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**

- **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
  ...

**life cycle context**

- creation
- life cycle business

**enterprise & users**
## 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

**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
  - compartmentalization
  - authentication
  - encryption

**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
- compartmentalization
- authentication
- encryption

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

- **Screen**
  - **Client**
  - **Network**
  - **Web Server**
  - **Database Server**
  - **Product Descriptions**
  - **Logistics ERP**
  - **Financial**
  - **Customer Relations**

**Response Time**
- Required server capacity
- Exhibit products
- Browse products

**Modeling and Analysis: System Model**

**Version:** 0.4

*September 9, 2018*

**MASMwebShopPictureCache**
zero order web server load model

Load = \( n_a \times t_a \)

\( n_a \) = total requests

\( t_a \) = 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

\[
n_{a,h} = n_a \cdot h
\]
\[
n_{a,m} = n_a \cdot (1-h)
\]

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

\[
\text{Load}(h) = n_a \cdot h \cdot t_h + n_a \cdot (1-h) \cdot t_m = n_a \cdot t_m - n_a \cdot h \cdot (t_m - t_h)
\]
quantified mid office server example

$\text{th} = 0.02 \text{ ms}$

$\text{tm} = 2 \text{ ms}$

$n_a = 1000$

$\text{Load}(h) = 1000 \times 2[\text{ms}] - 1000 \times h \times 1.98[\text{ms}]$

$\text{Load}(h) = 2000 - 1980 \times h [\text{ms}]$

hit rate

load in

seconds

utilizable capacity

working range

hit rate
Hit Rate Considerations

**quantified mid office server example**

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

\[ \text{Load}(h) = 1000 \times 2[\text{ms}] - 1000 \times h \times 1.98[\text{ms}] \]
\[ \text{Load}(h) = 2000 - 1980 \times h [\text{ms}] \]

Hit rate of well designed system is ample within working range (e.g. 95%) 

0\textsuperscript{th} order formula is valid:

\[ \text{Load} = 0.12 \times n_a [\text{ms}] \]

Hit rate is context dependent. 

*Life cycle changes or peak loads* may degrade hit rate.

Modeling and Analysis: System Model

version: 0.4

September 9, 2018

Gerrit Muller

MASMdiscussion
Response Time

human customer

press next

look

client

request picture

web server

check cache

request picture

store in cache

transfer to client

process

display

data base server

retrieve picture

time in milliseconds in optimal circumstances

t₀  t₀ + 10  t₀ + 20  t₀ + 30  t₀ + 40  t₀ + 50  t₀ + 60  t₀ + 70
What Memory Capacity is Required for Picture Transfers?

required memory capacity?
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

k

m

n

Modeling and Analysis: System Model
Gerrit Muller
version: 0.4
September 9, 2018
MASMmidOfficeProcesses
picture memory = 

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

where
- \( n \) = number of database access threads
- \( m \) = number of picture cache threads
- \( k \) = number of web server threads
- \( s \) = picture size in bytes
- \( c \) = in memory cache capacity in number of 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>
</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 main</td>
</tr>
<tr>
<td>highly concurrent</td>
<td>2</td>
<td>4</td>
<td>10</td>
<td>100</td>
<td>20</td>
<td>5.3</td>
<td>main disk</td>
</tr>
<tr>
<td>large pictures</td>
<td>2</td>
<td>4</td>
<td>1000</td>
<td>100</td>
<td>100</td>
<td>296</td>
<td>main+disk</td>
</tr>
<tr>
<td>many pictures</td>
<td>2</td>
<td>4</td>
<td>1000</td>
<td>100</td>
<td>100,000</td>
<td>9,540</td>
<td>main+disk</td>
</tr>
<tr>
<td>all at once</td>
<td>2</td>
<td>4</td>
<td>1000</td>
<td>100</td>
<td>100,000</td>
<td>98,234</td>
<td>disk</td>
</tr>
</tbody>
</table>

picture memory = 3 n s + 5 m s + c s + 3 k s

where
n = # back office access threads
m = # picture cache threads
k = # web server threads
s = picture size in bytes
c = in memory cache capacity in # pictures

memory use
product browsing only
pictures only
single server

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>
</tr>
</thead>
<tbody>
<tr>
<td>sales, order intake, payments</td>
<td></td>
</tr>
<tr>
<td>track, order handling</td>
<td></td>
</tr>
<tr>
<td>stock handling</td>
<td></td>
</tr>
<tr>
<td>financial bookkeeping</td>
<td></td>
</tr>
<tr>
<td>customer relation management</td>
<td></td>
</tr>
<tr>
<td>update catalogue</td>
<td></td>
</tr>
<tr>
<td>advertise</td>
<td></td>
</tr>
<tr>
<td>after sales support</td>
<td></td>
</tr>
</tbody>
</table>

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

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

aspect(d, f)

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

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)
...

creation

life cycle business

life cycle context

Modeling and Analysis: System Model

version: 0.4
September 9, 2018
MASSmfollowUp
## 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