Abstract

This module addresses Modeling and Analysis Performance. What are the customer performance needs, what are the operational performance considerations? What are the performance related design choices? How to analyze feasibility, explore design options, and how to validate performance?
**goal of this module**

- provide a stepwise approach to system modeling
- provide concrete examples of system models

**content of this module**

- web shop system model
- Non Functional requirements (NFR), System Properties and Critical Technologies
- zero order and first order load models
- budgeting

**exercise**

- model one NFR in relation to a critical technology choice
Abstract

This presentation uses a web shop service as example system to construct a system model. The caching of pictures of the products in the shop is modeled to analyze performance, robustness, scalability and reliability of the system.
content

What to model of the system

Stepwise approach to system modeling

Non Functional requirements (NFR), System Properties and Critical Technologies

Examples of web shop case
What to Model in System Context?

- **usage context**
  - enterprise & users
  - creation
  - life cycle business

- **system**
  - **NFR's:** performance, reliability, availability, scalability, maintainability, ...
  - **(emerging?) properties:** resource utilization, load, latency, throughput, quality, accuracy, sensitivity (changes, inputs), ...
  - critical technologies: caching, load balancing, firewalls, virtual networks, XML for customization and configuration, ...

- **life cycle context**
  - version: 0.4
  - August 21, 2020
  - Modeling and Analysis: System Model
  - version: 0.4
  - August 21, 2020
  - MASMwhatModeling
## Approach to System Modeling

1. determine relevant Non Functional Requirements (NFR's)
2. determine relevant system design properties
3. determine critical technologies
4. relate NFR's to properties to critical technologies
5. rank the relations in relevancy and criticality
6. model relations with a high score
**NFR’s:**
- performance browsing
- initial cost
- running costs
- reliability/availability
- scalability order rate
- maintainability
  - effort product changes
  - effort staff changes
- security

**Emerging?) properties:**
- resource utilization
  - server load, capacity
  - memory load, capacity
- response latency
- redundancy
- order throughput
- product data quality
- product definition flow
- staff definition flow
- security design
  - compartmentalization
  - authentication
  - encryption

**Critical technologies:**
- caching
- load balancing
- pipelining
- virtual memory
- memory management
- data base transactions
- XML for customization and configuration
- firewalls
- virtual networks
- ...
4. Determine Relations

NFR's:
performance browsing
initial cost
running costs
reliability/availability
scalability order rate
maintainability
effort product changes
effort staff changes
security

(emerging?) properties:
resource utilization
server load, capacity
memory load, capacity
response latency
redundancy
order throughput
product data quality
product definition flow
staff definition flow
security design
compartimentalization
authentication
encryption

4 critical technologies

caching
load balancing
pipelining
virtual memory
memory management

data base transactions
XML for customization
and configuration
firewalls
virtual networks

...
5. Rank Relations

NFR's:
- performance browsing
- initial cost
- running costs
- reliability/availability
- scalability
- order rate
- maintainability
- effort product changes
- effort staff changes
- security

(emerging?) properties:
- resource utilization
  - server load, capacity
  - memory load, capacity
- response latency
  - redundancy
  - order throughput

Critical technologies
- caching
- load balancing
- pipelining
- virtual memory
- memory management
- transactions optimization
- configuration
- firewalls
- virtual networks
  ...

ranking will be discussed in Modeling and Analysis: Reasoning
Purpose of Picture Cache Model in Web Shop Context

Modeling and Analysis: System Model

11 Gerrit Muller

version: 0.4
August 21, 2020
MASMwebShopPictureCache
Zero Order Load Model

zero order web server load model

\[ \text{Load} = n_a \times t_a \]

\( n_a = \text{total requests} \)

\( t_a = \text{cost per request} \)
First Order Load Model

first order web server load model

\[
\text{Load} = n_{a,h} \cdot t_h + n_{a,m} \cdot t_m
\]

- \(n_{a,h}\) = accesses with cache hit
- \(n_{a,m}\) = accesses with cache miss
- \(t_h\) = cost of cache hit
- \(t_m\) = cost of cache miss

\[
\begin{align*}
n_{a,h} &= n_a \cdot h \\
n_{a,m} &= n_a \cdot (1-h)
\end{align*}
\]

- \(n_a\) = total accesses
- \(h\) = hit rate

\[
\text{Load}(h) = n_a \cdot h \cdot t_h + n_a \cdot (1-h) \cdot t_m = n_a \cdot t_m - n_a \cdot h \cdot (t_m - t_h)
\]
Load(h) = 1000 * 2[ms] - 1000* h * 1.98[ms]
Load(h) = 2000 - 1980* h [ms]
Hit Rate Considerations

quantified mid office server example

\[ t_h = 0.02 \text{ ms} \]
\[ t_m = 2 \text{ ms} \]
\[ n_a = 1000 \]

Load(h) = 1000 * 2[ms] - 1000* h * 1.98[ms]
Load(h) = 2000 - 1980* h [ms]

Hit rate of well designed system is ample within working range (e.g. 95%) 0th order formula is valid:
Load = 0.12 * n_a [ms]

Hit rate is context dependent. Life cycle changes or peak loads may degrade hit rate.
### Response Time

#### Human Customer
- Press next

#### Client
- Request picture

#### Web Server
- Check cache
- Request picture
- Store in cache
- Transfer to client
- Process
- Display

#### Data Base Server
- Retrieve picture

<table>
<thead>
<tr>
<th>Time (milliseconds)</th>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
</tr>
</thead>
</table>

Time in milliseconds in optimal circumstances
What Memory Capacity is Required for Picture Transfers?
Process View of Picture Flow in Web Server

one copy per process

mid office server
web server
picture cache server
back office access

multiple copies per process

mid office server
web server
picture cache server
back office access

multiple copies per process and thread

mid office server
web server
picture cache server
back office access

Modeling and Analysis: System Model

version: 0.4
August 21, 2020

MASMmidOfficeProcesses
picture memory =

\[ 3 \times n \times s + 5 \times m \times s + c \times s + 3 \times k \times s \]

where

\( n = \# \text{ data base access threads} \)
\( m = \# \text{ picture cache threads} \)
\( k = \# \text{ web server threads} \)
\( s = \text{picture size in bytes} \)
\( c = \text{in memory cache capacity in \# pictures} \)
Web Server Memory Capacity

<table>
<thead>
<tr>
<th>use case</th>
<th>n</th>
<th>m</th>
<th>k</th>
<th>s</th>
<th>c</th>
<th>MB</th>
<th>storage type</th>
<th>picture memory = 3 n s + 5 m s + c s + 3 k s</th>
</tr>
</thead>
<tbody>
<tr>
<td>small shop</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>100</td>
<td>10</td>
<td>1.5</td>
<td>L3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>10</td>
<td>100</td>
<td>20</td>
<td>5.3</td>
<td>main</td>
<td></td>
</tr>
<tr>
<td>highly concurrent</td>
<td>2</td>
<td>4</td>
<td>1000</td>
<td>100</td>
<td>100</td>
<td>296</td>
<td>main</td>
<td></td>
</tr>
<tr>
<td>large pictures</td>
<td>2</td>
<td>4</td>
<td>1000</td>
<td>100</td>
<td>100</td>
<td>2,962</td>
<td>main+disk</td>
<td></td>
</tr>
<tr>
<td>many pictures</td>
<td>2</td>
<td>4</td>
<td>10</td>
<td>100,000</td>
<td>100,000</td>
<td>9,540</td>
<td>main+disk</td>
<td></td>
</tr>
<tr>
<td>all at once</td>
<td>2</td>
<td>4</td>
<td>1000</td>
<td>100,000</td>
<td>100,000</td>
<td>98,234</td>
<td>disk</td>
<td></td>
</tr>
</tbody>
</table>

memory use
product browsing only
pictures only
single server

processor caches
L1 L2 L3 main memory disk
kB MB GB

What is the performance impact of memory use on other processing?
We Have only Modeled a Small Part of the System...

<table>
<thead>
<tr>
<th>function</th>
<th>browse/exhibit products</th>
<th>sales, order intake, payments track, order handling stock handling financial bookkeeping customer relation management update catalogue advertise after sales support</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>data</th>
<th>picture</th>
<th>structured (product attributes, logistics, ...) program code</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>aspect</th>
<th>server memory use response time server load</th>
<th>network use reliability any resource, any NFR</th>
</tr>
</thead>
</table>

aspect result = \[ \sum_{d = \text{all data}} \] \[ \sum_{f = \text{all functions}} \] aspect(d, f)

ignoring other dimensions such as applications, users, circumstances
... to Understand Some of the Systems Aspects

static
mostly assumptions and coarse estimates
some insight in:
what are key design issues
what are relevant use case areas
Refinement After Context Modeling

usage context

enterprise & users

NFR's:
- performance
- reliability
- availability
- scalability
- maintainability
...

(system)

(emerging?) properties:
- resource utilization
- load
- latency, throughput
- quality, accuracy
- sensitivity
  (changes, inputs)
...

critical technologies
- caching
- load balancing
- firewalls
- virtual networks
- XML for customization and configuration
...

creation

life cycle business

life cycle context
## Conclusions

Non Functional Requirements are the starting point for system modeling

Focus on highest ranking relations between NFR's and critical technologies

Make simple mathematical models

Evaluate quantified instantiations

### Techniques, Models, Heuristics of this module

- Non functional requirements
- System properties
- Critical technologies
- Graph of relations
Abstract

This presentation addresses the fundamentals of budgeting: What is a budget, how to create and use a budget, what types of budgets are there. What is the relation with modeling and measuring.
content of this presentation

What and why of a budget

How to create a budget (decomposition, granularity, inputs)

How to use a budget
What is a Budget?

A budget is

a quantified instantiation of a model

A budget can

prescribe or describe the contributions

by parts of the solution

to the system quality under consideration
Why Budgets?

- to make the design explicit
- to provide a baseline to take decisions
- to specify the requirements for the detailed designs
- to have guidance during integration
- to provide a baseline for verification
- to manage the design margins explicitly
Visualization of Budget Based Design Flow

can be more complex than additions

model

measurements existing system

micro benchmarks aggregated functions applications
design estimates; simulations

V4aa


dsims
estimates;
simulations

V4aa

IO

micro benchmarks aggregated functions applications

budget

feedback
tuning

spec

SRS

\( t_{\text{boot}} \) 0.5s
\( t_{\text{zapp}} \) 0.2s

measurements new (proto) system

new (proto) system

model

tuning

feedback

spec

SRS

\( t_{\text{boot}} \) 0.5s
\( t_{\text{zapp}} \) 0.2s
<table>
<thead>
<tr>
<th>step</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A measure old systems</td>
<td>micro-benchmarks, aggregated functions, applications</td>
</tr>
<tr>
<td>1B model the performance starting with old systems</td>
<td>flow model and analytical model</td>
</tr>
<tr>
<td>1C determine requirements for new system</td>
<td>response time or throughput</td>
</tr>
<tr>
<td>2 make a design for the new system</td>
<td>explore design space, estimate and simulate</td>
</tr>
<tr>
<td>3 make a budget for the new system:</td>
<td>models provide the structure measurements and estimates provide initial numbers specification provides bottom line</td>
</tr>
<tr>
<td>4 measure prototypes and new system</td>
<td>micro-benchmarks, aggregated functions, applications profiles, traces</td>
</tr>
<tr>
<td>5 Iterate steps 1B to 4</td>
<td></td>
</tr>
</tbody>
</table>
Budgets Applied on Waferstepper Overlay

- Process overlay, 80 nm
- Matched machine, 60 nm
- Process dependency sensor, 5 nm
- Reticule, 15 nm
- Lens matching, 25 nm
- Single machine, 30 nm
- Matching accuracy, 5 nm
- Stage overlay, 12 nm
- Stage grid accuracy, 5 nm
- Metrology stability, 5 nm
- Global alignment accuracy, 6 nm
- Stage Al. pos. meas. accuracy, 4 nm
- System adjustment accuracy, 2 nm
- Off axis meas. accuracy, 4 nm
- Off axis Sensor repro, 3 nm
- Blue align sensor repro, 3 nm
- Interferometer stability, 1 nm
- Frame stability, 2.5 nm
- Tracking error WS, 2 nm
- Tracking error X, Y, 2.5 nm
- Tracking error phi, 75 nrad
- Tracking error RS, 1 nm
## Budgets Applied on Medical Workstation Memory Use

### Memory Budget in Mbytes

<table>
<thead>
<tr>
<th></th>
<th>code</th>
<th>obj data</th>
<th>bulk data</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>shared code</td>
<td>11.0</td>
<td></td>
<td></td>
<td>11.0</td>
</tr>
<tr>
<td>User Interface process</td>
<td>0.3</td>
<td>3.0</td>
<td>12.0</td>
<td>15.3</td>
</tr>
<tr>
<td>database server</td>
<td>0.3</td>
<td>3.2</td>
<td>3.0</td>
<td>6.5</td>
</tr>
<tr>
<td>print server</td>
<td>0.3</td>
<td>1.2</td>
<td>9.0</td>
<td>10.5</td>
</tr>
<tr>
<td>optical storage server</td>
<td>0.3</td>
<td>2.0</td>
<td>1.0</td>
<td>3.3</td>
</tr>
<tr>
<td>communication server</td>
<td>0.3</td>
<td>2.0</td>
<td>4.0</td>
<td>6.3</td>
</tr>
<tr>
<td>UNIX commands</td>
<td>0.3</td>
<td>0.2</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>compute server</td>
<td>0.3</td>
<td>0.5</td>
<td>6.0</td>
<td>6.8</td>
</tr>
<tr>
<td>system monitor</td>
<td>0.3</td>
<td>0.5</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td>application SW total</td>
<td>13.4</td>
<td>12.6</td>
<td>35.0</td>
<td>61.0</td>
</tr>
<tr>
<td>UNIX Solaris 2.x</td>
<td></td>
<td></td>
<td></td>
<td>10.0</td>
</tr>
<tr>
<td>file cache</td>
<td></td>
<td></td>
<td></td>
<td>3.0</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td></td>
<td></td>
<td>74.0</td>
</tr>
</tbody>
</table>
Alternative Power Visualization

- power supplies
- cooling
- UI and control
- paper path
- paper input module
- finisher
- paper
- procedé
- electrical power
- heat

Modeling and Analysis: Budgeting

version: 1.0
August 21, 2020
Evolution of Budget over Time

- fact finding through details
- aggregate to end-to-end performance
- search for appropriate abstraction level(s)

- from coarse guesstimate
- to reliable prediction

- from typical case
- to boundaries of requirement space

- from static understanding
- to dynamic understanding

- from steady state
- to initialization, state change and shut down

- from old system
- to prototype
- to actual implementation

---

\[ \text{time} \]

\[ \text{start} \quad \text{later} \quad \text{only if needed} \]
Potential Applications of Budget based design

- resource use (CPU, memory, disk, bus, network)
- timing (response, latency, start up, shutdown)
- productivity (throughput, reliability)
- Image Quality parameters (contrast, SNR, deformation, overlay, DOF)
- cost, space, time
What kind of budget is required?

<table>
<thead>
<tr>
<th>Budget Type</th>
<th>Base on</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>wish, empirical data, extrapolation, educated guess, or expectation</td>
</tr>
<tr>
<td>Typical case</td>
<td>dynamic</td>
</tr>
<tr>
<td>Global</td>
<td>worst case</td>
</tr>
<tr>
<td>Approximate</td>
<td>detailed</td>
</tr>
<tr>
<td></td>
<td>accurate</td>
</tr>
</tbody>
</table>

is the budget based on wish, empirical data, extrapolation, educated guess, or expectation?
Summary of Budgeting

A budget is a quantified instantiation of a model

A budget can prescribe or describe the contributions by parts of the solution to the system quality under consideration

A budget uses a decomposition in tens of elements

The numbers are based on historic data, user needs, first principles and measurements

Budgets are based on models and estimations

Budget visualization is critical for communication

Budgeting requires an incremental process

Many types of budgets can be made; start simple!
The Boderc project contributed to Budget Based Design. Especially the work of

Hennie Freriks, Peter van den Bosch (Océ),

Heico Sandee and Maurice Heemels (TU/e, ESI)

has been valuable.