

IMPACT 2021

JANUARY 2021 VIRTUAL CONFERENCE EVENT

THE PAST, PRESENT, AND FUTURE OF PERFORMANCE ENGINEERING



1

About the Speaker



- Alexander Podelko
apodelko@yahoo.com
@apodelko
<https://alexanderpodelko.com/blog/>
- Specializing in performance since 1997
- Currently Staff Performance Engineer at MongoDB, performance testing and optimization of the MongoDB server
- Board director at Computer Measurement Group (CMG) – a non-profit organization of performance and capacity professionals

Disclaimer: The views expressed here are my personal views only and do not necessarily represent those of my current or previous employers. All brands and trademarks mentioned are the property of their owners.



2

Agenda

A Short History of Performance Engineering

The Current State of Performance Engineering

What Does the Future Hold?



3

Computing Paradigms

Looks like a new wave of performance engineering starts 10+ years after new computing paradigm

Mainframes

- Late 50s

Distributed Systems [Decentralization]

- Late 70s

Web, Mobile (as a client), Cloud [Centralization]

- Late 90s

? - Mobile (as a device), Peer-to-Peer, IoT, Fog, Edge [Decentralization]

- Late 10s



4

1966: Instrumentation

1966 - SMF (System Management Facilities)
released as part of OS/360

- Still in use

Observability
IT Operations
Analytics
Big Data
Deep Diagnostics



5

1968: Response Times

1968 - Robert Miller (IBM) in his "Response Time in Man-Computer
Conversational Transactions" paper described several threshold levels of
human attention

- Widely cited by many later researchers



6

1974: Monitoring

1974 – RMF (Resource Measurement Facility) released as part of MVS

- Still in use

1975 – OMEGAMON for MVS by Candle

- Claimed to be the first real-time monitor
- Acquired by IBM in 2004



7

1975: Community

1975 – The Computer Measurement Group (CMG) was created, holding annual conferences since



Computer Measurement Group

<http://cmg.org>



8

1977: Performance Analysis Tool

1977 – BEST/1 was released by BGS Systems, capacity and performance management tool

- the first commercial package for computer performance analysis to be based on analytic models.

*BGS Systems was acquired
by BMC Software in 1998*

```
Work with BEST/1 Model
Performance data . . . : BESTDATA (Q930400309)
Model/Text . . . . . : BEST1MODEL

Select one of the following:

  1. Work with workloads
  2. Specify objectives and active jobs

  5. Analyze current model
  6. Analyze current model and give recommendations
  7. Specify workload growth and analyze model

 10. Configuration menu
 11. Work with results

More ...
```

9

Late 70s - Distributed Systems

Distributed Systems

- Client/Server, Three-Tier, N-Tier, etc.

Less control, less instrumentation

More need for performance design and testing



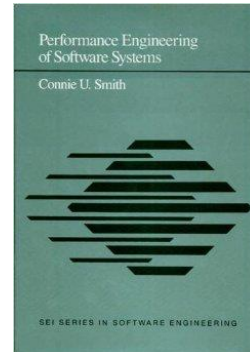
10

1990: Software Performance Engineering

1990 – Dr. Connie Smith's book Performance Engineering of Software Systems was published

- 15 pages of bibliography

☞ Fix-it-later was once a viable approach, but it is now inappropriate and dangerous



11

Does It Remind You Anything?

☞ Fix-it-later was a viable approach in the 1970s, but today the original promises no longer hold, and fix-it-later is archaic and dangerous. The original premises were:

- Performance problems are rare.
- Hardware is fast and inexpensive.
- It's too expensive to build responsive software.
- You can tune software later, if necessary.

Performance Engineering of Software Systems, 1990

12

1991: Load Testing Tools

1991 – LoadRunner was released by Mercury Interactive

- Acquired by HP in 2006 for \$4.5B

Performance expertise moves to performance testing teams



Application Performance Management (APM)

The term was coined by Programart Corp. (acquired by Compuware in 1999) in 1992 (in the mainframe context, as a combination of their STROBE and APMpower tools)

- STROBE had been on the market since the '70s

Introscope by Wily Technology, founded by Lew Cirne in 1998 (acquired by CA in 2006)

If supply can't go up, demand must come down. And the key to decreasing demand is improving your applications' performance.

Now there's a way to do just that. Application Performance Management.

With an APM strategy, IS departments in MVS mainframe environments can manage performance and cost throughout an application's life cycle. As a result, applications are deployed on time and provide high performance from Day One. What's more, APM proves truly invaluable in the long term, because it helps you keep performance high even when business requirements,

workload and technology change.

The bottom line? You save money, satisfy customers and give your company a competitive edge.

Who knows, you might even bring a smile to your Directors' faces.

How do you get Application Performance Management working for your IS department? Through STROBE and APMPOWER from Programart. For a free demonstration of just how much you can save, talk to us today at 617-498-4005.

It's got to be better than talking to the Board.

PROGRAMART
CORPORATION

APM, STROBE, and Programart are trademarks or service marks of Programart Corporation. ©1998 Programart Corporation

1998: End-User Monitoring

1998 – ETEWatch released by Candle

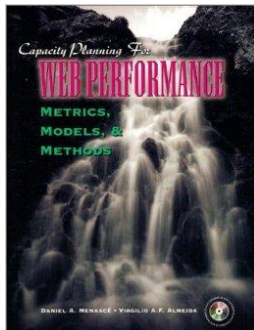
- Application response time monitor
- End-to-End Watch
- Candle acquired by IBM in 2004, then a part of Tivoli



15

Mid 90s – Web

Most of existing expertise was still applicable to back end
Books applying existing expertise to Web (1998)



16

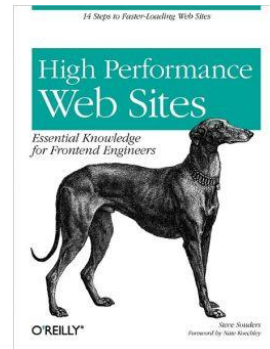
2007: Web Performance Optimization

2007 – Steve Souders' book High Performance Web Sites: Essential Knowledge for Front-End Engineers

New Web-specific expertise

2008 – WPO Community

- First Velocity conference
- WebPerf meetups



17

2006: Cloud Computing

References to "cloud computing" appeared as early as 1996, with the first known mention in a Compaq internal document

Cloud computing was popularized with Amazon releasing its Elastic Compute Cloud product in 2006

Third-party autoscaling software Scalr and RightScale appeared in 2008. Amazon launched its own autoscaling feature in 2009.



18

Agenda

A Short History of Performance Engineering

The Current State of Performance Engineering

What Does the Future Hold?



19

What Changed?

Web

- Centralization [back to mainframes ?]
- Open / unlimited workload

Cloud

- Further centralization [chargeback - back to mainframes ?]
- Dynamic Configurations

Agile / iterative development

- Continuous Integration / Delivery / Deployment

Mobile / WPO / Rich Internet Client

- Shift to single-user client performance optimization



20

Performance Risk Mitigation

Single-user performance engineering

- Profiling, WPO, single-user performance

Software Performance Engineering

- Modeling, Performance Patterns

Instrumentation / APM / Monitoring

- Production system insights

Capacity Planning/Management

- Resources Allocation

Continuous Integration / Deployment

- Ability to deploy and remove changes quickly

Three Groups:

- Analytical
- Experimental
- Observational



21

Load Testing Role

*All of them don't
replace load testing:*

*As load testing remains
the only proactive
experimental way to
mitigate multi-user
performance risks!*



22

What Load Testing Adds

Verifying that the system can handle peak loads

Verifying multi-user performance


- As single-user improvement may lead to multi-user degradation

Performance optimization

- Apply exactly the same load
- See if the change makes a difference

Debugging/verification of multi-user issues

Testing self-regulation functionality

- Such as auto-scaling or changing the level of service depending on load
- 

23

Changing Dynamic / Historical View


Mainframes

- Methods: Instrumentation, Scheduling, Capacity Planning
- Titles: Performance Analyst, Capacity Planners

Distributed Systems

- Load Testing, System Monitoring
- Titles: Performance Tester [Engineer, Architect]

Web / Cloud

- Methods: App Monitoring, Automation, Observability
 - Titles: Performance Engineer, SRE
- 

24

Five Trends of Performance Engineering

- Adjusting to Agile and CI/CD
- Integrating performance engineering into DevOps
- Context-driven performance engineering
- Integrating everything: tools, processes, roles
- Chaos engineering: renaissance of reliability

*Originally published in State of Performance
Engineering Report 2020*



25

Adjusting to Agile and CI/CD

Agile development should be rather a trivial case for performance testing

- You have a working system each iteration to test early by definition.
- You need performance engineer for the whole project
 - Savings come from detecting problems early

Performance Engineering teams don't scale well

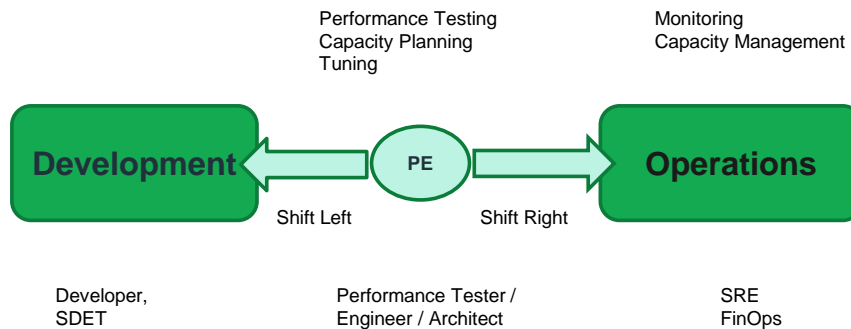
- Even assuming that they are competent and effective
- Increased volume exposes the problem
 - Early testing
 - Each iteration

Remedies: automation, making performance everyone's job



26

Integrating Performance Engineering into DevOps



Expand or Move Out ?

27

Context-Driven Performance Engineering

Select a combination of performance mitigation methods optimal for *your* context

- Software Performance Engineering (SPE)
- Single-user performance engineering
- Performance Testing
- Instrumentation / APM / Monitoring / Observability
- Capacity Planning/Management
- Continuous Integration / Deployment

A holistic approach to performance

- Get all these methods complement each other and work together

28

Integrating Everything: Tools, Processes, Roles


Tools get integrated across functionality and areas of application. No name / concept of the integrated tool yet.

- APM (Application Performance Monitoring / Management)
- ITOM (IT Operations Management)
- ITOA (IT Operations Analytics)
- AIOps (Artificial Intelligence for / Algorithmic IT Operations)
- Digital Operations Management
- DEM (Digital Experience Management)
- Service Mesh ...

Processes

- Integration into CI/CD and DevOps

Roles

- Looks like silo performance roles are getting extinct
- 

29

Chaos Engineering: Renaissance of Reliability

Reincarnation of reliability testing

- Which was practically ignored for a long time
- Started in production
 - Slowly moving back to testing
- Basically reliability testing tools
 - Not touched by load testing tool vendors as far as I know (except Network simulation)

Developing into a separate discipline – *Resilience Engineering*



30

Pack Mentality

Companies with non-critical business set fashions

- Google, Facebook, Netflix, etc.
- Actively promote their processes (SRE, etc.)
- Could tolerate temporary inconsistencies and infrequent user issues
 - Canary, A/B testing may work well
- Focus on handling failures, not preventing them
- Things they almost never mention:
 - Performance testing
 - Performance modeling



31

Tools Downwards Spiral

In 2014 I wrote 'Good Times for Load Testing'. Not anymore.

Less interest to
load testing

Less interest to load
testing tools



Load testing is
less efficient

Less investments in load
testing tools



32

Load Testing Tools Dilemma

Everything as code

- Doesn't quite fit the "performance is everybody's job" paradigm
- Requires programming *and* specialized skills

Scriptless, "AI based"

- Doesn't fit the "everything as code" paradigm
- May be a challenge in more sophisticated cases



33

Agenda

A Short History of Performance Engineering

The Current State of Performance Engineering

What Does the Future Hold?



34

What is Ahead for Performance Testing ?



- Replaced by other methods of performance risk mitigation
 - Due to centralization
- Moving responsibilities
 - Left - to developers
 - Right - to SRE (admins)
- Self-managed systems
 - Auto-scaling, etc.

- System's scale skyrocket
- System's sophistication skyrocket
- Cost of resources becomes significant for large-scale systems
- Cost of failures increases
- Decentralization trends
 - Mobile, P2P, IoT, Fog, Edge



35

Survival Scenario

Continuous PE / PT

- Seamless integration with both Dev and Ops

Performance as code

Holistic view of performance

- Resilience, scalability, etc.

Be the center of performance expertise

**What
Else ?**



36

Alternative Scenario

Performance responsibilities move to

- Developers / SDETs on left - "full automation"
- SREs on right - "testing in production"



Performance testing eventually get extinct

Performance professionals migrate to SDETs/SREs/whatever

"Automated" regression tests remain the only experimental pro-active to mitigate performance risks

- *Leaving holes in performance engineering strategy*



37

Summary

The feeling that we are close to solving performance problems exists last 50+ years

- Probably will stay with us for a while

There are many way to mitigate performance risks

- Define a strategy based on your context
- Centralized environments provide more alternatives to performance testing

Performance testing must adjust to industry trends

- Some trends are clear: continuous testing, automation, performance as code, etc.
- We are rather in the beginning, the future of the trade is not set yet...
 - And a new paradigm is on the horizon



38

JANUARY 2021 VIRTUAL CONFERENCE EVENT

IMPACT 2021

JANUARY 2021 VIRTUAL CONFERENCE EVENT

THE PAST, PRESENT, AND FUTURE OF PERFORMANCE ENGINEERING

CMG
YOUR TECHNOLOGY COMMUNITY