Lecture 12
Parameter Estimation
Summary

HW/SW Monitors
Events/Sampling
Overhead
Example
Summary

Order Processing Example

- Create model, Fig 9.1
- How to get parameter values?
  - OS
  - TP monitor
  - DBMS
  - Application

Customers	disk A
Inventory	disk A?
Orders	disk B
Paging	disk C

Job classes:	Order entry	init R=3.1 sec
	Order query	

Measurement Process

- See Fig. 9.2
- Which data is needed?
- Where to place instrumentation?
- When to collect data?
- How long to collect data?
- How to account for measurement error?
  - load caused by the instrumentation?
  - does instrumentation have side effects?

HW Monitors

- Collect raw data
- Wire monitors to physical units

Fig. 2.7 & Tbl 2.3 [Ferrari 78]

- HW tool characteristics (1978)

Tbl. 2.2 [Ferrari 78]

- Can be made to work with no (or very little) side effects or load
  - why?

SW Monitors

- Modify SW to collect data

Application? OS? DBMS?

- Use off-the-shelf monitoring system

OS? DBMS? Other?

- Problem: runs in the same system that is being monitored
  - biased data. uses system resources
- Compare to HW monitors

Tbl 7.1 [Jain 91]

Use Accounting Data for Parameter Estimation

- Good: it is there already (or is it?)
- Bad: it is built for accounting purposes
- Who used resources?
  - user id, program, project, account nr or class
- What resources were used?
  - CPU time, I/O op counts, NW use, mem use
- When were resources used?
  - user time only?
  - Granularity may be too coarse?

Program Analyzers

- Runs with some programs
- Designed for one program

IBM DB2, IMS, CICS

- Generic analyzers
  – gprof
  – get nr. of transactions, CPU time, mpl

Hybrid Monitors

- HW + SW

Special HW monitor —events→ SW monitor

- Data filtering
- Post run analysis
- Can be very large monitoring system
  – as large as actual monitored system

same system?
different system?

[Wybranietz & Haban]

Events or Sampling

- Event driven or time driven data gathering?
- Volume?
- Data collection method?
- Data storage? Where?
- Overhead?

Events

- Predefined system events
  – “I/O request completed”
  – “process terminates”
- Create events
  – how much overhead?
- Log them into log files, or
- Send them to filtering and analysis

Sampling

- Good: smaller volume
- Bad: may miss something
- Sampling rate? too high? too low?
- Collect state information
  – “CPU busy”, “disk busy”
  – “amount of memory at use”
- Assign data to threads/processes/programs?
  – “CPU busy 5 ms due to process P” ???

Measurement Unit

- Module
- Subroutine
- Statement
  – language dependent?
- Machine instruction?

Time Unit

- Real time?
- System time?
- External time?
- CPU time?
- Elapsed time?

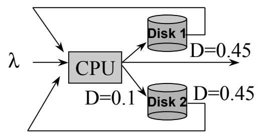
Instrumentation

- When?
  - source code
  - compiler generated code
  - before linking, or at load time
  - run time
- How?
  - source code? Obj. code?
  - run time env.? OS? HW?

Report

- Hierarchical
- Graphics
- What time is used?
  - inherited time
  - own time

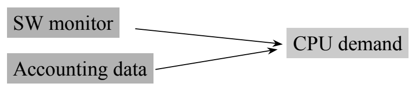
Parameter Values



- Job classes
  - clustering? factor analysis?
- N_r λ_r Z_r M_r
  - external to system: plain measurements
- D_{ir}
  - internal: need internal data - e.g., U_{ir}
  - $D_{ir} = U_{ir} * T / C_{0r}$
  - sometimes get only U_i

Parameter Example

Fig. 9.3 [Men 94]



Transaction Processing System Data

components not known

- Have $U_{total} = U_{cpu,\,OS} + U_{cpu,TP} + U_{cpu,prog}$
- Need $U_{cpu,r}$ for each transaction class r

$$U_{cpu,r} = U_{cpu} * f_{cpu,tp} * f_{tp,r}$$

$$\frac{U_{CPU,TP}}{\sum_s U_{CPU,s}} \qquad \frac{T^{TP}_{CPU,r}}{\sum_{s \in TP} T^{TP}_{CPU,s}}$$

sampling system monitor

analyzer for TP monitor

TP Example (1)

- System with 3 classes:
  - Batch (B), Interactive (I)
  - TP (queries & updates)

  for our TP model we need demands for these!
- System monitor: T=1800, $U^{total}_{cpu} = 72\%$
- Accounting: $U^{no\,OS}_{cpu,B} = 32\%$ $U^{no\,OS}_{cpu,I} = 10\%$ $U^{no\,OS}_{cpu,TP} = 28\%$
- TP analyzer:
  - 1200 queries in 120 sec
  - 400 updates in 140 sec

  0.35 sec per update
- Now,

$$U_{cpu,U} = 0.72 * \frac{28}{70} * \frac{140}{260} = 15.5\%$$
$$D_{cpu,U} = U_{cpu,U} X_{0,U} = 15.5\% * \frac{1800}{400} = 0.70\text{sec}$$
$$D_{cpu,Q} = \left[0.72 * \frac{28}{70} * \frac{140}{260}\right] * \frac{1800}{1200} = 0.20\text{sec}$$

Arrival Rate & MPL

- Assume flow balance
  - arrival count ≅ completion count
  - $\lambda_r = C_{0r} / T$
- Average mpl:

$$\overline{N} = \lambda\, R = \frac{1}{T/n} * \frac{\sum_i e_i}{n} = \frac{\sum_i e_i}{T}$$

elapsed time for job i

nr of jobs

Fig 9.4 [Men 94]

M and Z (2)

- M - Number of terminals

session length (from accounting logs)

$$\overline{M} = \text{"session rate" * "session time"} = \frac{n}{T} * \frac{\sum S_i}{n} = \frac{\sum S_i}{T}$$

- Z:

$$Z = \frac{M}{X} - R = \frac{M}{C_0/T} - R$$

average measured elapsed time in system

Example: T =1 hour = 3600 sec
M=40 4900 interactive commands
2.5 sec aver resp time
Z = 40 / (4900/3600) - 2.5 = 29.4 - 2.5 = 26.9 sec.

CPU Demand: $D_{cpu,r}$

$$D_{cpu,r} = U_{cpu,r} * \frac{T}{C_{0,r}} = U^{Total}_{cpu} * f_{cpu,r} * \frac{T}{C_{0,r}}$$

where $f_{cpu,r} = \frac{C_{0,r}}{\sum_s C_{0,s}} = f(jobs)$

or $f_{cpu,r} = \frac{U^{mon}_{cpu,r}}{\sum_s U^{mon}_{cpu,s}} = f(time)$

or $f_{cpu,r} = \frac{nrIO(r)}{\sum_s nrIO(s)} = f(\text{nr IO's})$

or …

Capture Ratio (1)

- Capture ratio $C_r = \frac{measured\ T_{cpu,r}}{actual\ T_{cpu,r}}$
  - class dependent!
- Example:

measure estimate calibrate quess

$C_{batch} = 0.80$ $C_{term} = 0.60$
$T = 7200$ sec $U^{Total}_{cpu} = 64\%$
$T^{Meas}_{cpu,batch} = T^{Meas}_{cpu,term} = 1500$ sec
$T_{cpu,batch} = \frac{1500}{0.80} = 1875$ sec (actual!)
$T_{cpu,term} = \frac{1500}{0.60} = 2500$ sec $T_{cpu} = 4375$ sec
$U_{cpu,batch} = U^{Total}_{cpu} * f_{cpu,batch} = 0.64 * \frac{1875}{4375} = 0.27$

actual, not measured

D_{ir} for Disks

IO count Mean service time

$$U^{Total}_i = \frac{IOC_i * MST_i}{T} \text{ (or from HW monitor)}$$

$$D_{ir} = \frac{U^{Total}_i * T}{C_{0,r}} * f_{ir} = \frac{IOC_i * MST_i}{C_{0,r}} * f_{ir}$$

where $f_{ir} = \frac{IOC_{ir}}{IOC_{0,r}}$ (user data)

or $f_{ir} = \frac{SWAPC_{ir}}{SWAPC_{0,r}}$ (swap data)

or $f_{ir} = \frac{PageInC_{ir}}{PageInC_{0,r}}$ (paging data)

Delay Node

- High disk utilization: include all disks in model
- Low (< 5%?) disk utilization: may use one aggregate **delay** node in model
  - no queueing likely ….

$$D_{ir} = \sum_{disks\ j} \frac{U_j^{Total} * T}{C_{0,r}} * f_{jr} = \frac{T}{C_{0,r}} \sum_{disks\ j} U_j^{Total} * f_{jr}$$

Order Processing Example

Tbl 9.1 [Men 94]

- Want: Arrival rates, demands
- CPU demands — Tbl 9.3
- File ops per application class — Tbl 9.4
- File ops per disk — Tbl 9.5
- Disk util. & demands per class — Tbl 9.6
- Z_r's when R_r's are know: use Little's Law

Estimate Z_r's if R_r unknown

- Use subsystem as open model
- Use $\lambda_r = X_{0r}$
- Solve model, get R_r = (3.00, 0.98, 2.23)
- Use those estimates to compute
  $Z_r = M_r / X_{0r} - R_r$

Z_{OE} = 10/1.43 - 3.0 = 3.99 (vs 3.75)
Z_{OI} = 5/0.83 - 0.98 = 5.04 (vs 5.04)
Z_{Ot} = 5/0.31 - 2.23 = 13.9 (vs. 13.9)

Calibration

- Use parameters to build baseline model
- Use solver
  - e.g., PMVA — fig.9.1.out
  - get R_{OE} = 2.5 (not 3.25 as in Tbl 9.1)
- What now? Calibrate model!
  - scale (or add to) output to match data
  - adjust MPL or its distribution
  - add new class — fig.9.1a.out
  - modify demands (at botleneck) — fig.9.1b.out
  - add ghost server for unaccounted for load

Summary

Probability Theory
Modelling
Solution Methods
Approximations
Parameter Estimation

Probability Theory

- Distributions
- Sample
- Variance
- Confidence intervals

Modeling

- Capacity planning process
- System model
- Workload model
- Capacity
- Performance

Solution Methods

- M/M/1, Markov Chain
- Operational analysis
- Closed models
  - Convolution
  - MVA
  - Approximate MVA
- Open models

Approximations

- Flow equivalent server
- Aggregate model
- Iterative solutions
  - multiple class memory
- Load concealment
- Shadow server

Parameter Estimation

- Measurement process
- HW/SW monitors
  - hybrid monitors
- Accounting data
- Program analyzers
- Parameter values
  - combine data from various sources
  - fractions
  - overhead