; the disk can support $10^4$ accesses per second; the network can support $10^4$ packets per second. We would like to know how many transactions per second the system can support.

A. Do a queuing theory analysis
B. Do a simulation
C. None of the above
D. I don’t know
Bottlenecks are Your Friends

Example 1.12: CPU model. A detailed screening of a transaction system shows that one transaction costs in average: 1’238’400 CPU instructions; 102.3 disk accesses and 4 packets sent on the network. The processor can handle $10^9$ instructions per second; the disk can support $10^4$ accesses per second; the network can support $10^4$ packets per second. We would like to know how many transactions per second the system can support.

The resource utilization per transaction is: CPU: $0.12\%$ – disk: $1.02\%$ – network: $0.04\%$; therefore the disk is the bottleneck. The capacity of the system is determined by how many transactions per second the disk can support, a gross estimate is thus $\frac{100}{1.02} \approx 99$ transactions per second.

If we would like more accuracy, we would need to model queuing at the disk, to see at which number of transactions per seconds delays start becoming large. A global queuing model of CPU, disk access and network is probably not necessary.

In Section 8.2.4 we study bottleneck analysis for queuing systems in a systematic way.
Bottlenecks are Your Friends

- Simplify your life, analyze bottlenecks!
- In many cases, you may ignore the rest
Behind a Bottleneck May Hide Another Bottleneck

etc. The author describes 14 possible components, any of which, if present, is candidate for being the bottleneck, and suggests to remove all of them. Doing so leaves as bottlenecks network access and server CPU speed.
Another pattern...

One UDP source at every node, 2-hop flow, circular symmetry

For large offered load $\lambda$, what happens?
Congestion Collapse

Definition: Offered load increases, work done decreases

Frequent in complex systems

May be due to

- cost per job increases with load
- Impatience (jobs leave before completion)
- Rejection of jobs before completion

Designer must do something to avoid congestion collapse

- Eg. Admission control in web servers
- Eg. TCP congestion control

Analyst must look for congestion collapse
Sources use TCP (= fair scheduling). Increase capacity of link 5 to 100 kb/s; what happens to source 1?

A. Its rate increases
B. Its rate decreases
C. Nothing happens
D. I don’t know
Before: \( x_1 \approx 100 \text{ kb/s}, \ x_2 \approx 10 \text{ kb/s} \) (Pareto efficiency of TCP)

After: \( x_1 \approx 55 \text{ kb/s}, \ x_2 \approx 55 \text{ kb/s} \) (assuming same RTT; TCP loss throughput formula)
Competition Side Effect

System balances resources according to some scheduling
Putting more resources changes the outcome of the scheduling
Apparent paradox: put more resources, some get less
No TCP, users send as much can
Increase capacity of link 2 from 10 to 1000 kb/s
Latent Congestion Collapse

System is susceptible to congestion collapse
Low speed access prevents congestion collapse
Adding resources reveals congestion collapse

Apparent paradox: put more resources, all get less
Take Home Message

Watch for patterns, they are very frequent

- Bottlenecks
- Congestion collapse
- Competition side effects
- Latent Congestion collapse
Now it’s your turn...

Performance Evaluation

Homework 1

1 Assignment

Customers in Joe’s shop are not satisfied because the downloading time is very large. Joe has hired you as performance analyst to understand the problem and propose some solutions.