Alexander Podelko

Oracle. Stamford, CT, USA

Performance Requirements
The Backbone of the Performance Engineering Process

Introduction

- The topic is more complicated than it looks
- Performance requirements are supposed to be tracked through the whole system lifecycle
- Each group of stakeholders has its own view and terminology
- An overview of existing issues and an attempt to create a holistic view

1

SDLC

Performance Eng Life Cycle

SDLC	Performance Eng Life Cycle
• Requirements	• Performance Requirements
• Architecture and Design	• Design for Performance and Performance Modeling
• Construction / Implementation	• Unit Performance Tests and Code Optimization
• Testing	• Performance Testing
• Deployment and Maintenance	• Performance Monitoring and Capacity Management

Agenda

- <u>Metrics</u>
- Elicitation
- Analysis and Specification
- Validation and Verification

High-Level View of System

```
┌─────────────────┐        ┌───────────────────┐
│                 │  ●→     │   ┌───────────┐   │
│   Users    ●←   │  ●→     │   │ Software  │   │
│            ●←   │  ●→     │   └───────────┘   │
│            ●←   │  ●→     │         ↕         │
│            ●←   │  ●→     │   ┌───────────┐   │
│            ●←   │  ●→     │   │ Hardware  │   │
│            ●←   │         │   └───────────┘   │
└─────────────────┘        └───────────────────┘
```
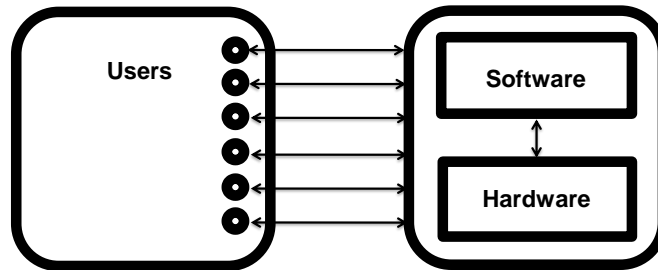
Business Performance Requirements

- For today's distributed business systems
- Throughput
- Response / processing times
- All are important

3

Throughput

- The rate at which incoming requests are completed
  - Usually we are interested in a steady mode
- Straightforward for homogeneous workloads
  - Not so easy for mixed workloads: mix ratio can change with time
- Varies with time
  - Typical hour, peak hour, average, etc.

Number of Users

- Number of users by itself doesn't define throughput
  - Without defining what each user is doing and how intensely
  - 500 users running one short query each minute: throughput 30,000 queries per hour
  - 500 users running one short query each hour: throughput 500 queries per hour
  - Same 500 users, 60X difference between loads

Concurrency

- Number of simultaneous users or threads
  - Number of active users
- Take resources even if doing nothing
- Number of named users
  - Rather a data-related metric
- Number of "really concurrent" users
  - Number of requests in the system
  - Not a end-user performance metric

Response Times

- How fast requests are processed
- Depends on context
  - 30 minutes may be excellent for a large batch job
- Depends on workload
  - Conditions should be defined
- Aggregate metrics usually used
  - Average, percentiles, etc.

Context

- All performance metrics depend on context like:
  - Volume of data
  - Hardware resources provided
  - Functionality included in the system
    - Functionality is added gradually in agile methodologies

Internal (Technological) Requirements

- Important for IT
- Derived from business and usability requirements
  - During design and development
- Resources
- Scalability

Resources

- CPU, I/O, memory, and network
- Resource Utilization
  - Related to a particular configuration
  - Often generic policies like CPU below 70%
- Relative values (in percents) are not useful if configuration is not given
  - Commercial Off-the-Shelf (COTS) software
  - Virtual environments

Resources: Absolute Values

- Absolute values
  - # of instructions, I/O per transaction
    - Seen mainly in modeling
  - MIPS in mainframe world
- Importance increases again with the trends of virtualization, cloud computing, and SOA
  - VMware: CPU usage in MHz
  - Microsoft: Megacycles
  - Amazon: EC2 Compute Units (ECU)

Scalability

- Ability of the system to meet performance requirements as the demand increases

- Increasing # of users, transaction volumes, data sizes, new workloads, etc.

- Performance requirements as a function, for example, of load or data and configuration
  - No free / ideal scalability

Agenda

- Metrics
- Elicitation
- Analysis and Specification
- Validation and Verification

IEEE SWEBOK

- IEEE Software Engineering Book of Knowledge defines four stages for requirements:
  - Elicitation
    - Where come from and how to collect them
  - Analysis
    - Classify / Elaborate / Negotiate
  - Specification
    - Production of a document
  - Validation

Where do performance requirements come from?

- Business

- Usability

- Technology

Business Requirements

- Comes from the business, may be caught before design starts

    - Number of orders per hour

- The main trap is to immediately link them to a specific design and technology thus limiting the number of available choices

    - For example, it may be one page per order or a sequence of two dozen screens

    - Each of the two dozen may be saved separately or all at the end

Requirements Elicitation

- *Final* requirements should be quantitative and measurable

- Business people know what the system should do and may provide some information

    - They are not performance experts

- Document real business requirements in the form they are available

    - Then elaborate them into quantitative and measurable

Goals vs. Requirements

- Most response times "requirements" are goals

  - Missing them won't prevent deploying the system

- For response times, the difference between goals and requirements may be large

  - For many web applications goals are two-five seconds and requirements somewhere between eight seconds and one minute

Determining Specific Requirements

- It depends
- Approach the subject from different points of view
- Just to illustrate here are 10 methods suggested by Peter Sevcik to find T in APDEX
  - T is threshold between satisfied and tolerating users; should be strongly correlated with the response time goal

Methods to Find T
(by Peter Sevcik)

- Default value (4 sec)
- Empirical data
- User behavior model (# of elements/task repetitiveness)
- Outside references
- Observing users
- Controlled perf experiment
- Best time multiple
- Calculate from frustration threshold F (F=4T in APDEX)
- Interview stakeholders
- Mathematical inflection point

Suggested Approach

- Peter Sevcik suggests to use several of these methods: if all come approximately to the same number it will be T

- A similar approach can be used for performance requirements: use several methods to get the numbers – you get goal/requirement if they are close
  - Investigate / sort out if they differ significantly

Usability Requirements

- Many researchers agree that
  - Users lose focus if response times are more than 8 to 10 seconds
  - Making response times faster than one to two seconds doesn't help productivity much
- Sometimes linked closely to business requirements
  - Make sure that response times are not worse than competitor's

Response Times: Review of Research

- In 1968 Robert Miller defined three threshold levels of human attention
- Instantaneous 0.1-0.2 seconds
- Free interaction 1-5 seconds
- Focus on dialog 5-10 seconds

Instantaneous Response Time

- Users feel that they directly manipulate User Interface (UI)
- For example, between typing a symbol and its appearance on the screen
- 0.1-0.2 seconds
- Often beyond the reach of application developers
  - System/UI libraries, client-side

Free Interaction

- Notice delay, but "feel" the computer is "working"
- Earlier researchers reported 1-2 sec
  - Simple terminal interface
- For problem solving tasks no performance degradation up to 5 sec
  - Depends on the number of elements and repetitiveness of the task

Does It Change with Time?

- 2009 Forrester research suggests 2 second response time, in 2006 similar research suggested 4 seconds

  - The approach is often questioned: they just ask. It is known that user perception of time may be misleading

  - What page are we talking about?

Web Performance Insights

- Tammy Everts' book 'Time is Money: The Business Value of Web Performance'

- WPO Stats https://wpostats.com/

- HTTP Archive https://httparchive.org/

Business Web Metrics

- bounce rate
- cart size
- conversions
- revenue
- time on site
- page views
- user satisfaction

- user retention
- organic search traffic
- brand perception
- productivity
- bandwidth/CDN savings

Focus on Dialog

- Users are focused on the task: 5-10 sec

- Half of users abandon Web pages after 8.5 sec - Peter Bickford, 1997

    - 2 min delay after 27 quick interactions
    - Watch cursor kept users 20 sec, animated cursor 1 min, progress bar until the end

- Users should reorient themselves after a delay above the threshold

Agenda

- Metrics
- Elicitation
- <u>Analysis and Specification</u>
- Validation and Verification

Technological Requirements

- Comes from the chosen design and used technology
  - We call ten web services sequentially to show a page within 3 sec. It translates into requirements of 200-250 ms for each web service
  - Resource utilization requirements

Analysis and Modeling

- Final requirements are elaborated from business requirements by applying usability and technological requirements

- Requirements traceability
  - Where it came from

- Input for Software Performance Engineering
  - For example, defining service / stored procedure response times by its share in the end-to-end performance budget

Documenting Requirements

- Requirements / Architect's vocabulary

- Quality Attributes
  - Part of Nonfunctional Requirements

- Approaches
  - Text
  - Quality Attribute Scenarios (SEI)
  - Planguage

Quality Attribute Scenarios

- QA scenario defines:
  - Source   [users]
  - Stimulus   [workload]
  - Environment   [system state]
  - Artifact   [part of the system]
  - Response   [expected actions]
  - Response Measure   [response times]

Planguage

- Tag: unique identifier
- Gist: brief description
- Scale: unit of measure
- Meter: how to measure
- Minimum / Plan / Stretch/ Wish : levels to attain
- Past / Record / Trend

What Metrics to Use?

- Average
- Max
- Percentiles (X% below Y sec)
- Median
- Typical
- etc.

The Issue

- SLA (Service Level Agreement)
  - "99.5% of all transactions should have a response time less than five seconds"
- What happens with the rest 0.5%?
  - All 6-7 seconds
  - All failed/timeout
- Add different types of transactions, different input data, different user locations, etc.

Observability

- Four different viewpoints
  - Management
  - Engineering
  - QA Testing
  - Operations
- Ideal would be different views of the same performance database
- Reality is a mess of disjoint tools

May Need a Few Metrics

- Combination of percentile and availability metric works in many cases
  - 97% below 5 sec, less than 1% failed/timeout
- An example of another metric:
  - Apdex (Application Performance Index)
  - Objective user satisfaction metric
  - A number between 0 and 1
  - 0 no users satisfied, 1 all users satisfied

Agenda

- Metrics
- Elicitation
- Analysis and Specification
- <u>Validation and Verification</u>

Requirements Validation

- Making sure that requirements are valid
  - Quite often used to mean checking against test results (instead of verification)
- Checking against different sources
- Reviews, modeling, prototyping, etc.
- Iterative process
- Tracing
  - Tracing back to the original requirement

Requirements Verification

- Checking if the system performs according to the requirements

- Both requirements and results should use the same aggregates to be compared

- Many tools measure only server time (or server and network)
  - End user time may differ significantly, especially for rich web clients or thick clients

- Both in load testing and production !

Verification Issue

- Let's consider the following example

- Response time requirement is 99% below 5 sec

- 99% 3-5 sec, 1% 5-8 sec
  - Looks like a minor performance issue

- 99% 3-5 sec, 1% failed or had strangely high response times (more than 30 sec)
  - Looks like a bug or serious performance issue

Requirements Verification: Performance vs. Bug

- Two completely different cases
  - Performance issue: business decision, cost vs. response time trade off.
  - Bug exposed under load: should be traced down first to make decision

The equipment is not operating as expected, and therefore there is a danger that it can operate with even wider deviation in this unexpected and not thoroughly understood way. The fact that this danger did not lead to a catastrophe before is no guarantee that it will not the next time, unless it is completely understood.

Dr. Richard Feynman
Roger Commission Report on the
Challenger space shuttle accident

Summary

- Specify performance requirements at the beginning of any project
- What to specify depends on the system
  - Quantitative and measurable in the end
- Elaborate and verify requirements throughout Development – Testing – Production

Questions ?

Alexander Podelko

apodelko@yahoo.com
@apodelko