CMG'12

Performance Requirements: the Backbone of the Performance Engineering Process

Paper 1102

Session 601

Alexander Podelko

alex.podelko@oracle.com

@apodelko

December 7, 2012

1

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

2

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

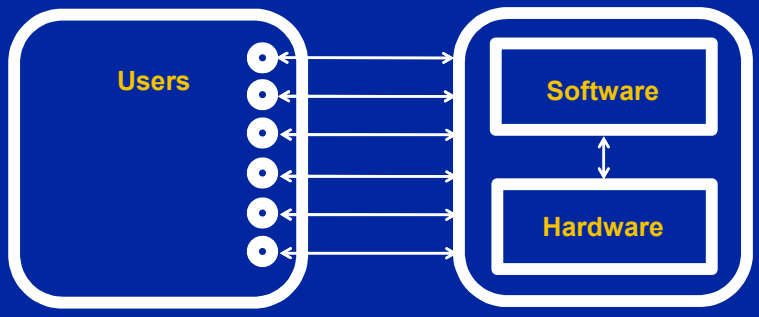
3

Agenda

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

4

High-Level View of System



5

Business Performance Requirements

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

6

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.

7

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

8

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

9

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.

10

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

11

Internal (Technological) Requirements

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

12

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

13

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

14

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

15

Agenda

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

16

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

17

Where do performance requirements come from?

- Business
- Usability
- Technology

18

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

19

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

20

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

21

See The Whole Picture

- For example, the requirement is 10 seconds
- We got 15 seconds for peak load
- But what if
  - Only on busiest day of the year
  - All other days it will be below 10 seconds
  - It is CPU-constrained and may be fixed by additional hardware

22

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

23

Methods 1-5 to Find T (by Peter Sevcik)

- Default value (4 sec)
- Empirical data
- User behavior model (# of elements/task repetitiveness)
- Outside references
- Observing users

24

Methods 6-10 to Find T (by Peter Sevcik)

- Controlled performance experiment
- Best time multiple
- Find frustration threshold F first and calculate T from F (F=4T in APDEX)
- Interview stakeholders
- Mathematical inflection point

25

Suggested Approach

- So 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

26

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

27

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

28

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

29

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

30

Does It Change with Time?

- Do expectations increase 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?

31

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

32

Agenda

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

33

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

34

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

35

Documenting Requirements

- Requirements / Architect's vocabulary
- Quality Attributes
  - Part of Nonfunctional Requirements
- Approaches
  - Text
  - Quality Attribute Scenarios (SEI)
  - Planguage

36

Quality Attribute Scenarios

- QA scenario defines:
  - Source
  - Stimulus
  - Environment
  - Artifact
  - Response
  - Response Measure

37

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

38

What Metrics to Use?

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

39

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.

40

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

41

Metrics to Use

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

42

Agenda

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

43

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

44

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 !

45

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

46

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

47

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

48

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

49

Questions ?

Alexander Podelko

alex.podelko@oracle.com
@apodelko

Links and references may be found in the paper and at www.alexanderpodelko.com

50