COMP 190-088: Systems Performance Analysis

Performance Metrics

Jasleen Kaur

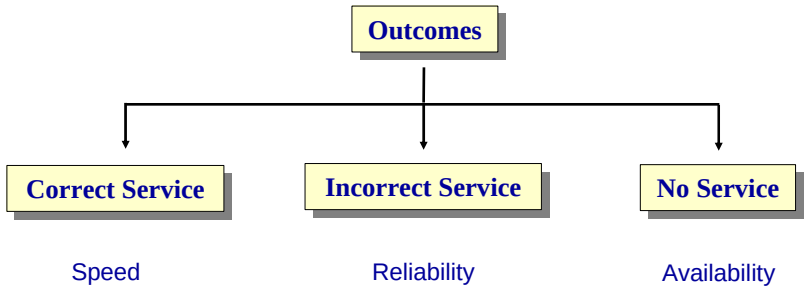
Department of Computer Science
The University of North Carolina at Chapel Hill

Spring 2005

Performance Metrics: Outline

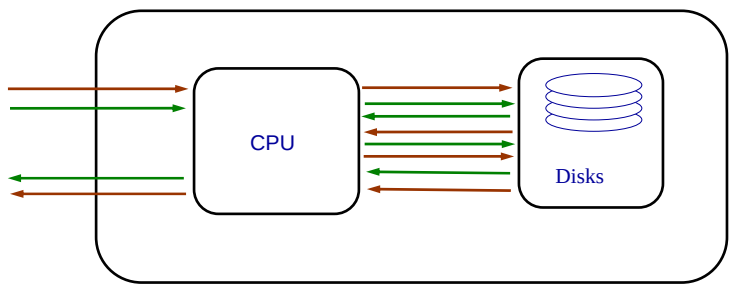
- Common Performance Metrics
- Measuring Time
- Metric Selection
- Ratio Games

Outcomes For Any Service



Time-related performance matters only in case of correct service

When Does Performance Degrade?



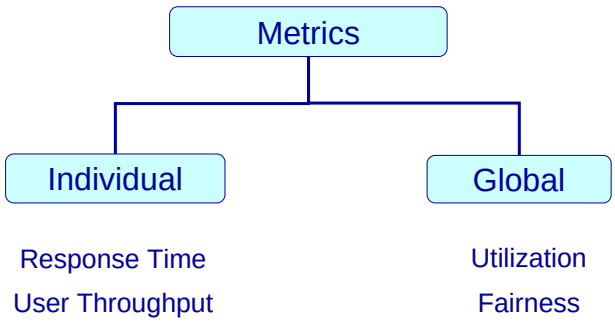
- Bottlenecks
  - Component with longest processing time
  - Key determinant of end-to-end response time
- Congestion in multi-user systems
  - Increases response time at the bottleneck
  - Determine system load that maximizes revenue

Need metrics that capture all performance-related issues

Notions of Performance

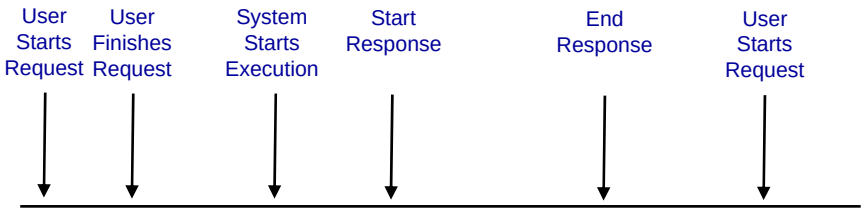
- At least 4 ways of talking about performance:
  - Responsiveness
    - Time taken to service a request
  - Productivity
    - Rate at which requests are served
  - Utilization
    - Extent to which a system is left non-idle with respect to resource usage
  - Fairness
    - Extent to which different clients receive "fair" service

Metrics in Multi-User Systems

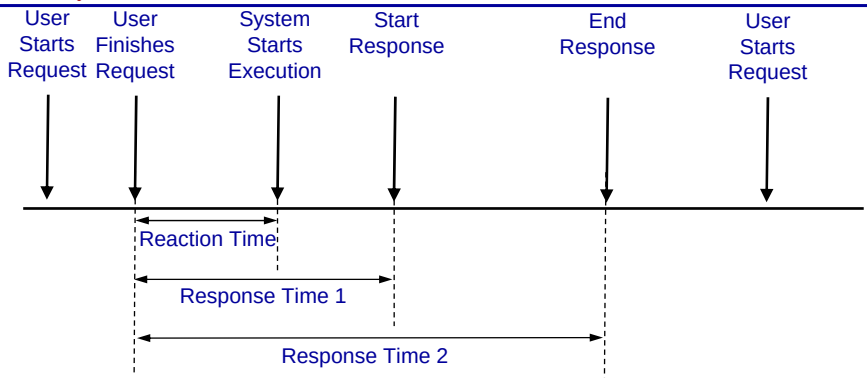


Commonly Used Performance Metrics

- Time
  - The zeroth-order performance metric
  - The basis of all computer performance measurement and analysis



Response Time



- Response Time
- Turnaround Time
  - For batch systems
- Reaction time
- Stretch Factor
  - (Response time at a particular load)/(Response time at minimum load)

Productivity-related Metrics

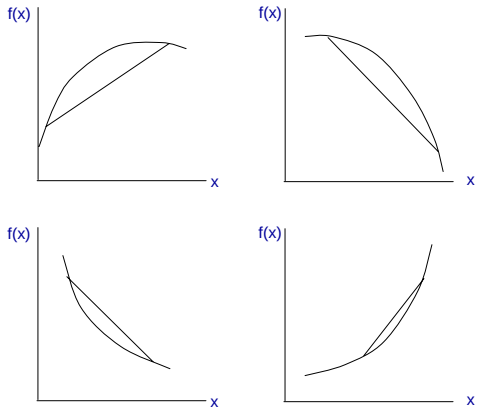
- Throughput
  - Rate at which requests are served
- Nominal capacity or bandwidth
  - Max achievable throughput under ideal workload conditions; MIPS
- Usable capacity
  - Max throughput without exceeding a response-time limit
- Knee capacity
  - Throughput at knee of response-time/throughput curve
- Efficiency
  - (Usable capacity)/(Nominal capacity)
- Utilization
  - Fraction of time a resource is busy servicing requests
- Idle time
- Cost/Performance ratio
  - Cost: computed over a given number of years
  - Performance: throughput under a given response time constraint

Reliability and Availability

- Reliability = R(t)
  - Probability of errors
  - Probability that system has remained up until time t, given that it was already up at t=0
- Error-free seconds
  - Mean time between errors
- Mean time between failure (MTBF) $= \int_0^\infty R(t)dt$
  - Avg uptime
- Failure rate $= \lim_{t \to 0} \frac{R(t) - R(t+dt)}{R(t)dt} = \ldots = -\frac{d(\ln R(t))}{dt}$
- Mean time to repair (MTTR)
  - Avg downtime
- Availability
  - Fraction of time the system is available to serve user requests
    - (avg uptime) / (avg uptime + avg downtime)
  - Joint availability in a distributed system can never be greater than that of the weakest link in a chain of components

Metric Types

- Lower is better (LB) vs HB
  - Response time, MTTR, Cost/Performance; Nominal capacity, MTBF, Availability
- Concave vs Convex Utility
  - Concave functions => Diminishing returns



Concave Functions

Convex Functions

Performance Metrics: Outline

- Common Performance Metrics
- Measuring Time
  - Issues
- Metric Selection
- Ratio Games

Measuring Time

- Discrete vs Continuous Time
  - Discrete systems
    - Events can occur at only specific time instances
  - Most computer systems are discrete systems
    - Finest granularity: clock ticks
  - Most performance measurements deal with discrete time instances
  - Discrete time approaches continuous time as the tick intervals become infinitesimally small

Measuring Time: Time-scales

Computer Device	Device access time	Actual seconds	Scaled unit
CPU clock	1 ns	1 ns	1 s
CPU register	1 CPU cycle	10 ns	10 s
L1 cache	2 CPU cycles	20 ns	20 s
L2 cache	10 CPU cycles	100 ns	1.67 min
Main memory	100 CPU cycles	1 μs	16.67 min
Disk	10 ms	10 ms	3.86 months
NFS op	50 ms	50 ms	1.59 years
TPC-A update	1 s	1 s	31.71 years
Tape archive	10 s	10 s	3.17 centuries

- Range of time-scales is huge!
  - More than 10 orders of magnitude
- Do not have to measure at all time-scales
  - Track only those changes that occur at time-scales similar to the quantity you're trying to predict/measure

How big is a nano-second?

- Time it takes for current to travel the length of your forearm
  - ~ 1 ft
- System buses that operate at 1 Gb/s are restricted to about 1 ft in length. Why?
  - CPU carrier pinouts limit data paths to about 16 bytes in width
  - => To support 1 Gb/s bus bandwidth, bus clock frequency = 64 MHz
  - Bus may not be very efficient => use 100 MHz
  - => 10 ns bus cycle time
  - => all devices that interface to the bus must settle in 10 ns
  - ~ 6 ns required to drive voltage levels and allow clock skew
  - => 4 ns available to set appropriate voltage levels on bus
  - On a loaded bus, it takes 2 ns/ft to propagate a signal
  - => bus length = 4ns/(2 ns/ft) = 2 ft
  - It takes two phases to set voltage levels
  - => max advisable bus length ~ 1 ft

Measuring Time in Distributed Systems: Clocks

- Clock properties:
  - Offset
    - Time difference between two clocks
  - Skew
    - Change in clock offset with respect to time = d(offset)/dt
  - Drift
    - Time variation in skew = d(skew)/dt
  - Stability
    - Measure of how well a clock maintains a constant frequency
  - Accuracy
    - How well its frequency and time compare to defined standards
  - Precision
    - How accurately stability/accuracy can be maintained within a particular timekeeping system
- Clock synchronization
  - To match clocks in both frequency and epoch
    - Clock stability (maintaining frequency) more important
  - Common reference frame: atomic clock