Computational Systems Engineering

Christoph Kirsch
University of Salzburg

3 Unit Graduate Course, Winter 2004/2005
Chapter 4: Benchmarking

Heisenberg

- Uncertainty Relation:

  The more precisely the position is determined, the less precisely the momentum is known in this instant, and vice versa.

- Uncertainty Principle

© C. Kirsch 2004

Throughput

- the rate at which a system processes objects end-to-end
- unit: objects/unit of time
  - example: bits/sec, cars/day, people/year
  - it is not speed, which is distance/unit of time
  - good for absolute performance
  - bad for perceived performance

© C. Kirsch 2004

Latency

- the time a system needs to process an object end-to-end
- unit: unit of time
- example: milliseconds
- again, it is not speed
- jumbo vs. supersonic, truck vs. fiber
- throughput and latency are opposed goals

© C. Kirsch 2004

It's the Latency, Stupid

- Stuart Cheshire, May 1996
- Making more throughput is easy
  - Get 10 phone lines instead of one
  - Making less latency is not
  - It is better to have a small share of a high-throughput, low-latency line than to have all of a low-throughput, high-latency line

© C. Kirsch 2004

Stanford - MIT, 1996

- The distance from Stanford to Boston is 4320km.
- The speed of light in vacuum is 300 x 10^6 m/s.
- The speed of light in fibre is roughly 66% of the speed of light in vacuum.
- The speed of light in fibre is 300 x 10^6 m/s * 0.66 = 200 x 10^6 m/s.
- The one-way delay to Boston is 4320 km / 200 x 10^6 m/s = 21.6ms.
- The round-trip time to Boston and back is 43.2ms.
- The current ping time from Stanford to Boston over today's Internet is about 85ms:

```
[cheshire@nitro]$ ping -c 1 lcs.mit.edu
PING lcs.mit.edu (18.26.0.36): 56 data bytes
64 bytes from 18.26.0.36: icmp_seq=0 ttl=238 time=84.5 ms
```

- So: the hardware of the Internet can currently achieve within a factor of two of the speed of light.

© C. Kirsch 2004

Salzburg - Berkeley, 2004

- Wahine: ck$ traceroute www.eecs.berkeley.edu

```
traceroute to web1.eecs.berkeley.edu (169.229.60.94), 30 hops max, 40 byte packets
 1  10.0.0.254 (10.0.0.254)  24.201 ms  3.239 ms  3.303 ms
 2  81-223-189-113.itzling.xdsl-line.inode.at (81.223.189.113)  157.561 ms  13.088 ms  31.037 ms
 3  voe2-vl-00-010.voesend.vien.inode.at (62.99.170.221)  18.832 ms  21.808 ms  80.836 ms
 4  vie2-vl-00-020.shuttle.vien.inode.at (62.99.170.205)  33.394 ms  23.258 ms  16.192 ms
 5  otta-gb-03-002.shuttle.vien.inode.at (62.99.170.17)  17.28 ms  26.757 ms *
 6  ff-m-po-02-001.frankfm.germ.inode.de (62.99.170.106)  155.312 ms  32.134 ms  32.479 ms
 7  po1-0.core01.fra03.atlas.cogentco.com (80.81.192.63)  75.607 ms  42.227 ms  34.975 ms
 8  p12-0.core01.dca01.atlas.cogentco.com (154.54.1.17)  131.743 ms  125.974 ms  119.981 ms
 9  p15-0.core01.dca01.atlas.cogentco.com (66.28.4.21)  130.403 ms *  127.108 ms
10  p10-0.core02.sfo01.atlas.cogentco.com (66.28.4.209)  303.818 ms  203.276 ms  201.534 ms
11  cenic.demarc.cogentco.com (38.112.6.226)  193.605 ms  208.736 ms  206.211 ms
12  inet-ucb--lax-isp.cenic.net (137.164.24.142)  304.41 ms  264.708 ms  190.957 ms
13  vlan194.inr-202-doecev.berkeley.edu (128.32.0.251)  194.394 ms  196.212 ms  234.06 ms
14  doecev-soda-br-6-4.eecs.berkeley.edu (128.32.255.170)  192.544 ms  214.33 ms  194.122 ms
15  sbd2a.eecs.berkeley.edu (169.229.59.226)  203.337 ms  437.41 ms  427.557 ms
16  web1.eecs.berkeley.edu (169.229.60.94)  354.457 ms  197.736 ms  210.725 ms
```

© C. Kirsch 2004

Thrashing

- a condition in which a computational system spends more time administrating resources than utilizing resources
- little or no progress of application functionality
- typically patterns of requests for resources followed by inadequate access emerge
- examples: virtual memory page faults, network collisions

© C. Kirsch 2004

What to Benchmark?

- throughput: bits/seconds
- latency: milliseconds
- error rate: %
- maximum number of connections
- thrashing
- system time: disk, network, reactor, scheduler
- user time: parser, error handler, generator

© C. Kirsch 2004

How to Benchmark

- Hardware: server plus multiple client machines plus network switch
- Software: httperf, apachebench, autobench
- Problems: bottlenecks (network, clients)
- Solution: vary network/client capacity
- Check out: SPECweb99

© C. Kirsch 2004

Static Benchmarking

- request the same file many times
- simple but not real world (caching)
- tool: httperf
-
```
httperf --server host.mydomain.com \
        --uri /index.html \
        --num-conn 5000 \
        --num-call 10 \
        --rate 200 \
        --timeout 5
```

© C. Kirsch 2004

Automatic Benchmarking

- request the same file many times at increasing rates
- tool: autobench
- generates output in CSV/TSV format

© C. Kirsch 2004

Dynamic Benchmarking

- request different files many times
- use sequences of requests obtained from web logs

- tool: httperf

- ```
  httperf --server host.mydomain.com \
          --wsesslog 1000,2,session.log \
          --max-piped-calls 5 \
          --rate 20 \
          --timeout 5
  ```

© C. Kirsch 2004
  ```

Jobs

- System Administration
- Website
- Benchmarking
- Webserver development
- Library development

© C. Kirsch 2004