Selection of Techniques and Metrics

Le, Hong Trang (lhtrang@hcmut.edu.vn)
Tran, Van Hoai (hoai@hcmut.edu.vn)

Faculty of Computer Science & Engineering
HCMC University of Technology

2019-2020/Semester 1

Outline

1. Technique selection
2. Metric selection
   - Case study
3. Common performace metrics
4. Utility classification of metrics

Outline

1 Technique selection

2 Metric selection
- Case study

3 Common performace metrics

4 Utility classification of metrics

Criteria in selection

Criterion	Analytical modeling	Simulation	Measurement
Stage	Any	Any	Postprototype
Time required	Small	Medium	Varies
Tools	Analysts	Computer languages	Instrumentation
Accuracy	Low	Moderate	Varies
Trade-off evaluation	Easy	Moderate	Difficult
Cost	Small	Medium	High
Saleability	Low	Medium	High

Criteria in selection

Criterion	Analytical modeling	Simulation	Measurement
Stage	Any	Any	Postprototype
Time required	Small	Medium	Varies
Tools	Analysts	Computer languages	Instrumentation
Accuracy	Low	Moderate	Varies
Trade-off evaluation	Easy	Moderate	Difficult
Cost	Small	Medium	High
Saleability	Low	Medium	High

- Analytical modeling: to provide the best insight (effects of various parameters and their interactions).
- Simulation: to search the space of parameter values for the optimal combination.
- Measurement: to prove outcomes in practice and also to validate modeling and simulation.

Rules of thumb

Until validated, all evaluation results are suspect.

- A simulation model trusted if validated by analytical modeling, measurements
- An analytical model trusted if validated by simulation modeling, measurements
- A measurement trusted if validated by analytical modeling, simulation modeling

Rules of thumb

Until validated, all evaluation results are suspect.

- A simulation model trusted if validated by analytical modeling, measurements
- An analytical model trusted if validated by simulation modeling, measurements
- A measurement trusted if validated by analytical modeling, simulation modeling

Techniques can be used sequentially.

1. Simple analytical modeling to find range of system parameters
2. Simulation to study performance in that range ⇒ reducing number of simulations.

Outline

1 Technique selection

2 Metric selection
- Case study

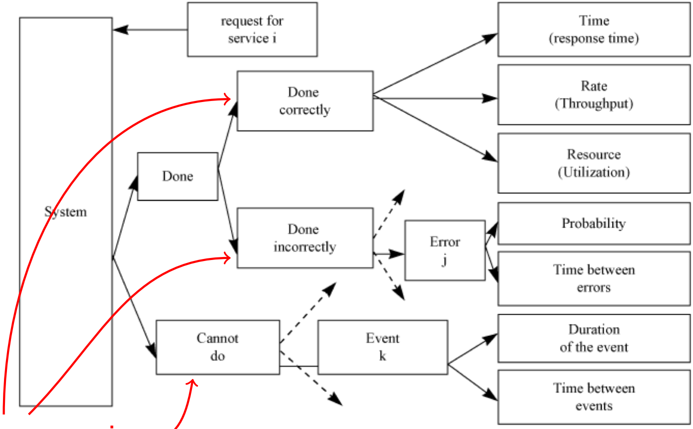
3 Common performace metrics

4 Utility classification of metrics

A systematic way to metrics

Path to metrics

List system's services → list outcomes per service → determine metrics per outcomes.



3 outcome categories

Selecting metrics (1)

- Done correctly:

  Time-rate-resource $\equiv$ responsiveness-productivity-utilization
- Done incorrectly: rate, probability of errors
- Cannot do: time to failure and duration

- (Computer network) Responsiveness $\ni$ response time
- (Operating system) Productivity $\ni$ throughput
- (System) Highest utilization $\equiv$ bottleneck
- (Computer network) Timeout rate

Selecting metrics (2)

Correct service	Incorrect service	Not service
Time	Rate	Resource
Responsiveness	Productivity	Utilization
Speed	Reliability	Availability

Aspects of a metric

- Mean and Variability: both need to be considered.
- Global and individual
  - Resource utilization, reliability, availability: global metrics.
  - Response time, throughput: individual and global metrics.
  - Only using system (global) or individual thoughtput ⇒ unfair situations.

Congestion control algorithms

A service and its outcomes

System definition

- A computer network consists of a number of end systems interconnected via a number of intermediate systems.
- Intermediate systems forward the packets along the right path.
- Congestions occur when
  - Number of packets waiting at intermediate systems $>$ their buffer capacity.
  - Some packets have to be dropped.

Congestion control algorithms

A service and its outcomes

System definition

- A computer network consists of a number of end systems interconnected via a number of intermediate systems.
- Intermediate systems forward the packets along the right path.
- Congestions occur when
  - Number of packets waiting at intermediate systems $>$ their buffer capacity.
  - Some packets have to be dropped.

- Service: Send packets from specified source to specified destination in order.
- Possible outcomes:
  - Some packets are delivered in order to the correct destination.
  - Some packets are delivered out-of-order to the destination.
  - Some packets are delivered more than once (duplicates).
  - Some packets are dropped on the way (lost packets).
  - ...

Packet delivery service

Done correctly: delivered in order

- Time-rate-resource
  - Response time to deliver the packets
  - Throughput: the number of packets per unit of time.
  - Processor time per packet on the source end system.
  - ...
- Variability of the response time → retransmissions
  - Response time: the delay inside the network.

Packet delivery service

Done incorrectly: out-of-order delivery

- Out-of-order packets consume buffers → Probability of out-of-order arrivals.
- Duplicate packets consume the network resources → Probability of duplicate packets.
- Lost packets require retransmission → Probability of lost packets.
- Too much loss cause disconnection → Probability of disconnect.

Packet delivery service

Sharing system: a fairness metric

- Given set of user throughputs: $x_1, x_2, \ldots, x_n$.
- Fairness metric (Jain's fairness index)

$$f(x_1, x_2, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2} = \frac{\bar{\mathbf{x}}^2}{\mathbf{x}^2}.$$

- Fairness Index Properties
    - Always lies between 0 and 1.
    - Equal throughput ! Fairness $= 1$.
    - If k of n receive x and $n - k$ users receive zero throughput: the fairness index is k/n.

Outline

1 Technique selection

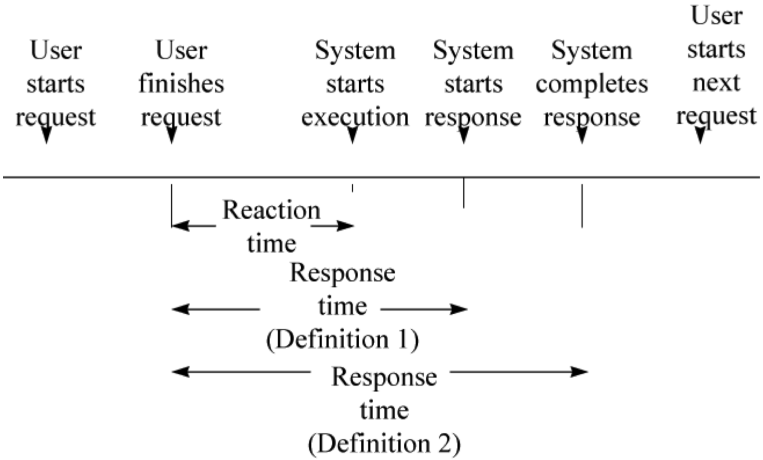
2 Metric selection

- Case study

3 Common performace metrics

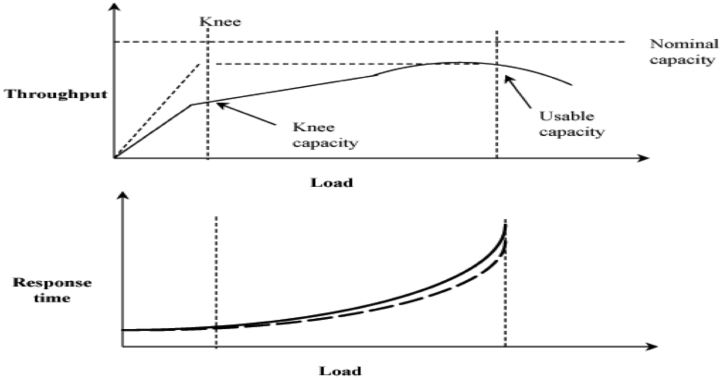
4 Utility classification of metrics

Response time, reaction time, turnaround time



- Turnaround time: the time between the submission of a batch job and the completion of its output.

Throughput
Capacity vs. load (1)



- Jobs/requests per second
- Millions of Instructions Per Second (MIPS)
- Millions of Floating Point Operations Per Second (MFLOPS)
- Packets Per Second (PPS)
- Bits per second (bps)
- Transactions Per Second (TPS)

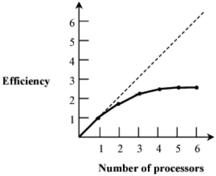
Throughput

Capacity vs. load (2)

- Nominal Capacity: Maximum achievable throughput under ideal workload conditions.
- Usable capacity: Maximum throughput achievable without exceeding a pre-specified response-time limit.
- Knee Capacity: Knee = Low response time and High throughput ⇒ optimal operating point

Efficiency and Utilization

- Efficiency: ratio between usable capacity and nominal capacity
  - Example: maximum throughput of 100Mbps LAN = 85 Mbps ⇒ Efficiency = 85%.
  - (Multiprocessor system): Efficiency = ratio of the performance of an n-processor system to that of a one-processor system.



- Utilization: fraction of time the resource is busy servicing requests.

Reliability and Availability

- Reliability
  - Probability of errors.
  - Mean time between errors (error-free seconds).
- Availability
  - Mean Time to Failure (MTTF).
  - Mean Time to Repair (MTTR).
  - MTTF/(MTTF+MTTR).

Outline

1. Technique selection

2. Metric selection
   - Case study

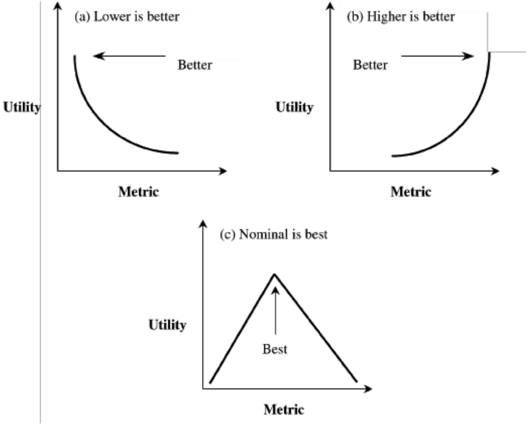
3. Common performace metrics

4. Utility classification of metrics

Utility classification

Which values are better/worse?

- Higher is Better (HB): e.g., System throughput
- Lower is Better (LB): e.g., Response time
- Nominal is Best (NB)



Setting performance requirements

SMART

- *Non-Specific*: No clear numbers are specified.
- *Non-Measurable*: No way to measure/verify with requirements.
- *Non-Acceptable*: Low numerical values in order to be realistic ⇒ unacceptable.
- *Non-Realizable*: High performance ⇒ unrealizable.
- *Non-Thorough*: no all possible outcomes.

Setting performance requirements

SMART

- *Non-Specific*: No clear numbers are specified.
- *Non-Measurable*: No way to measure/verify with requirements.
- *Non-Acceptable*: Low numerical values in order to be realistic ⇒ unacceptable.
- *Non-Realizable*: High performance ⇒ unrealizable.
- *Non-Thorough*: no all possible outcomes.

The system should be both processing and memory efficient. It should not create excessive overhead.