UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Priit Danelson

Case Study: Optimizing the Automated Acceptance Testing Infrastructure at SaleMove

Master's Thesis (30 ECTS)

Supervisor(s): Dietmar Pfahl
Carlos Paniagua

Tartu 2017

Case Study: Optimizing the Automated Acceptance Testing Infrastructure at SaleMove

Abstract:

SaleMove is a software-as-a-service company that places a high emphasis on the quality of its software and is therefore constantly looking to improve its software development process. An important part of that software development process is automated acceptance test execution. This thesis presents solutions that optimize the infrastructure related to that important part of the development process in order to reduce the associated hardware costs, reduce the delay in receiving acceptance testing feedback, and increase the infrastructure's flexibility in dealing with a growing number of acceptance tests.

To achieve these goals, the limitations of the initial infrastructure are analyzed, after which a number of enhancements are proposed. These enhancements are implemented and their impact is evaluated on an experimental duplicate of SaleMove's automated acceptance testing infrastructure in order to avoid negatively interfering with the ongoing software development. Finally, the enhancements are also implemented on Salemove's active automated acceptance testing infrastructure and observations are made about the effects of the enhancements in practical usage.

Keywords:

Acceptance test automation, parallel test execution, distributed test execution

CERCS: P170

Juhtumiuuring: SaleMove'i automatiseeritud vastuvõtutestimise infrastruktuuri optimeerimine

Lühikokkuvõte:

SaleMove on ettevõte, mis pakub tarkvara kui teenust. Kuna SaleMove asetab suurt rõhku oma tarkvara kvaliteedi tagamisele, on ettevõte huvitatud oma tarkvaraarenduse protsessi pidevast täiustamisest. Üks oluline osa selles tarkvaraarenduse protsessis on automatiseeritud vastuvõtutestide käitamine. Käesolev magistritöö optimeeribki selle tarkvaraarenduse protsessi olulise osaga seotud infrastruktuuri, eesmärgiga vähendada infrastruktuuriga seotud riistvaralisi kulutusi, vähendada viivitust vastuvõtutestide tagasiside saamisel ja suurendada infrastruktuuri paindlikkust kasvava arvu vastuvõtutestidega toime tulemise osas.

Nende eesmärkide saavutamiseks analüüsitakse kõigepealt esialgse infrastruktuuri puudusi, mille järel pakutakse välja mitmeid täiendusi. Et vältida negatiivseid mõjusid ettevõtte igapäevatööle, katsetatakse neid täiendusi esialgu eksperimentaalsel koopial SaleMove'i tõelisest automatiseeritud vastuvõtutestimise infrastruktuurist. Peale esialgset mõjude hindamist, võetakse täiendused kasutusele ka SaleMove'i tõelises automatiseeritud vastuvõtutestimise infrastruktuuris ning tehakse tähelepanekuid nende täienduste mõju kohta praktilises kasutuses.

Võtmesõnad:

Vastuvõtutestide automatiseerimine, testide paralleelne käitamine, testide hajus käitamine

CERCS: P170

1 Conclusion

The aim of the thesis was to improve the automated acceptance testing infrastructure at SaleMove in order to reduce the associated hardware costs, shorten the acceptance testing feedback delay, and increase the infrastructure's flexibility in regards to scaling its test parallelization capacity as well as choosing a cloud computing provider for hosting it.

That aim was achieved by analyzing the limitations of the initial infrastructure and proposing a number of enhancements that were implemented and evaluated on an experimental duplicate of the active automated acceptance testing infrastructure in order to avoid interfering with the SaleMove software development process. The enhancements, which consisted of modifying the architecture of the automated acceptance testing infrastructure and improving the efficiency of parallel test execution, resulted in an approximately 80% reduction in acceptance test execution duration, a 40% reduction in monthly hardware costs and greatly improved the scalability of the infrastructure in the experimental setup. As the benefits from the enhancements were so significant when implemented on the experimental automated acceptance testing infrastructure, the enhancements were ultimately also implemented on the active automated acceptance testing infrastructure, where the implemented enhancements showed similar benefits.

Although the enhancements implemented in this thesis were highly specific to the automated acceptance testing infrastructure at SaleMove, some of the findings could nonetheless be considered to be of interest to a wider audience. Firstly, the finding that the latency between the test executor and the Selenium Grid is a critical factor in the performance of distributed Selenium testing due to the relatively inefficient communication protocol being used. Therefore, making it highly recommended to locate the test executor within the same data center as the Selenium Grid components to minimize latency and achieve significantly faster test execution times. And secondly, the finding that managing the Selenium Grid in-house and only outsourcing the necessary hardware via cloud computing providers is at this point in time a significantly more economical option for supplying the remote browser instances necessary for parallel automated test execution than using a SaaS provider such as BrowserStack.

2 References

[1] T. Dyba and T. Dingsoyr, "What Do We Know about Agile Software Development?", *IEEE Software*, vol. 26, no. 5, pp. 6-9, 2009.

[2] C. Solis and X. Wang, "A Study of the Characteristics of Behaviour Driven Development", *37th EUROMICRO Conference on Software Engineering and Advanced Applications*, Oulu, 2011, pp. 383-387.

[3] D. Janzen and H. Saiedian, "Test-driven development concepts, taxonomy, and future direction", *Computer*, vol. 38, no. 9, pp. 43-50, 2005.

[4] A. Bacchelli and C. Bird, "Expectations, outcomes, and challenges of modern code review", *35th International Conference on Software Engineering (ICSE)*, San Francisco, 2013, pp. 712-721.

[5] A. Begel and N. Nagappan, "Pair programming: what's in it for me?", *Proceedings Second ACM-IEEE international symposium on Empirical software engineering and measurement (ESEM '08)*, New York, 2008, pp. 120-128.

[6] P. Duvall, S. M. Matyas, and A. Glover, *Continuous Integration: Improving Software Quality and Reducing Risk*, Addison-Wesley, 2007.

[7] J. Thönes, "Microservices", *IEEE Software*, vol. 32, no. 1, pp 116-116, 2015.

[8] D. M. Rafi, K. R. K. Moses, K. Petersen and M. V. Mäntylä, "Benefits and limitations of automated software testing: Systematic literature review and practitioner survey", *7th International Workshop on Automation of Software Test (AST)*, Zurich, 2012, pp. 36-42.

[9] C. Parnin and S. Rugaber, "Resumption Strategies for Interrupted Programming Tasks", *Software Quality Journal*, vol. 19, no. 1, pp. 5-34, 2010.

[10] S. Jerkku, "Case study: Improving the Performance of Automated Acceptance Testing with Selenium", M. S. thesis, Univ. Tartu, Tartu, Estonia, 2016.

[11] A. H. Karp and H. P. Flatt, "Measuring parallel processor performance", *Communications of the ACM*, vol. 33, no. 5, pp. 539-543, 1990.

I. License

Non-exclusive licence to reproduce thesis

I, **Priit Danelson**,

(*author's name*)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

reproduce, for the purpose of preservation, including for the purpose of preservation in the DSpace digital archives until expiry of the term of validity of the copyright

Case Study: Optimizing the Automated Acceptance Testing Infrastructure at SaleMove,

(title of thesis)

supervised by Dietmar Pfahl,

(supervisor's name)

2. Making the thesis available to the public is not allowed.
3. I am aware of the fact that the author retains the right referred to in point 1.
4. This is to certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **22.05.2017**