Views from the Trenches: Current Trends in Performance Engineering

Alexander Podelko
Amazon
Stamford, CT, USA
apodelko@yahoo.com

ABSTRACT

Performance engineering is changing before our eyes adjusting to current industry trends – such as cloud computing, agile development, and DevOps. As systems scale and sophistication skyrocket, performance gets more attention. While it looks like some performance concepts like algorithm complexity became a must for anybody working in the industry, it still doesn't result in a consistent view of performance. So it remains an open question what computer professionals should learn about performance – and, even more challenging, what is needed to prepare performance professionals (that actually have never been clearly answered - and now it is even less defined than it was before).

CCS CONCEPTS

• General and reference ~Cross-computing tools and techniques ~Performance •Software and its engineering ~Software organization and properties ~Extra-functional properties ~Software performance •Computer systems organization ~Architectures ~Distributed architectures ~Cloud computing •Software and its engineering ~Software organization and properties ~Software system structures ~Software architectures •Software and its engineering ~Software creation and management ~Software development techniques •Software and its engineering ~Software creation and management ~Software verification and validation

KEYWORDS

Performance engineering, performance testing, continuous performance testing, early performance testing, modeling, queuing theory.

ACM Reference format:

Alexander Podelko. 2020. Views from the Trenches: Current Trends in Performance Engineering. In *the Companion of the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE'23 Companion), April 15–19, 2023, Coimbra, Portugal.* ACM, New York, NY, USA. *3 pages.*: https://doi.org/10.1145/3578245.3584355

ICPE '23 Companion, April 15–19, 2023, Coimbra, Portugal

ACM ISBN 979-8-4007-0072-9/23/04...$15.00.
DOI: https://doi.org/10.1145/3578245.3584355

1 INTRODUCTION

This is a short position paper summarizing author's observations on the current trends in performance engineering and their possible impact to education to be presented at the 4th Workshop on Education and Practice of Performance Engineering (WEPPE 2023). The views expressed here are author's personal views only and do not necessarily represent those of his current or previous employers. These observations are just listed here without diving into details - partially due to length limitation, partially because they are just informal observations not backed by any serious research.

However, any research on these subjects would be very challenging. For example, [2] concluded that "most surveyed companies do not regularly conduct performance evaluations" and there is "low adoption of performance engineering practices in DevOps". It is not surprising that survey gets to such conclusions – as majority of companies don't practice performance engineering in a systematic way (as it always was). Of course, the survey is dated 2016 – but probably results wouldn't be drastically different even if it were conducted today. The issue here is that trends are defined by frontrunners, not by the majority. Surveys may help to evaluate the current state – but it would be always well behind the leading trends.

These trends in performance engineering become evident – but they are still in the beginning. Usually, it takes about a decade for performance engineering to catch up with a change in computing paradigm [9]. Still the main trend is clear - integration of performance engineering into cloud computing and agile development [8,12].

2 ADJUSTING PERFORMANCE ENGINEERING TO CLOUD COMPUTING AND AGILE DEVELOPMENT

The main trends in the industry are cloud computing and agile development (as well as other trends on the top of them - such as DevOps), which drive changes to performance engineering. As mentioned earlier, it may be not evident yet in more traditional environments, but it may be definitely observed in trend-setting

companies. The two more established trends in performance engineering are early performance testing and continuous performance testing [12].

2.1 Early Performance Testing

Agile development should be rather a trivial case for performance testing. The system should be working on each iteration by definition. But you need performance engineer throughout all stages of the project and it drastically increase the amount of work. Savings come from detecting problems early [7, 8].

Early performance testing does require significant changes in approaches. For example:

- from late record/playback performance testing to early performance engineering,
- from system-level requirements to component-level requirements,
- from record/playback approach to programming to generate load and create stubs,
- from, practically, "black box" to "grey box".

2.2 Continuous Performance Testing

Continuous performance testing is the foundation of performance engineering integration into agile and Continuous Integration (CI). While it is limited to regression testing, it is a very important part to integrate performance testing into agile development. It is very context-dependent, there is no one-size-fit-all recipes or out of the box product support beyond trivial cases at the moment [7].

Continuous performance testing also requires significant changes in approaches. The biggest change comparing with traditional approach is the change from realistic testing and checking SLO to coverage and checking the difference between builds. The challenges to implement include [10], [11]:

- integration,
- coverage optimization,
- variability / noise reduction,
- change detection,
- advanced analysis,
- operations / maintenance.

3 PERFORMANCE TOPICS GETTING MORE ATTENTION

It appears that awareness of some performance-related topics significantly increased in the industry and it becomes a must for computer professionals. Not as much as a real need to have a holistic understanding of performance - but at least as the current trend in hiring practices. It is already an established practice that almost every coding or algorithmic interview (which is practically a standard nowadays) has questions about algorithmic complexity (time complexity, space complexity, big-O notation). While everybody should know performance aspects of algorithms nowadays– it appears to be a challenge to connect it with practical performance engineering.

But now even system design interviews heavily involve performance aspects – see, for example, [3], where everything is built around performance and capacity. That is the reflection of getting more attention to performance topics in system architecture – and the best example is the AWS Well-Architected Framework [1]. Out of six pillars three are related to performance – in addition to performance efficiency, there are also reliability and cost optimization. Its applications may be seen, for example, across all AWS training materials and certifications.

There the heavy use of performance patterns and anti-pattern can be seen - incorporating Software Performance Engineering approach, as it was introduced by Dr. Connie Smith in [13, 14], into current architecture practice.

There is significant interest to more practical topics in the industry – where [4] is probably the best review of such topics. For example, using eBPF and flame graphs for performance analysis. However, very few sources are available beyond basic documentation.

That probably means that performance topics should be integrated in standard computer education at least to make students competitive on the labor market. Apparently, it also means that performance professionals should have deeper understanding of all these topics and probably stand out by a more holistic view of performance.

4 PERFORMANCE TOPICS NOT SO POPULAR ANYMORE

The first topic that has significant lesser popularity in the industry than in academia is queueing models. At least since [5] most performance-related books and advanced performance classes had a significant part devoted to queuing theory. While it does have didactic value, its application in industry is rather limited to rather simpler system of significant importance. Actually, multi-core servers, horizontal scalability, and not using servers up to 100% probably allow using linear models in many cases [6]. It is interesting that a quite popular book in the industry [4] had just a half of a page on queuing system.

The new trend is to use Machine Learning (ML) models for modeling performance, but it also rather applicable to stable systems of significant importance. Using them for continuously updated systems is an open question [6].

Another topic losing its popularity in the industry is load testing tools. As performance engineering shifted to other ways to mitigate performance risk and to more closely integrated continuous performance testing, load testing tools are not in the center of discussions anymore [12]. However, it never got track in academia anyway and people usually learned load testing tools on the job. Still traditional record / playback approach (with correlation and parameterization) may have some didactic value and familiarity with such tools may be helpful.

5 CONCLUSION

As systems scale and sophistication skyrocket, performance gets more attention. Some concepts become common knowledge. However, it doesn't result in a holistic view of performance. There are many ways to mitigate performance risks and performance engineering strategy should be defined based on specific context.

Performance engineering is adjusting to industry trends – such as cloud computing, agile development, and DevOps. Some changes are clear: continuous performance testing, early performance testing, further automation and integration. But it is rather in the beginning; the future of the trade is not set yet.

It remains an open question what computer professionals should learn about performance – and, even more challenging, what is needed to prepare performance professionals.

REFERENCES

[1] Amazon Web Site. The 6 Pillars of the AWS Well-Architected Framework. https://aws.amazon.com/blogs/apn/the-6-pillars-of-the-aws-well-architected-framework/

[2] Bezemer C.-P., et al., 2018. How is Performance Addressed in DevOps? A survey on Industrial Practices. DOI: https://doi.org/10.48550/arXiv.1808.06915

[3] Dejwakh V., 2022. System Design Interview Cheat Sheet. https://vahid.blog/post/2022-05-05-system-design-interview-cheat-sheet/

[4] Gregg, B., 2020. Systems Performance, 2nd edition. Pearson.

[5] Jain, R. K., 1991. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley.

[6] Podelko, A., 2019. The Forgotten Art of Performance Modeling. Performance Advisory Council. https://www.slideshare.net/SocitNeotys/pac-2019-alexander-podelko

[7] Podelko, A., 2019. Context-Driven Performance Testing. CMG impact. https://www.alexanderpodelko.com/docs/Context_Driven_Performance_Testing_CMG19.pdf

[8] Podelko, A., Zibitsker, B., 2020. Performance Testing and Modeling for New Analytic Applications. Big Data Conference, Minsk, Belarus. https://alexanderpodelko.com/docs/Performance%20Testing%20and%20Modeling%20of%20New%20Analitic%20Applications.pdf

[9] Podelko, A., 2021. The Past Present and Future of Performance Engineering. CMG imPACt. https://alexanderpodelko.com/docs/The%20Past%20Present%20and%20Future%20of%20Performance%20Engineering%20CMG%202021.pdf

[10] Podelko, A., 2022. A Review of Modern Challenges in Performance Testing. Workshop on Load Testing and Benchmarking of Software Systems (LTB 2022). https://alexanderpodelko.com/docs/LTB22_Review_Modern_Challenges_in_Performance_Testing.pdf

[11] Podelko, A., 2022. Continuous Performance Testing: Challenges and Approaches. CMG imPACt. https://alexanderpodelko.com/docs/CMG%20IMPACT%2022%20-%20Continuous%20Performance%20Testing.pdf

[12] Podelko, A., 2023. The Future of Performance Engineering: Empirical Analysis of Current Trends. CMG imPACt. https://alexanderpodelko.com/docs/IMPACT23_Podelko.pdf

[13] Smith, C. U., 1990. Performance Engineering of Software Systems. Addison-Wesley.

[14] Smith, C. U., Williams L.G., 2001. Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software. Addison-Wesley.