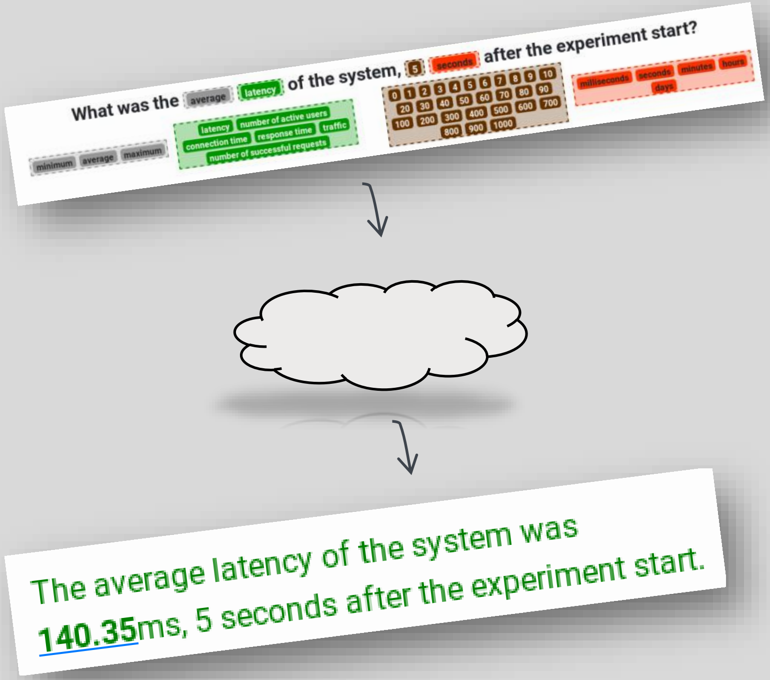


What was the average latency of the system, 5 seconds after the experiment start?
minimum average maximum
latency number of active users connection time response time traffic number of successful requests
0 1 2 3 4 5 6 7 8 9 10 20 30 40 50 60 70 80 90 100 200 300 400 500 600 700 800 900 1000
milliseconds seconds minutes hours days
The average latency of the system was 140.35ms, 5 seconds after the experiment start.

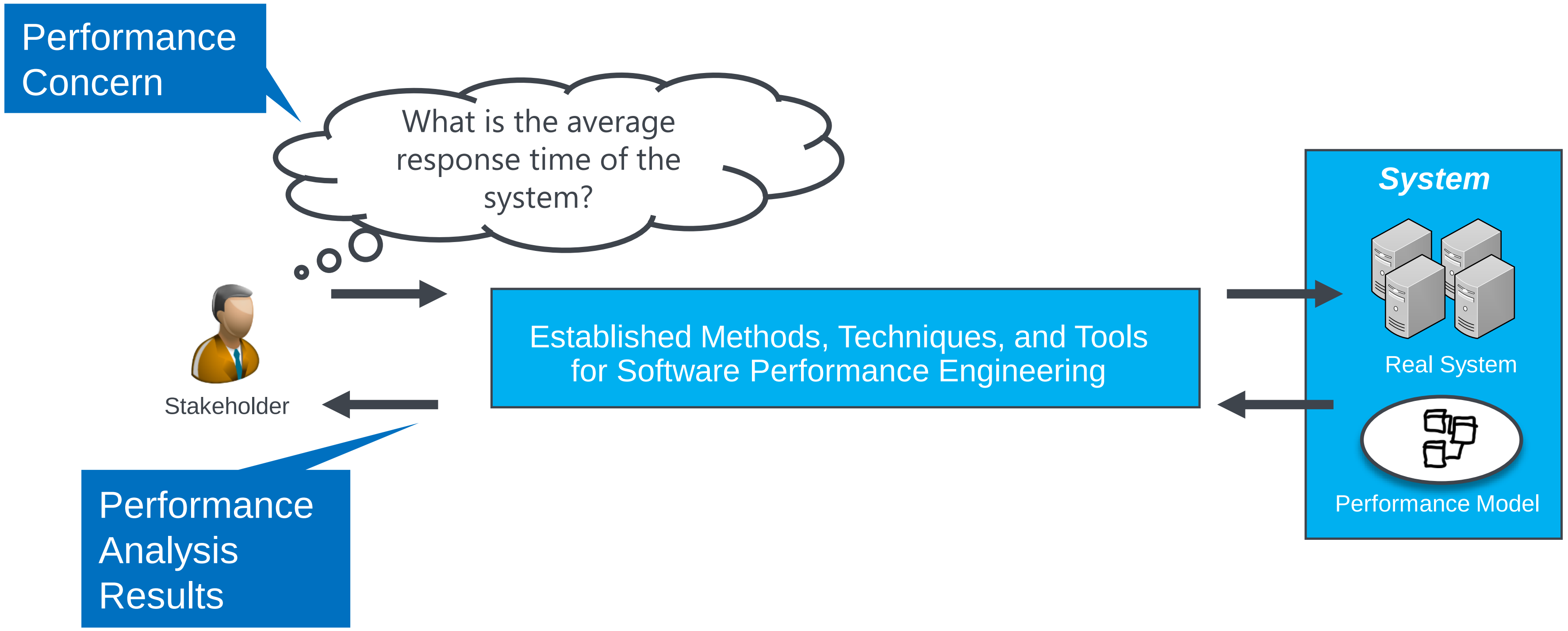
Work-in-progress/vision

Concern-driven Reporting of Software Performance Analysis Results

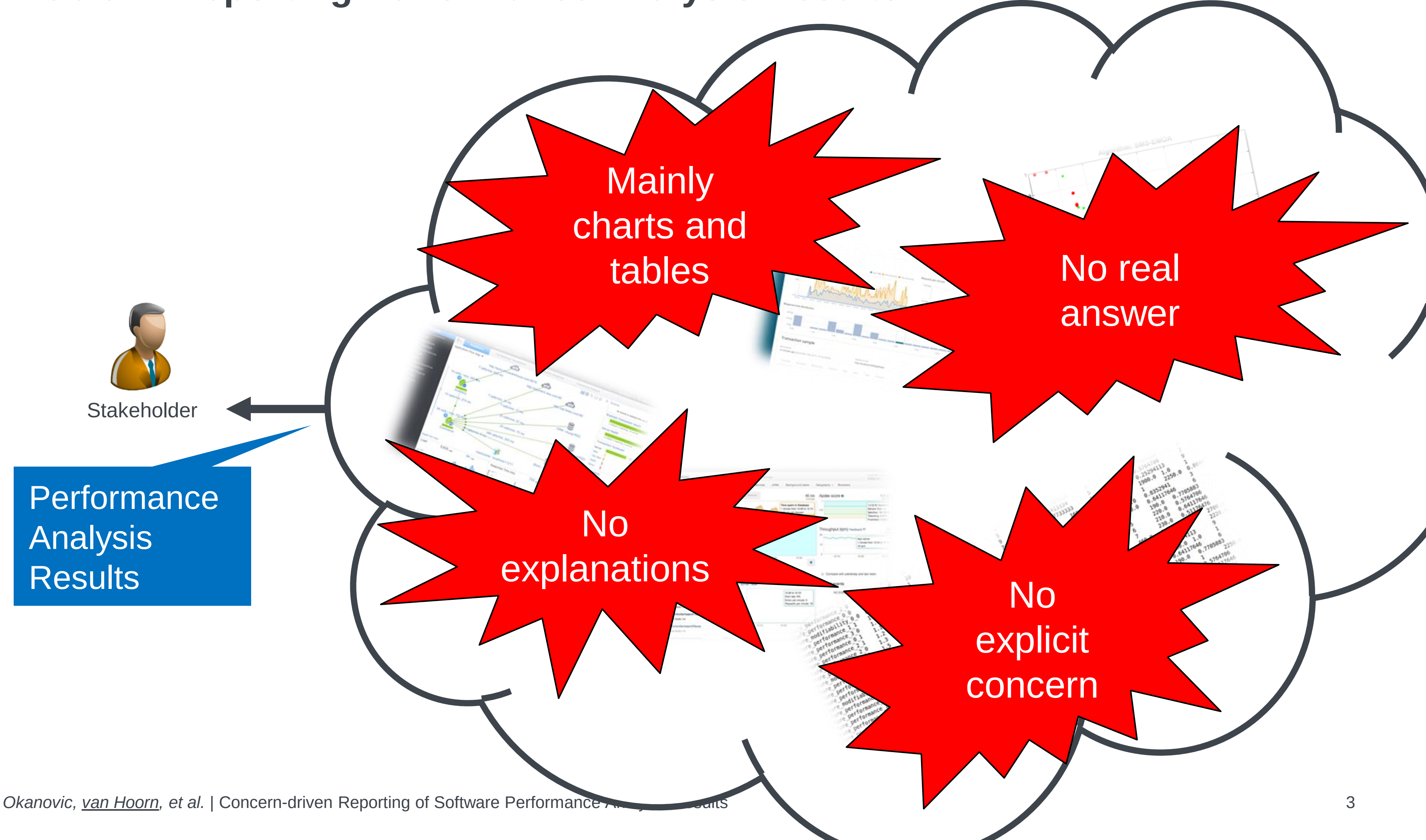
Dušan Okanović,
André van Hoorn @andrevanhoorn
Christoph Zorn, Fabian Beck,
Vincenzo Ferme, Jürgen Walter

10th ACM/SPEC Int. Conf. on Performance Engineering (ICPE 2019)
Mumbai, India – April 10, 2019

Background: Performance Analysis Workflow

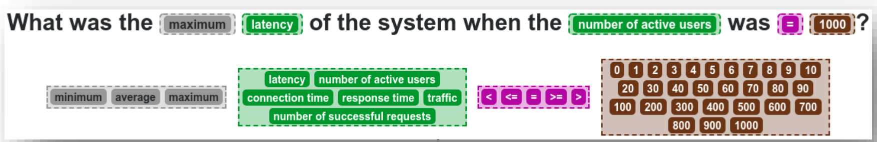

Performance Concern
What is the average response time of the system?
Stakeholder
Established Methods, Techniques, and Tools for Software Performance Engineering
System
Real System
Performance Model
Performance Analysis Results

Problem: Reporting Performance Analysis Results



Approach and Tool (PoC) – Concern-driven Reporting

VIZARD tool



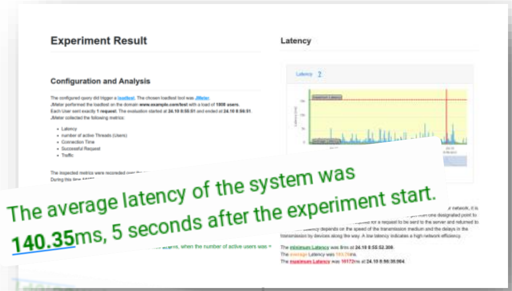
What was the maximum latency of the system when the number of active users was = 1000?

minimum average maximum

latency number of active users connection time response time traffic number of successful requests

< <= = >= >

0 1 2 3 4 5 6 7 8 9 10 20 30 40 50 60 70 80 90 100 200 300 400 500 600 700 800 900 1000

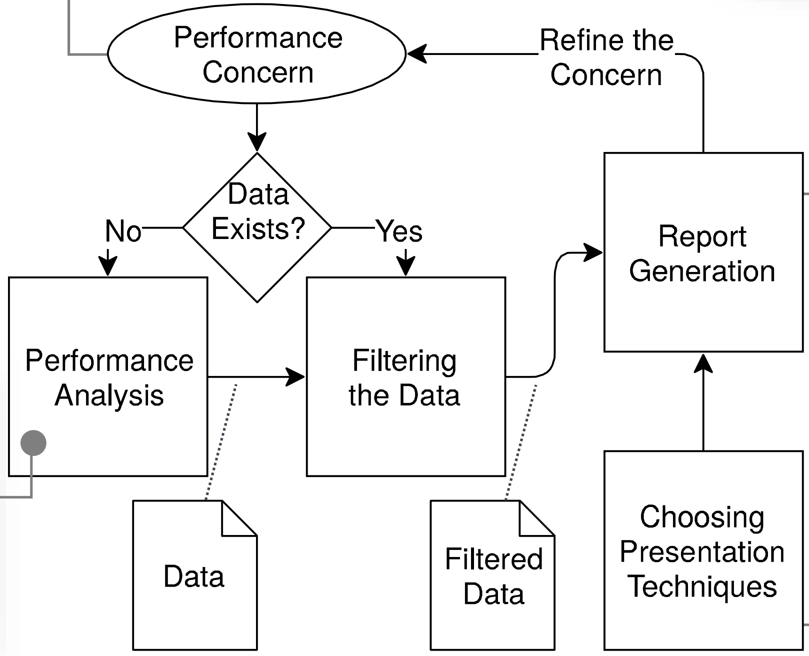


Experiment Result

Latency

Configuration and Analysis

The average latency of the system was 140.35ms, 5 seconds after the experiment start.



Declaro

Asking *"What?"*, automating the *"How?"!*

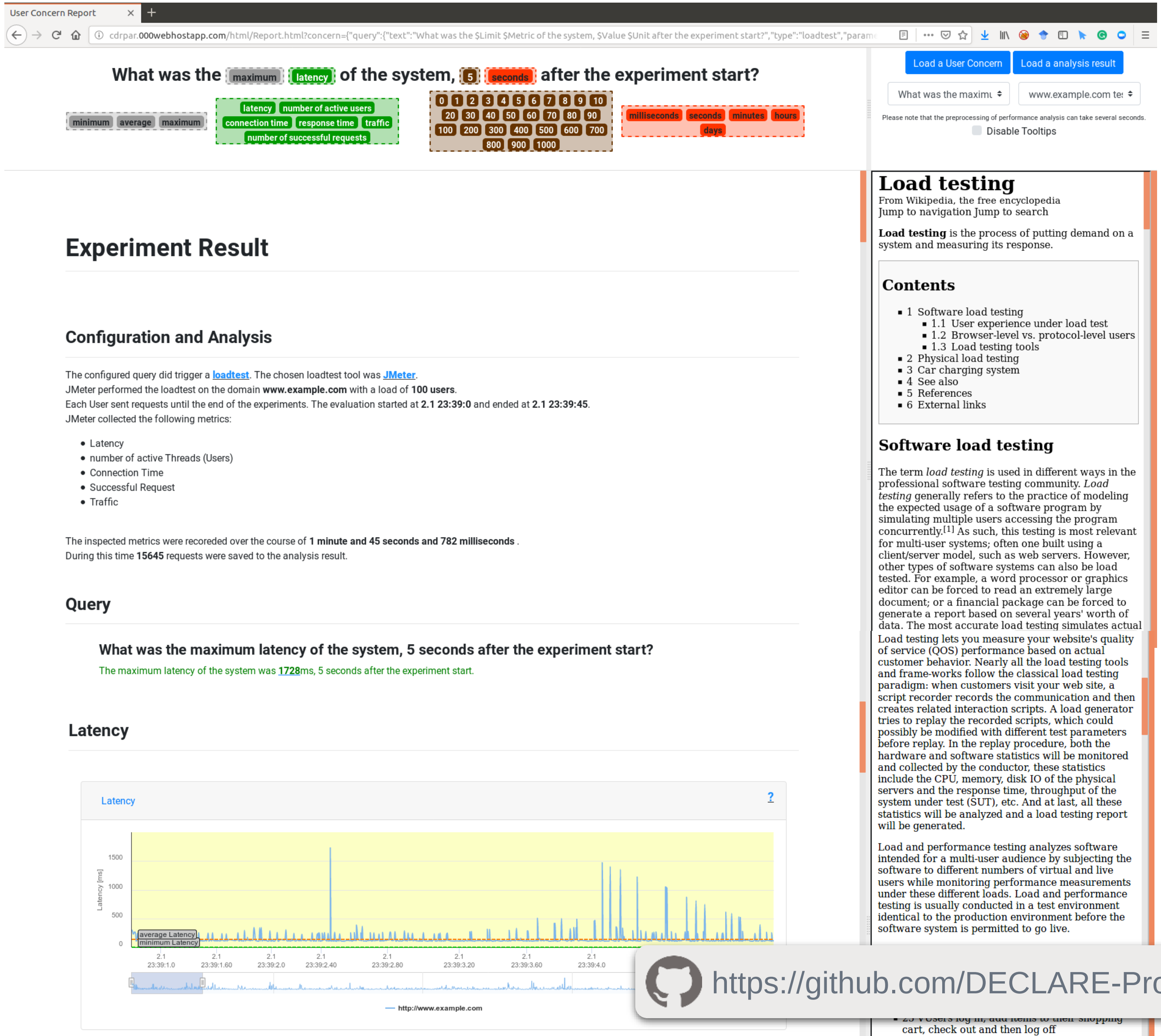
Walter et al., ICPE 2016

"State of the Art of Performance Visualization"

Isaac et al., EuroVis 2014

Demo

Concern-driven VIZARD Report


User Concern Report
cdrpar.000webhostapp.com/html/Report.html?concern={"query":{"text":"What was the $Limit $Metric of the system, $Value $Unit after the experiment start?","type":"loadtest","param
Load a User Concern
Load a analysis result
What was the maximu
www.example.com te
Disable Tooltips

Please note that the preprocessing of performance analysis can take several seconds.

What was the maximum latency **of the system,** 5 seconds **after the experiment start?**

Experiment Result

Configuration and Analysis

The configured query did trigger a loadtest. The chosen loadtest tool was JMeter.
JMeter performed the loadtest on the domain **www.example.com** with a load of **100 users**.
Each User sent requests until the end of the experiments. The evaluation started at **2.1 23:39:0** and ended at **2.1 23:39:45**.
JMeter collected the following metrics:

- Latency
- number of active Threads (Users)
- Connection Time
- Successful Request
- Traffic

The inspected metrics were recoreded over the course of **1 minute and 45 seconds and 782 milliseconds** .
During this time **15645** requests were saved to the analysis result.

Query

What was the maximum latency of the system, 5 seconds after the experiment start?

The maximum latency of the system was 1728ms, 5 seconds after the experiment start.

Latency

Load testing

From Wikipedia, the free encyclopedia
Jump to navigation Jump to search

Load testing is the process of putting demand on a system and measuring its response.

Contents

Software load testing

The term *load testing* is used in different ways in the professional software testing community. *Load testing* generally refers to the practice of modeling the expected usage of a software program by simulating multiple users accessing the program concurrently.[1] As such, this testing is most relevant for multi-user systems; often one built using a client/server model, such as web servers. However, other types of software systems can also be load tested. For example, a word processor or graphics editor can be forced to read an extremely large document; or a financial package can be forced to generate a report based on several years' worth of data. The most accurate load testing simulates actual

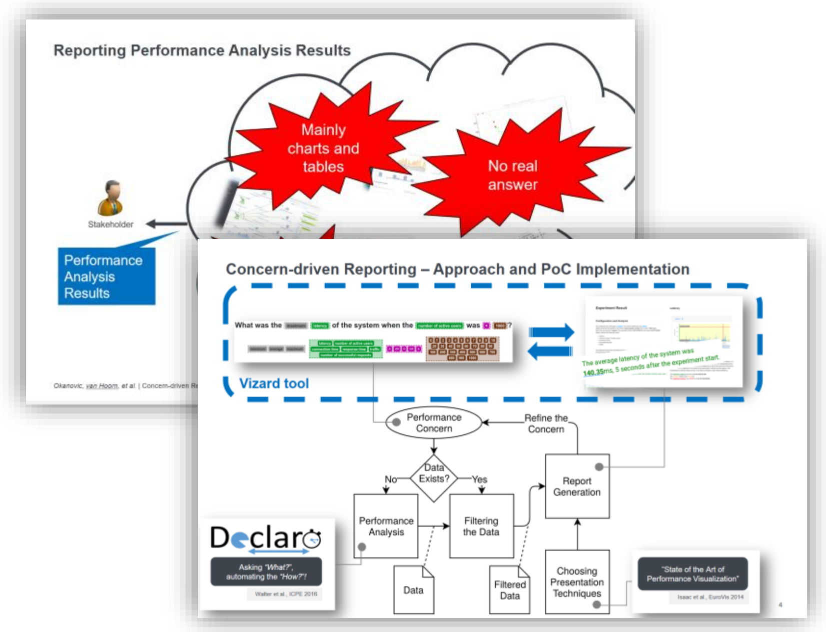
Load testing lets you measure your website's quality of service (QOS) performance based on actual customer behavior. Nearly all the load testing tools and frame-works follow the classical load testing paradigm: when customers visit your web site, a script recorder records the communication and then creates related interaction scripts. A load generator tries to replay the recorded scripts, which could possibly be modified with different test parameters before replay. In the replay procedure, both the hardware and software statistics will be monitored and collected by the conductor, these statistics include the CPU, memory, disk IO of the physical servers and the response time, throughput of the system under test (SUT), etc. And at last, all these statistics will be analyzed and a load testing report will be generated.

Load and performance testing analyzes software intended for a multi-user audience by subjecting the software to different numbers of virtual and live users while monitoring performance measurements under these different loads. Load and performance testing is usually conducted in a test environment identical to the production environment before the software system is permitted to go live.

- 25 VUsers log in, add items to their shopping cart, check out and then log off

https://github.com/DECLARE-Project/Vizard

Summary


Reporting Performance Analysis Results
Mainly charts and tables
No real answer
Stakeholder
Performance Analysis Results
Concern-driven Reporting – Approach and PoC Implementation
Vizard tool
Performance Concern
Refine the Concern
Data Exists?
No
Yes
Performance Analysis
Filtering the Data
Report Generation
Data
Filtered Data
Choosing Presentation Techniques
Declare
Asking "What?", automating the "How?"
"State of the Art of Performance Visualization"

https://github.com/DECLARE-Project/Vizard

Future Work

- Concern specification
  - Mappping to other languages (DQL, behavior-driven, …)
  - Support for additional concerns
- Reporting
  - Interactive analysis, e.g., chat bots
  - Other types of reports, e.g., videos, VR
- Vizard Tool
  - Integration with DPE tooling
- Evaluation
  - User study
  - Expert review

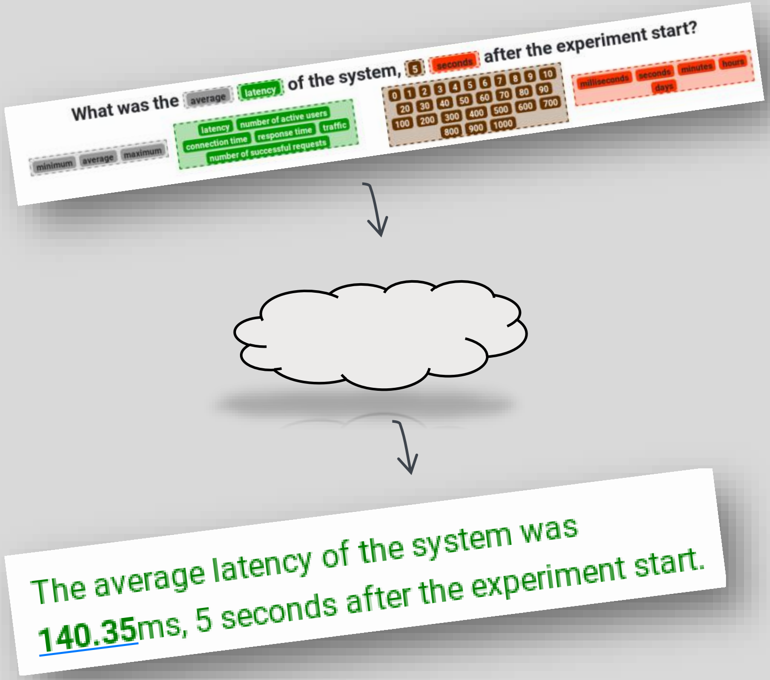
We have conducted a preliminary pilot study

Does Vizard help experts and/or (non-experts)?

References

Isaac et al., EuroVis 2014 — Katherine E. Isaacs, Alfredo Giménez, Ilir Jusufi, Todd Gamblin, Abhinav Bhatele, Martin Schulz, Bernd Hamann, Peer-Timo Bremer: *State of the Art of Performance Visualization.* EuroVis (STARs) 2014

Walter et al., ICPE 2016 — Jürgen Walter, André van Hoorn, Heiko Koziolek, Dusan Okanovic, Samuel Kounev: *Asking "What"?, Automating the "How"?: The Vision of Declarative Performance Engineering.* ICPE 2016: 91-94


What was the average latency of the system, 5 seconds after the experiment start?

The average latency of the system was 140.35ms, 5 seconds after the experiment start.

Work-in-progress/vision

Concern-driven Reporting of Software Performance Analysis Results

Dušan Okanović,
André van Hoorn @andrevanhoorn
Christoph Zorn, Fabian Beck,
Vincenzo Ferme, Jürgen Walter

10th ACM/SPEC Int. Conf. on Performance Engineering (ICPE 2019)
Mumbai, India – April 10, 2019