On the Multiple Dimensions of Performance and Implications to System Architectures

Dr. Yitzhak (Tsahi) Birk
Electrical Engr. Dept.
Technion
Background

- Everybody wants “high performance”
- There are many measures for performance
- Technological advances affect different system components and different measures differently
- Better may even appear worse.
Outline

• Performance measures

• Examples

• Conclusions
Performance Measures

• **User perspective:** time to accomplish mission (delay; response time; latency)
  - queuing delay
  - service time

• **Service-provider perspective:** throughput (Missions/sec; bits/sec; transactions/sec)

Without queuing: delay = 1/throughput?
Pipelining

Tasks:
- Wash: 55 min
- Dry: 45 min
- (Transfer: 5 min)

<table>
<thead>
<tr>
<th></th>
<th>Maximum throughput</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combo</td>
<td>1/110</td>
<td>110</td>
</tr>
<tr>
<td>Separate</td>
<td>1/65</td>
<td>115</td>
</tr>
</tbody>
</table>

Increases throughput; may hurt latency.
A Little Queuing Theory: Little’s Theorem

- Queuing models assume state of equilibrium: input rate = output rate
- Little’s Law:
  - Mean number of customers in system = arrival rate $\times$ mean time in system.
  - Mean number of customers in queue = arrival rate $\times$ mean time spent in queue.
- This is one of the very few general theorems in queuing theory.
Example: Throughput-Delay-Throughput Tradeoff in Web Browsing

For a given server:

High server throughput (efficient use of server)  
⇒ high server load  
⇒ long response time to user  
⇒ user waits  
⇒ low user throughput!
Example: Disk-Size Selection

Choosing a disk drive for an on-line transaction processing (OLTP): small accesses.

- Any single disk drive: 100 TPS
- Application requirement: 0.002 TPS/MB

Disk selection:
- Maximum disk capacity \( C \): 
  \[ C \cdot 0.002 \leq 100 \Rightarrow C \leq 50 \text{GB} \]
- At 50GB, cost/GB still dropping

\( \Rightarrow \) use 50GB drives
“Faster”

• Doing the same thing in less time?

• Doing more in the same time?

• Are these two identical?

• If not, is one easier than the other?
“Faster”: Same Mission in Less Time

• 1K - processor computer takes 10s to simulate 1s of a 1k-point wind tunnel.
• Is a 10K-processor computer 10 times faster?
  – can’t simulate this tunnel any faster, but
  – can simulate 1s of a 10K-point tunnel in 10s.
• A 1K-processor machine with a 10X clock rate can simulate the original tunnel in real time.

More in equal time ≠ same in less time

Is “more in equal time” always easier?
“Faster”: Doing More in Equal Time

- Problem: given $N$ numbers, visit them in random order and sum up.
- 100MHz processor takes 100s with $N=10^9$.
- 1,000MHz processor takes 10s with $N=10^9$, but
- 1,000MHz processor takes 100 hours for $N=10^{10}$:
  - processor (and memory) indeed faster, but
  - $10^{10}$ doesn’t fit in memory $\implies$ disk is the bottleneck.
- So, “more in equal time” can be harder.

More in equal time $\implies$ same in less time
Instructions/meter

• **Definition:** the number of instructions that a computer can execute while a signal travels one meter.

• **Independent of transmission rate!!!**

• **Example:**
  - Computing rate: $2 \cdot 10^9$ Instructions/sec
  - Speed of light in optical fiber: $2 \cdot 10^8$ m/sec
  \[ \Rightarrow \ 10 \text{ Instructions/meter} \]

• **Implication:** a computer in Haifa that must synchronize with one in Tel-Aviv will wait at least the execution time of 2 million instructions.

• **Trend:** for interactive computing, the world is “expanding,” and the importance of geographical proximity is increasing with time.
Multiple Slow Vs. Single Fast Server

- Options:
  - one 100ppm printer
  - four 25ppm printers

- Printing capacity: equal

- Single-printer delay:
  - shorter service time
  - “full steam” unless no jobs

- Multiple-printer delay:
  - can’t get stuck behind a single long job

- Optimum: $f$ (coefficient of variation of job length).

Life is complicated...
Required Amount of Work per Mission

- Given: 4 disk drives
  - Time to access a block: $t_s$
  - Time to transfer a block: $t_t$
- Requests to read all same-colored blocks of a randomly-chosen color
- Goal: maximize maximum throughput
- Data layout options:
  - Same-colored blocks on different disks
  - Same-colored blocks on same disk
- For both options: assume perfect load balancing and that all disks are busy all the time.
- $S_{max} = 4/W$, where $W$ is the amount of disk work per mission

For given resources, maximum throughput is determined by the amount of work per mission!
Maximizing Deadline-Constrained Capacity
(Multi-channel ALOHA satellite network for on-line transactions)

• Scenario:
  – Contention channel
  – No sensing
  – User: deadline and permissible Pr(failure)
  – Service provider’s goal: max. capacity
  – Example:
    » Delay permits 2 rounds.
    » P(collision) = 0.1
    » P(failure) = 0.0001

• Send 4 copies in 1st attempt:
  – minimum mean delay
  – 4 copies per message

• Send (1,(3))
  – longer mean delay
  – 1.3 copies per message
  – 3-fold higher capacity!

Conclusion: Address the exact problem!
The Arrow ABM System [Dov Raviv]

- Arrow miss probability: 0.1
- System requirement: 0.001
- Possible solution: 3 arrows
- Key observations:
  - Advance radar warning allows a 2nd attempt upon failure.
  - Interception time is not important.
- Optimal solution:
  - Fire one arrow
  - Iff it misses, fire two more
  - Probability of failure: 0.001
  - Average of 1.2 Arrow per target

Exact requirement + smart solution ⇒ 60% savings!
Learn from other applications!
Configuring a Distributed Computing Environment

- **Scenario**
  - collaborating computers, including small exchanges.
  - users scattered across 100s of meters or even several km.
- **Components:**
  - users (cannot be moved)
  - user I/O (must be with user)
  - “engines” (computation, storage, memory): flexible.

Where should the “engines” be located?
Distributed Environment Options

• **Network of workstations:**
  – User, user-I/O and engine collocated.
  – Long lines between engines

• **Engine cluster:**
  – User with user-I/O.
  – Engines of different users are collocated.
Key observations:
- Users measure delay in seconds. 
  \(<50\text{ms} = \text{“instantaneous”}\)
- Processors measure delay in cycles.
- High data rates are becoming cheap.

Conclusion:
- Trends favor engine cluster
- Engine <-> user I/O: exploitation of high data rate (4Gbit/s) with no delay complications.

“Engine cluster” with Fiber to user I/O
Engine Cluster

- Tailored to the true performance sensitivities of the system components

- New opportunities:
  - physical security and safety
  - pooling of delay-sensitive resources (memory!)

- Can progress step-by-step towards a mainframe, but can stop at any point!

Is “user I/O” the same as “network computer”?
Summary

• “Performance” is multi-faceted.
• True meaning of a given “measure” (e.g., delay) may depend on the measuring entity.
• “Innocent” technology trends may lead to a “discontinuity” in system architecture.
• It is important to understand the interplay among system elements and those of performance and to match the resources to the true needs of applications.

Inventing a novel solution is great, but fully understanding the problem is equally important!
Inventing novel solutions is great, but fully understanding the problems is equally important!