Performance Analysis

for Oracle DBAs Who Can't Wait

Dr. Neil J. Gunther

Performance Dynamics

NorCal ORACLE Users Group (NoCOUG)
Winter Conference, Feb 11, 2010
Technical Session

Outline


KONI
27
KUMHO
crefo/factoring
ERS
ATS

When Should You Think About Performance?

Function first!

- No point tuning system or app that is pathologically broken
- Many fixes are cheap, e.g., memory paging

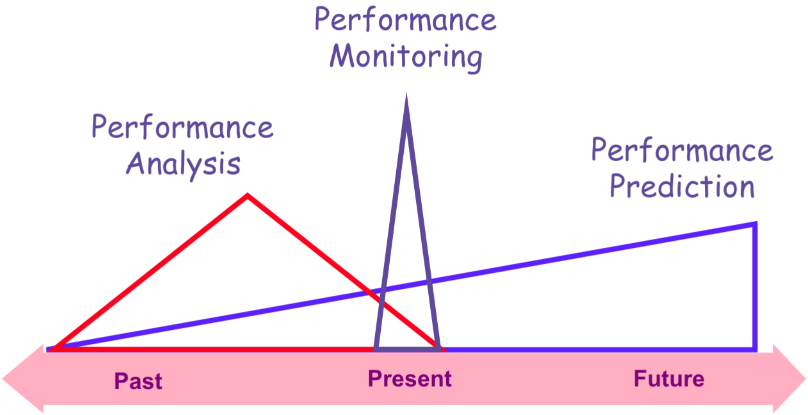
Scalability

"Scalability is hard because it cannot be an after-thought. Good scalability is possible, but only if we architect and engineer our systems to take scalability into account." —Werner Vogels, Amazon.com CTO

Velocity Conference 2009

"Measure everything and throw science at it!" —John Adams, Twitter.com

Performance Management Spectrum



Performance Metrics

Primary metrics:

- **Time:** e.g., milliseconds (ms), minutes (min), hours (h)
- **Rate:** tx/min, pkt/s, MIPS (inverse time)
- **Capacity:** KB, MB, TB (disks, buffers, caches)

The following take us into capacity planning

Secondary metrics:

- Dollars ($): $/tx
- Watts (W): tx/W
- Various associated cost metrics

Standard Performance Metrics

Symbol	Metric	Definition
T	Measurement period	Same as the sample period
W	Waiting time	Time spent in a buffer
S	Service time	Time spent getting processed
B	Busy time	Total time the resource is busy
C	Completion count	Number of completed requests
R	Residence time	Time spent waiting and being serviced
$\mathcal{R}$	Response time	Sum of all the residence times
X	Throughput (C/T)	Rate at which work is completed
ρ	Utilization (B/T)	Fraction of T the resource is busy
Q	Queue length	Total number of requests in the system

NOTE: No symbol for *idleness*

Outline

What Makes Queueing Theory Difficult?

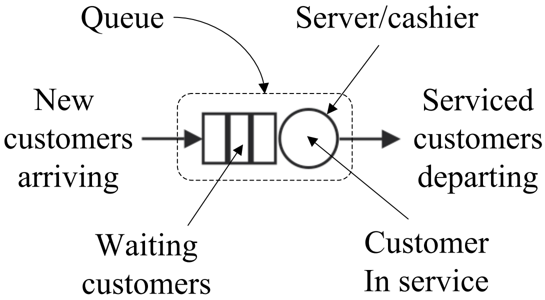
Performance grocery shopping SLO: get out of there ASAP.

Arriving . . . . . . Waiting . . . . . . Servicing . . . . . . Departing

Fluctuations

Fluctuations (instantaneous changes) make queuing unpredictable.
So, let's turn them (higher moments) off and just look at averages.

Characterizing a Queue

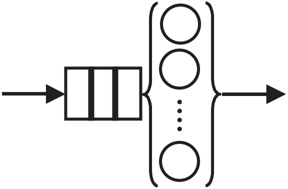


Ambiguity

Is the queue just the people waiting at the checkout or everybody, including the person being served?

Pop Quiz: What is the Queue Length?

Consider a single line with multiple servers:



Example

- Post office
- Bank branch

If nobody is waiting, how long is the queue?

Residence Time

. . . . . . Waiting time (W) + Service time (S) . . .

W + S

Theorem (Total time spent in the queueing system)

An arriving customer spends time (W) due to all those in the queue as well as their own service time (S) with the cashier: $R = W + S$.

Remark

Shopping time can be treated as a separate residence time (R_S).
The shopper is a *mobile server*—like the waiter in a restaurant.

Some Basic Metric Relations

Fundamental

$$X = \frac{C}{T}$$

$$\rho = \frac{B}{T}$$

$$S = \frac{B}{C}$$

$$R = W + S$$

Derived

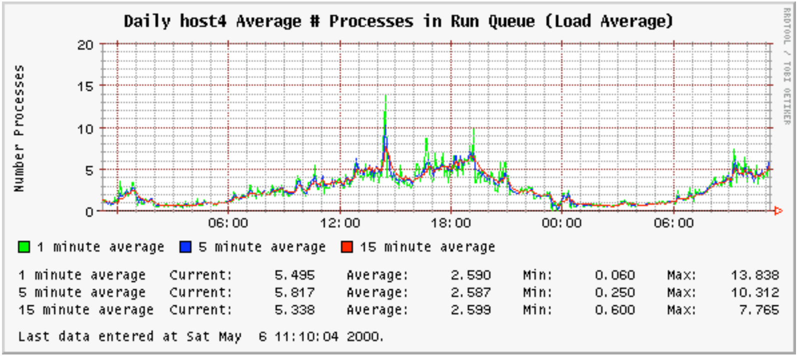
$$Q = XR \text{ (Little's law 1)}$$

$$\rho = XS \text{ (Little's law 2)}$$

$$R = \frac{S}{1 - \rho}$$

$$R = \frac{N}{X} - Z$$

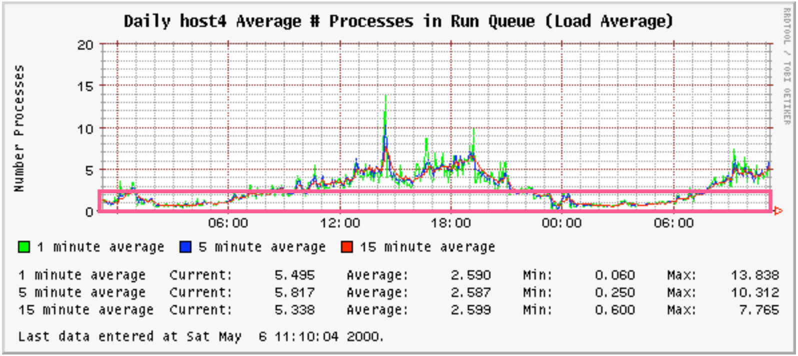
Linux Load Average



Remark

Staring at a time series like this is about as useful as trying to predict the weather by watching a weather-vane twist in the wind.

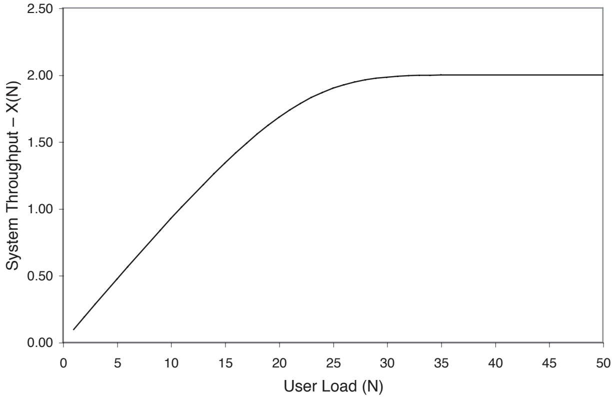
Average Queue Length




```
> colh<-c(1.25, 0.8, 0.8, 1.0, 1.1, 1.5, 2.25, 2.5, 3.0, 3.5, 3.25, 3.75, 4.75, 0.5*(14-5),
+ 4.0, 5.0, 5.1, 5.5, 5.0, 4.0, 2.25, 2.0, 1.0, 1.0, 0.9, 0.9, 1.0, 1.0, 1.1, 2.0, 2.25,
+ 3.5, 4.25, 4.75)
> sum(colh)/length(colh)
[1] 2.660294
```

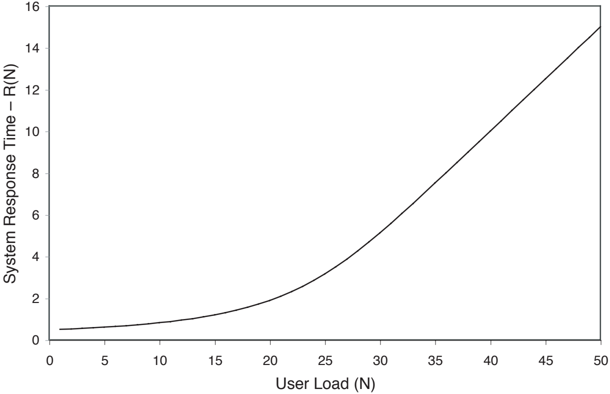
Rectangle height $Q = 2.66$ processes and $Q = XR$ from Little's law
If know R from ORACLE, then can estimate X for applications

Queueing Throughput



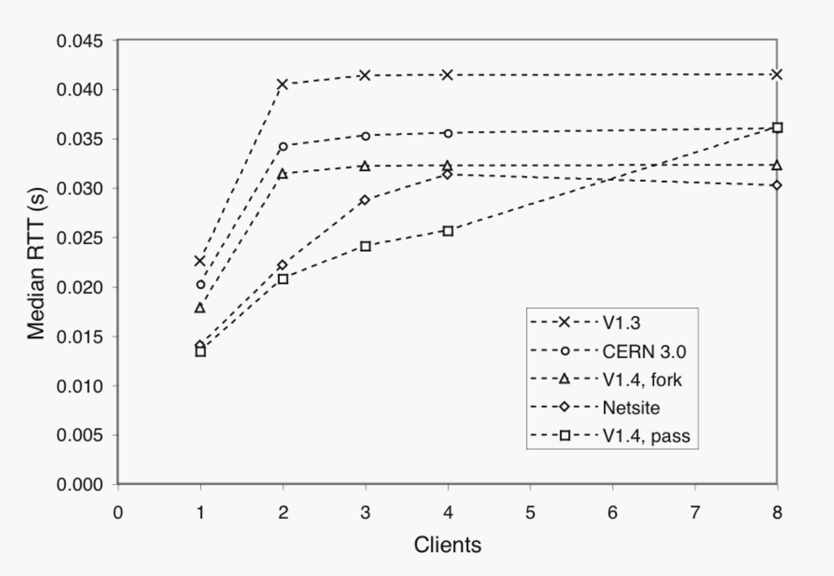
$X = X(N)$ is always **concave.** Any deviation must be explained.

Queueing Response Time

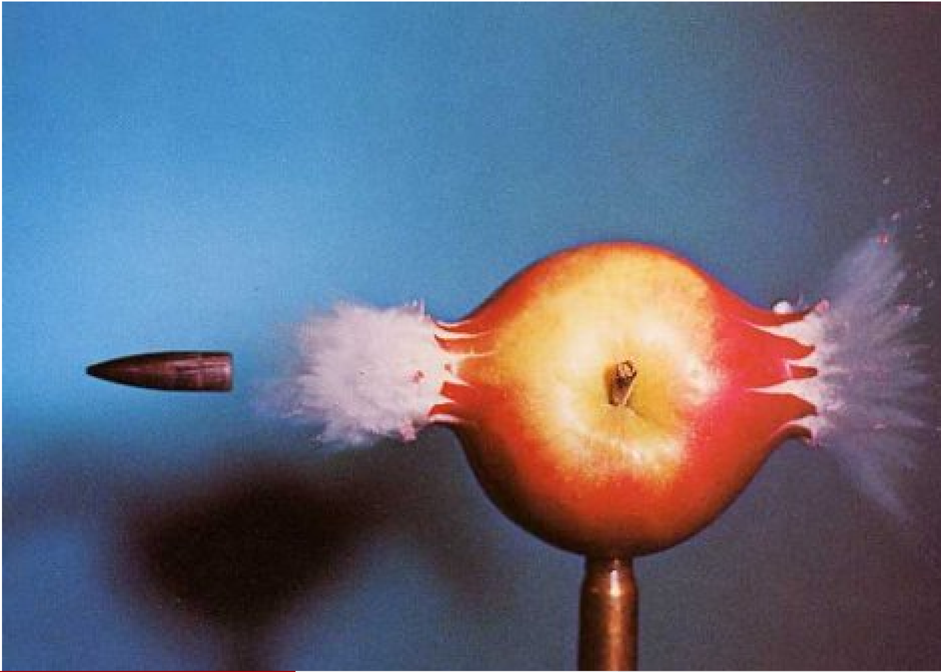


$R = R(N)$ is always **convex**. Any deviation must be explained.

Pop Quiz: Explain This?



Outline



Sequent Silver Bullet Benchmark

All competitive benchmarking is war

Sequent c.1985 first to break the 100 TPS barrier (TP1)

[Oracle/nCUBE broke 1000 tpsB c.1990]

But they cheated (OMG!)

- All transactions in memory
- DB tables not scaled
- No physical I/O
- Not reported
- No benchmark ”cop”

Led to formation of TPC.org (Transaction Processing Council)

SPEC Matrix300 Supercharging

c.1991 HP improved their SPECmark rating by 600%
Now, separate SPEC CINT2006 and SPEC FP2006 CPU ratings

But HP cheated (OMG!)

- Only improved `matrix300` code (out of 10 SPEC FP codes)
- Used special numerical compiler from David Kuck & Assoc.
- Footnote in SPEC Newsletter
- Could purchase separately from HP (sorta made it legal)

Led to current SPEC_base and SPEC_peak ratings

TPC-D Supercharging

TPC-D (DSS benchmark) took years to develop
Took more years for vendors to report their numbers

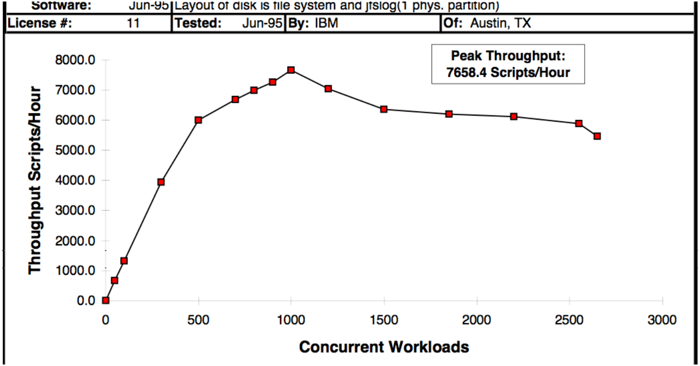
Then vendors cheated (OMG!)

- All 20 complex queries were specified
- Run query planner on them
- Optimize query plan (like -O3 compiler optimization)

Led to current TPC-H and TPC-R split

Benchmarks are Controlled Experiments

Should come with full disclosure of workload and runtime settings



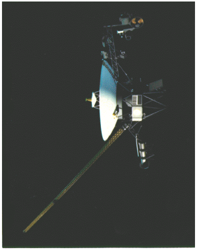
Should report more than single measurement, e.g., SPEC SDM
NOTE: $X(N)$ degrades as N client processes increases!

Outline

Talking to Voyager I

Voyager I is now 113 AUs from Earth. (launched 1977)
RTT is slightly longer than a day. (31 hrs)

⟶ CMD ⟶
⟵ ACK ⟵

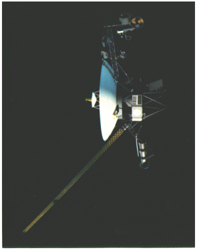


Talking to Voyager I

Voyager I is now 113 AUs from Earth. (launched 1977)
RTT is slightly longer than a day. (31 hrs)

⟶ CMD ⟶
⟵ ACK ⟵

⟶ CMD ⟶
⟵ ACK ⟵



Transmit simultaneously on another channel. (if Voyager had one)

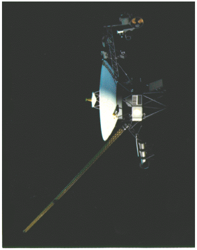
Talking to Voyager I

Voyager I is now 113 AUs from Earth. (launched 1977)
RTT is slightly longer than a day. (31 hrs)

⟶ CMD ⟶
⟵ ACK ⟵

⟶ CMD ⟶
⟵ ACK ⟵

⟶ CMD ⟶
⟵ ACK ⟵



Transmit simultaneously on another channel. (if Voyager had one)
More channels increase **bandwidth**. (More bits per second)

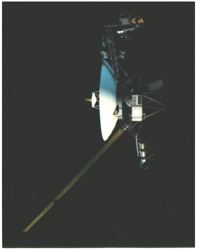
Talking to Voyager I

Voyager I is now 113 AUs from Earth. (launched 1977)
RTT is slightly longer than a day. (31 hrs)

⟶ CMD ⟶
⟵ ACK ⟵

⟶ CMD ⟶
⟵ ACK ⟵

⟶ CMD ⟶
⟵ ACK ⟵



Transmit simultaneously on another channel. (if Voyager had one)
More channels increase **bandwidth**. (More bits per second)
But **latency** remains the same. (Fixed by *c*)

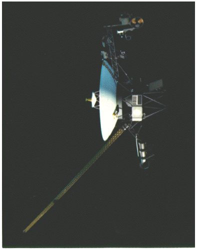
Talking to Voyager I

Voyager I is now 113 AUs from Earth. (launched 1977)
RTT is slightly longer than a day. (31 hrs)

$\longrightarrow$ CMD $\longrightarrow$
$\longleftarrow$ ACK $\longleftarrow$

$\longrightarrow$ CMD $\longrightarrow$
$\longleftarrow$ ACK $\longleftarrow$

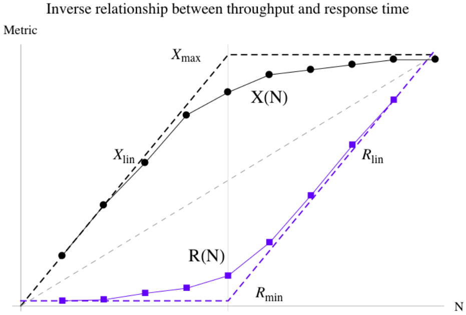
$\longrightarrow$ CMD $\longrightarrow$
$\longleftarrow$ ACK $\longleftarrow$



Transmit simultaneously on another channel. (if Voyager had one)
More channels increase **bandwidth**. (More bits per second)
But **latency** remains the same. (Fixed by c)
$\therefore$ Latency and bandwidth appear to be independent.

Throughput and Latency Are *NOT* Independent

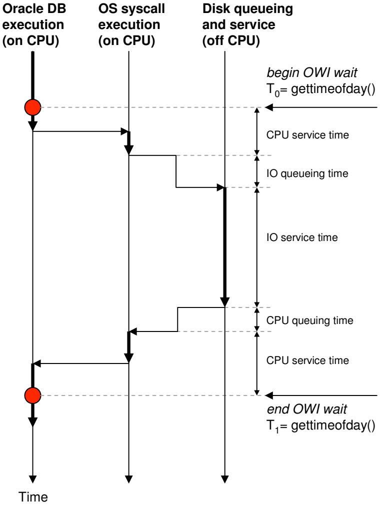
Voyager is an *illusion* due to a special case, i.e., no queueing.
Commonly seen in pkt networks. (deterministic)



Throughput (X) and latency (R) are related *nonlinearly* by queueing.
Especially true for RDBMS systems, like ORACLE.

Outline

OWI Waiting Times Components



[Source: Põder and Gunther, CMG Conf. 2008]

Will the Real OWT Please Stand Up?

OWI “wait time” is actually a *response time*, R in proper queueing theory parlance. [see Millsap & Holt]

The Real OWI

Each OWI wait metric (R_w) is the sum of all actual wait times (W_i), in the sense of waiting for service, measured during i intervals and their associated service times (S_i):

$$R_{OWI} = \sum_i W_i + \sum_i S_i$$

Some exceptions.

Example

`DB_CPU_PCT` measures only S_{CPU} (execution cycles) without any W_i. Determined by instrumentation and data sources are available to OWI.

Thank you for your attention

My brain is now open for questions

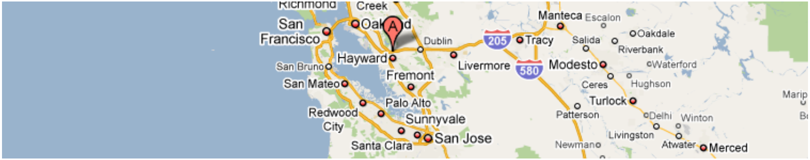
☺

Outline

References

- N. J. Gunther, *The Practical Performance Analyst*, iUniverse Press 2000
- N. J. Gunther, *Analyzing Computer System Performance with Perl::PDQ*, Springer 2005
- N. J. Gunther, *Guerrilla Capacity Planning*, Springer 2007
- C. Millsap and J. Holt, *Optimizing Oracle Performance* O'Reilly 2003
- T. Põder and N. J. Gunther, "Multidimensional Visualization of Oracle Performance Using Barry007," CMG Conference 2008
- N. J. Gunther, "Benchmarking Blunders and Things That Go Bump in the Night," Presented at WORP Workshop 2004, `arxiv.org/abs/cs/0404043`
- N. J. Gunther and P. Stalder, "How to Quantify Oracle Database Scalability: Parts I and II," Hotsos Symposium 2010, `www.hotsos.com/sym10`

Contact Coordinates



- Castro Valley, California, 94552
- www.perfdynamics.com
- perfdynamics.blogspot.com
- twitter.com/DrQz
- njgunther@perfdynamics.com
- +1-510-537-5758