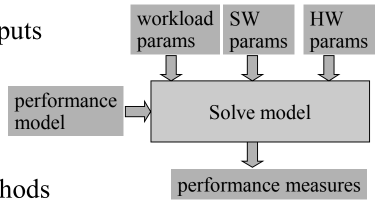
# Lecture 4
# Performance Evaluation Models

Building a Model
Multiple Class Models
Baseline Model
Modification Analysis



## Performance Model

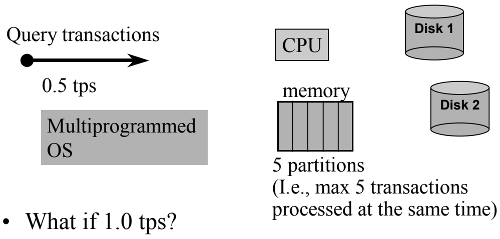
- Inputs & outputs




- Solution methods
  - depend on model
  - trivial: rules of thumb
  - complex: analytical, simulation, benchmarks



## Example: Database Server




- What if 1.0 tps?
  - need faster CPU? or more memory?
- Queues? Resources? Active? Passive?
- Use of resources? service time?



## Typical Transaction T

- Acquire memory partition
  - queue for memory?
- Use CPU ….
  - queue for CPU?
- …. until
  - I/O operation: use (and queue for) disk
  - timeslice expires: give CPU to next job
  - transaction completes: release mem & depart



## Active resource

- Server, device
- Gives service
- Must have in possession during service
- Kept only during service
- Waiting queue or line
- Speed or rate of service, service time
  - parameter to model?
  - aver. value? distribution?

  E.g., aver 4.6 ms
- Fig. 3.2 [Men 94]

  CPU, Disks



## Passive Resource

- Allocated, reserved
- Deallocated, freed
- Waiting queue or line
- Must have in order to proceed

  Memory
- Kept until deallocated
  - difficult for Markov Chain based analytical solutions
  - trouble: simultaneous resource possession
- Figs 3.3 & 3.4

## Queueing Network (QN)

- Network of Queues
- Open Queueing Network
  - Database server: Fig. 3.5 [Men 94]
  - system, arrivals
  - transitions, transition probabilities
  - queues, queue lengths
  - subsystem, easy to solve!
- Service demands ($D_i$)
  - Tbl 3.1

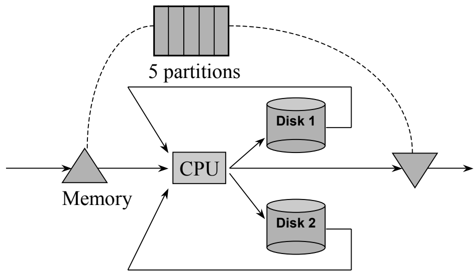
Operational Analysis

$D_i = (U_i * T) / C_0$ — Utilization ($U_i$); Total observation time ($T$); Completions from system ($C_0$)

$D_i = V_i S_i$



## Open Queueing Network






## Service Demand $D_i$

$D_i = (U_i * T) / C_0$

$= U_i / (C_0 / T) = U_i / X_0$

$= V_i S_i$

Device Utilization ($U_i$); Total observation time ($T$); Completions from system ($C_0$); System throughput ($X_0$); Device Service Time ($S_i$); Device Visit Ratio = Nr of visits to device ($V_i$)



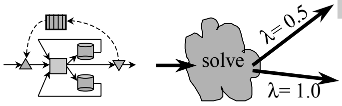
## Example Open QN Model

- Fig. 3.5
- Minimum response time: $T_0 = \sum_{i=1}^{K} D_i$
- Queueing time? $T = T_0 + W_{mem} + W_{cpu} + W_{disk1} + W_{disk2}$
- Average arrival rate $\lambda = 0.5$ tps
- Maximum degree of multiprogramming
  - how many jobs in subsystem?
  - $N^{max} = 5$



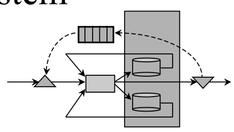
## How to Solve System Model?

- Depends on arrival rate!
  - Tbl 3.2
- Easy to solve with light load
- More complex to solve with heavy load
- What is $\lambda_1$ ?






## Disk Subsystem



- I/O channels? SCSI?
- Heads of strings?
- Device controllers
- Disk cache
- File access protocol
- Rotation speed?

Average demand
Si = 0.75 sec

OK or not OK?

## Interactive Systems

(osituskäyttö systeemit)

- People are part of model
- Terminals, work stations, …
- Response time (R), Think time (Z)
- Fig. 3.6
- Tbl 3.7
- How to solve?
  - depends on number of terminals
  - Tbl 3.4



## Batch Systems

(eräajosysteemit)

- No people
- Closed system
- Fixed number of multiprogramming level
- Tbl 3.5



## Multiple Job Classes

- More difficult than before
  - More parameters to estimate
  - More complex to solve
- Gives more usable information
- Open model: class arrival rates
- Closed model: class populations
- Tbl 3.6



## Aggregating Classes

(luokkien yhdistely)

- Combine classes to make model simpler
  - aggregate all uninteresting job classes together?
- To make parameter estimation simpler
- Tbls 3.7 & 3.8
- Need to compute derived parameter values for aggregate class from those of component classes
  - method varies depending on network type
  - Figs 3.9, 3.10 & 3.11



## Priorities

- Priorities are used in real systems
  - CPU, disks, etc
- May be dynamically changing
- Difficult to model well
- Models with priorities are more complex to solve (than those with no priorities)
- Example with Tbl 3.9



## Shared Domains

(yhteisalueet)

- Class limits or passive resources shared with other classes
  - multiprogramming level
  - memory partitions
  - Fig. 3.12

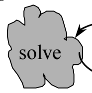
## Multiple Class Model Parameters

- Tbl 3.10



## Baseline Model and Modification Analysis

- Baseline: Tbl 3.11
- A: Use DBMS: Tbl 3.12
- B: Use DBMS + Optimizing compiler
  - $D_{CPU}$ down 50%
  - only for applications, 40% of CPU path length
  - So, $D_{CPU}$ really down only 20%




| | Trivial | Complex |
|---|---|---|
| $D_{CPU}$ | 0.2 | 0.45 |
| R | 1.77 | 3.09 |



## Baseline Model and Modification Analysis (contd)

- C: DBMS + Larger DB record buffer pool (I.e., larger disk cache)
  - $D_{DISK1}$ and $D_{DISK2}$ down 30%
  - Tbl 3.13

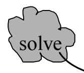


## Baseline Model and Modification Analysis (contd)

- D: Use also transaction logging for crash recovery?
  - disk update needs 2048 B record (38.7 msec)
  - logging only for complex transactions
  - assume each complex transaction causes one disk update, and so one log update (to DISK1)
  - $D_{DISK1}$ up 0.0387 sec
  - Tbl 3.14

HW specs

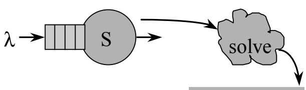
SW specs




| | trivial | complex |
|---|---|---|
| R | 1.17 | 2.26 |



## Solution Methods




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

- Analytical
  - much more complex for complex models
- Approximate
  - approximate reality with simpler model
  - approximate exact solution for complex model
- Simulation
  - Monte Carlo
  - statistical analysis