Performance Forecasting
- Introduction -

Matthias Mann

HII3DB

HVB Information Services

Contents

Forecasting – Brief Explanation 1/1

- Forecasting is everywhere: weather, traffic, … IT
- Forecasting ⇔ risk assessment
- Forecasting is not Tuning
- Forecasting as part of SLM:
  - Capacity-, Continuity-, Availability-, Financial Management
- Performance Forecasting Focus:
  - Does risk exist?
  - What is at risk?
  - When will risk occur?
  - How to minimize the risk?

Performance Forecasting: Methodology and Models 1/7

(1) Formulate the question of the study

e.g.: Workload of the database server is expected to double in the next three months. Will business requirements be met? If not, which areas require attention?

(2) Gather workload data

- automated, repeatable
- use repository
- application instrumentation?
- measure all components (OS, DB, Middleware, Application)
- minimize impact on performance

Performance Forecasting: Methodology and Models 2/7

(3) Characterize the data

- make they useful and understandable
- workload representation: basic unit of work (transaction) ?
- find the baseline

(4) develop / choose an appropriate model

- single / multiple component model
- input data: application / os / database related?
- productive / planned system
- time available for forecasting
- required precision of forecast

Performance Forecasting: Methodology and Models 3/7

Common fundamental models

(4a) Ballpark Figures

- very quick, rough forecasts: Server with 24 GB Memory runs 15 DB instances (each with SGA/PGA). Does it support four more instances?

(4b) Ratio Modelling

- quick, low precision
- relation between process categories (OLTP/Batch) and system resources (CPU)
- for budgeting approx., architecture validations, sizing of packaged appl.

Performance Forecasting: Methodology and Models 4/7

(4c) Linear Regression Analysis

- typical question: How much of some business activity can occur before the system is running "out of gas" ?
- precise, statistically validated
- solving of a system of equations

(4d) Queueing Theory

- understanding of performance issues for forecasting

Performance Forecasting: Methodology and Models 5/7

Model	Ballpark Figures	Ratio Modelling	Linear Regression	Queueing Theory
components				
- single	x	x	x	x
- multiple				x
Input Data				
- Application	x	x	x	x
- Infrastructure	x	x	x	x
Usable for				
- living systems	x	x	x	x
- planned systems		x		x

Performance Forecasting: Methodology and Models 6/7

Model	Ballpark Figures	Ratio Modelling	Linear Regression	Queueing Theory
Precision				
- low	x	x		x
- high			x	x
Project Duration				
- short		x		x
- long			x	x

Performance Forecasting: Methodology and Models 7/7

(5) Model Validation: determine precision

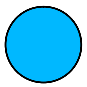
check:

- numerical error
- statistical error (average, standard deviation, skewness)
- histogram analysis
- residual analysis (areas of good/bad forecast)
- go / no-go decision

(6) Forecast

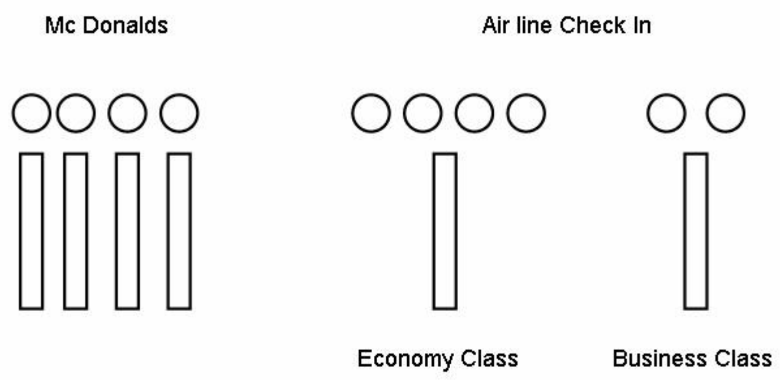
- answer the study question

Queueing Theory: Basics 1/4

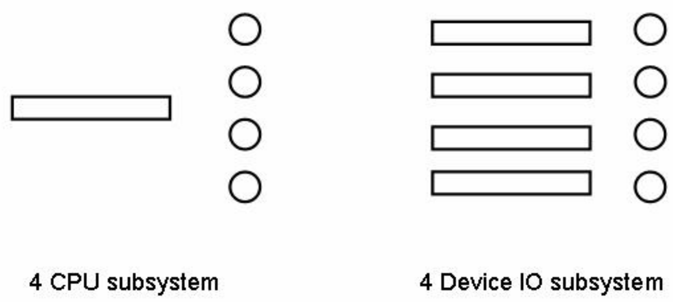
- most environments are queueing systems:
  - fuelling station
  - restaurant, supermarket, …
  - call center
  - …
- Notation:  Server (CPU, IO Device)

 Queue (waiting transactions)

Examples



Mc Donalds
Air line Check In
Economy Class
Business Class



4 CPU subsystem
4 Device IO subsystem

Queueing Theory: Basics 3/4

- Foundation

  John D. Little   MIT Sloan School of Management

  - Operations Research
  - Traffic Control
  - Marketing

- Littles Law: $N = \lambda T$

```
avg. # in system (served+queued) = arrival rate
                                   times
                                   avg time spent in system (resp.time).
```

Queueing Theory: Basics 4/4

- D.G. Kendall, Prof. of Math. Statistics at Cambridge:

Kendalls Notation (for a single queue): `A / B / m`

A … arrival pattern (distribution)

B … service time distribution

m … # of servers in the system

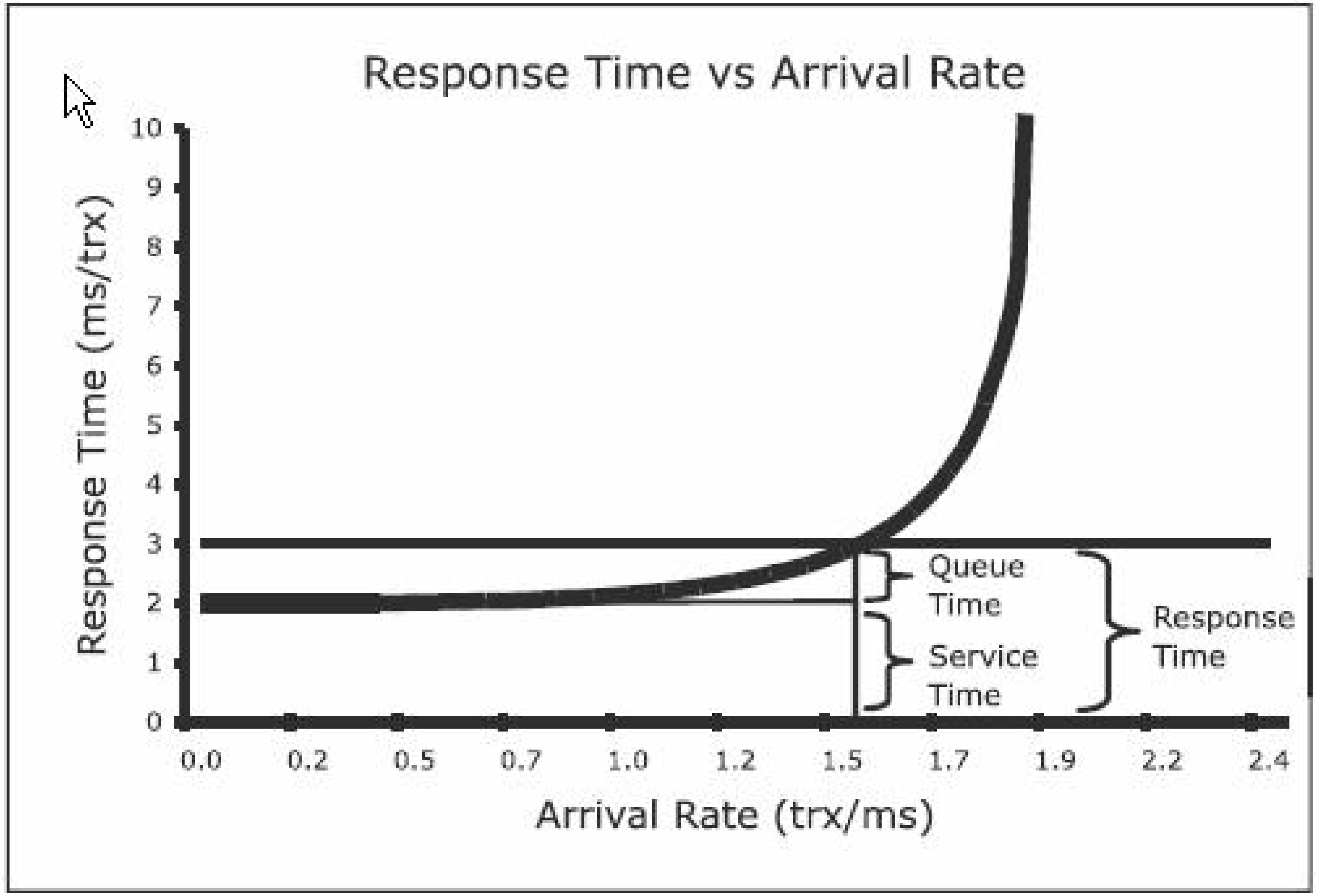
- computing transactions follow Markovian distribution
  - exponential, each transaction independent from others
- Examples:
  - `1x M / M / 12` 12 CPU Oracle Server
  - `5x M / M / 1` 5 Device IO subsystem

Queueing Theory: Terms

Term	Symbol	Definition	Unit
transaction		basic unit of work	trx
arrival rate	λ	# of trx entering the system within a time period	trx/ms
Service Time	S_t	how long it takes a server to service a transaction	ms/trx
Queue Length	Q	# of trx in the queue (goal: 0)	trx
Utilization	U	amount of time a resource is in use over a given time interval	%
Queue Time	Q_t	time a trx waits in the queue until served	ms/trx

Queueing Theory: Terms 2/3

Term	Symbol	Definition	Unit
Response Time	R_t	time of the trx in the system: $R_t=Q_t+S_t$	ms/trx
# of servers in the system	M		
# of servers per queue	m		



Response Time vs Arrival Rate
Response Time (ms/trx)
10
9
8
7
6
5
4
3
2
1
0
0.0
0.2
0.5
0.7
1.0
1.2
1.5
1.7
1.9
2.2
2.4
Arrival Rate (trx/ms)
Queue Time
Service Time
Response Time

- simplified version: underestimation of response time
- for IO subsystem: M=1

$$U = \frac{S_t \lambda}{M}$$

$$Q = \lambda Q_t$$

$$R_t = \frac{S_t}{1 - U^M} = S_t + Q_t$$

Queueing Theory: Basic Formulas 2/3

Erlang C Mathematics

- A. K. Erlang: telephone engineer, worked in traffic engineering and queueing theory => Erlang C Unit
- Traffic of 1 Erlang refers to a single resource being in continuous use. Used in telephony as a statistical measure of telco traffic.
- Efficiency of call centers (calls / hour, avg. call duration, target answer time, …)
- Application to computing systems:
  - CPU (single queue): o.k.
  - IO (multiple queue): conversion to multiple single queue / single server system necessary (perfectly balanced IO ssystem across all devices)

Queueing Theory: Basic Formulas 3/3

$$\lambda_q = \frac{\lambda_{sys}}{Q_n}$$

$$U = \frac{S_t \lambda_q}{m}$$

$$E_c = E_c(m, S_t, \lambda_q) = \frac{\frac{(m S_t \lambda_q)^m}{m!}}{(1 - m\lambda_q S_t) \sum_{k=0}^{m-1} \frac{(m S_t \lambda_q)^k}{k!} + \frac{(m S_t \lambda_q)^m}{m!}}$$

$$Q_t = \frac{E_c S_t}{m(1-U)}$$

$$Q = \lambda_q Q_t$$

Data Gathering and Workload Characterization 1/12

- minimize impact on system
- all data sources have to start / end at the same time

- Data Sources
  - OS: `sar, vmstat, iostat, …`
  - Application:
    - # of active OLTP sessions / Batch processes
    - # of orders per hour, …

Data Gathering and Workload Characterization 2/12

- Oracle Database:
  - gather at system / session level
  - use dynamic performance views
    - `v$sysstat, v$session`
    - `v$sysstat` joined with `v$statname`
    - `v$sess_io, v$mystat`

Data Gathering and Workload Characterization 3/12

- gatherdata.sh

```
#
# Settings
#
ora_access=/            # Login in the Database
seq=$(date +%H%M)       # Clock Time
cpu_file=${FCDIR}/dat/wl.cpu.$(uname -n).${Y}.${M}.${D}.dat
io_file=${FCDIR}/dat/wl.io.$(uname -n).${Y}.${M}.${D}.dat
app_file=${FCDIR}/dat/wl.oracle.${ORACLE_SID}.${Y}.${M}.${D}.dat
```

Data Gathering and Workload Characterization 4/12

```
sqlplus $ora_access <<EOF
drop table wl_stats;
col value format 9999999999999
set linesize 500
create table wl_stats as
select * from v\$sysstat
where name in (<any statistic#>);
exit;
EOF
```

Data Gathering and Workload Characterization 5/12

```
#
# Start CPU gathering
#
sar -u -o $sar_file $DURATION 1 >/dev/null 2>&1 &
#
# Start IO gathering
#
iostat -xn $DURATION 2 >$iostat_file &
#
# Wait for CPU and IO gathering to complete
wait
```

Data Gathering and Workload Characterization 6/12

```
#
# Gather final Oracle workload values and calculate delta activity
#
sqlplus $ora_access <<EOF  >>${log_file}
select '$seq'||','||b.name||',' the_line,
       b.value-a.value value,
       ',good' xxx
from (select name,statistic#,value from v\$sysstat
      where name in (<any statistic#>)) b,
     (select name,statistic#,value from wl_stats
      where name in (<any statistic#>)) a
where b.statistic# = a.statistic#;
exit;
```

Data Gathering and Workload Characterization 7/12

```
#
# Print general Oracle workload statistics
#
grep good $work_file | grep -v xxx | awk -F, '{print $1 "," $2 "," $3}' >>$app_file
```

Data Gathering and Workload Characterization 8/12

- crontab entry:

```
0,5,10,15,20,25,30,35,40,45,50,55 * * * *
   /opt/oracle/admin/BIP6T8/perf/bin/gatherdata.sh -s BIP6T8 -d 300
```

- CPU data collected: (`time, %usr, %sys, %wio, %idle`)

```
1345,59,5,1,35

1350,59,12,1,28

1355,57,13,1,28

1400,66,7,1,27

1405,57,5,1,37
```

Data Gathering and Workload Characterization 9/12

- IO data collected: (time, device, selected device statistics)

```
1345,d40,0.0,0.1,0.2,1.2,1,3
1350,d30,0.0,0.0,6.8,1.0,0,0
1350,d40,0.0,0.0,0.2,0.6,1,2
1355,d30,0.0,0.0,9.8,1.0,0,0
1355,d40,0.0,0.0,0.2,0.4,1,2
1400,d30,0.0,0.0,7.4,0.8,0,0
1400,d40,0.0,0.1,0.1,0.6,1,8
1405,d30,0.0,0.0,6.8,1.1,0,0
```

- Oracle Data collected:

1345,logons cumulative,	1
1345,user calls,	46
1345,physical reads,	801
1345,physical reads cache,	801
1345,physical reads direct,	0
1345,physical read IO requests,	801
1345,physical read bytes,	6561792
1345,db block changes,	169
1345,physical writes,	276
1345,physical writes direct,	1
1345,physical writes from cache,	275
1345,physical write IO requests,	161
1345,physical write bytes,	2260992
1345,redo size,	38356
1345,execute count,	13069
1345,OS User level CPU time,	0
1345,OS System call CPU time,	0

Data Gathering and Workload Characterization 11/12

- define: what is the transaction / workload?
- single category model: use any suitable column from `v$sysstat`
- multiple category model: grouping of Oracle activity in `v$sesstat`, `v$session`
- important: Baseline construction from the gathered data for later forecasts
    - Baseline: ref. point / interval (usually period of highest utilization during business)
- combine Oracle and OS data

Data Gathering and Workload Characterization 12/12

Example

- CPU subsystem
  - Oracle: `CPU used by this session`

    `user calls` (Transaction)

    `M` from `cpu_count`

    - => S_t=(# trx/(cpu used))
  - OS: `U` (sar)
  - => compute λ=UM/S_t with baseline consideration
    - `select avg ((CPU used)/(user calls)) … where logoff_time …`

Methods by example: Ratio Modelling

- Cook, Dudar, Shallahamer 1995
- works only for one resource type (mostly CPU)

$$P = U\ M = \Sigma\ (C_i\ /\ R_i)$$

P … # of fully utilized CPUs

U … CPU Utilization

C_i … # of workload occurences (eg. OLTP sess., Batch proc.)

R_i … ratio linking a workload category to a CPU

Methods by example: Ratio Modelling 2/4

Example

- 18 CPU database server

  U = 41 %

  250 conc. OLTP User : $R_{OLTP} = 125$

  12 conc. Batch processes: $R_{batch} = 2$

- What is the effect of adding 25 additional OLTP users?

  P = (250+25)/125 + 12/2 = 8.2 = 18 U => U = 46 % => no problem!

Methods by example: Ratio Modelling 3/4

- Question: Where do the ratios come from?
  - have to be derived
  - taken from similar systems, get from application vendor
  - Batch-to-CPU: shutdown OLTP, approximate
  - OLTP-to-CPU: calculate

$$UM = \frac{0}{R_{OLTP}} + \frac{C_{Batch}}{R_{Batch}}$$

$$R_{OLTP} = \frac{C_{OLTP}}{UM - \frac{C_{Batch}}{R_{Batch}}}$$

Methods by example: Ratio Modelling 4/4

- Drawback:
  - low precision
  - unvalidated
  - because of simplicity it can be easily misused

Methods by example: Queueing Theory 1/6

- Situation:
  - company acquisition
  - customers of both companies have to use existing database system
- Question: Is database server able to handle the increased workload?
- Consider: CPU and IO subsystem
- Prerequisite: data gathering is already active
- Workload representation:
  - `user calls` from `v$sysstat` (trx)

Methods by example: Queueing Theory 2/6

- data inspection: Baseline is `323243 trx/ms`
- OS data:
  - CPU: `U = 33%, 32 CPUs`
  - IO: `U= 45%, 60 devices`

Methods by example: Queueing Theory 3/6

CPU subsystem baseline: no queuing

$$Q_n = 1 \quad m = 32$$

$$\lambda_q^{CPU} = \lambda_{sys}/Q_n = 323243trx/ms$$

$$S_t = \frac{Um}{\lambda_q} = 0.033ms/trx$$

$$E_c(m, S_t, \lambda_q) = E_c(32, 0.033, 323243) = 0.0$$

$$Q_t = \frac{E_c S_t}{m(1-U)} = 0.0 \quad R_t = S_t + Q_t = 0.033ms/trx$$

Methods by example: Queueing Theory 4/6

IO subsystem baseline: significant queueing

$$Q_n = 60 \quad m = 1$$

$$\lambda_q^{IO} = \lambda_{sys}/Q_n = 5387trx/ms$$

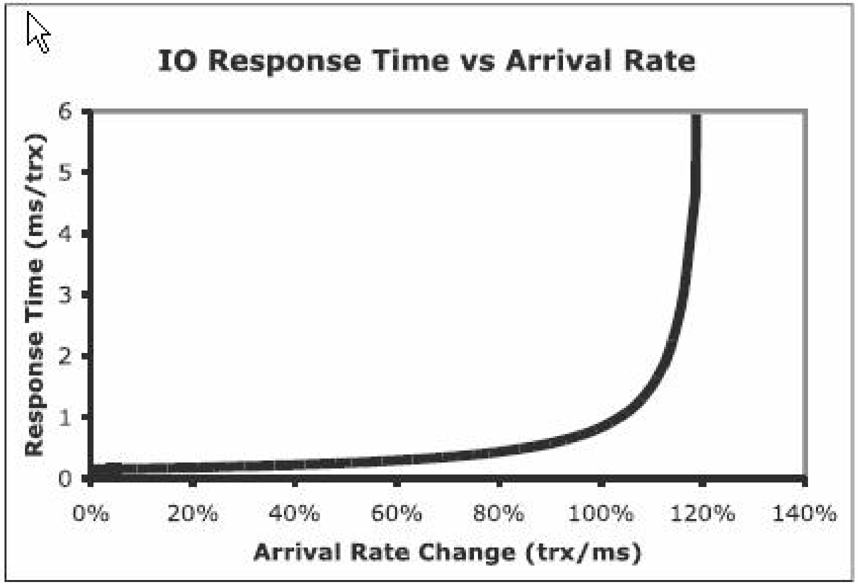
$$S_t = \frac{Um}{\lambda_q} = 0.084ms/trx$$

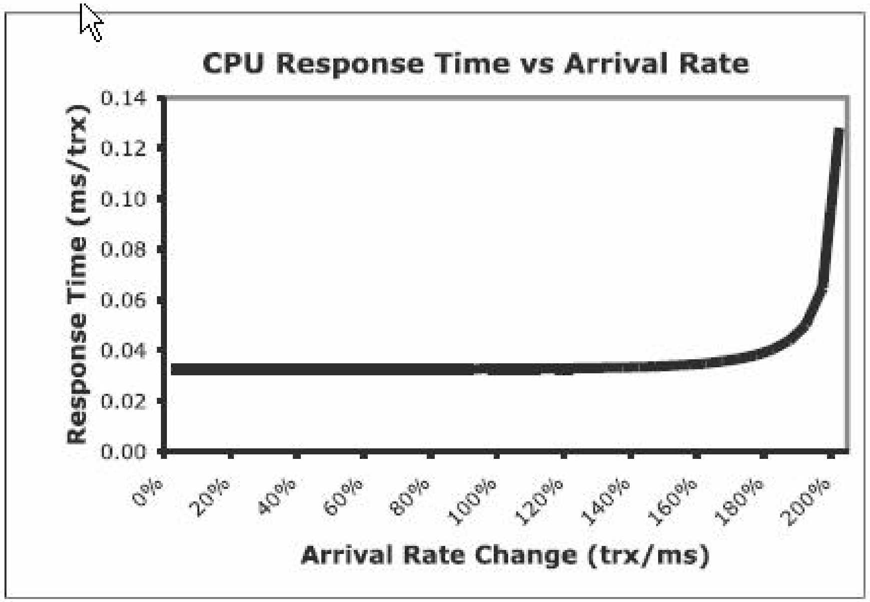
$$E_c(m, S_t, \lambda_q) = E_c(1, 0.084, 5387) = 0.45$$

$$Q_t = \frac{E_c S_t}{m(1-U)} = 0.069 \quad R_t = S_t + Q_t = 0.1522ms/trx$$

Methods by example: Queueing Theory 5/6

put formulas in Excel spreadsheet => produce forecasts





Methods by example: Queueing Theory 6/6

- Summary
  - IO subsystem is bottleneck
- Message to management:
  - workload pattern of acquired customer unknown
  - assuming equal pattern: system may be able to support 100% workload increase
  - risk mitigation:
    - better understand workload pattern of acquired company
    - reduce peak workload (Batch time shifts)
    - continue data gathering
    - new forecast soon after new users are using the system