PRESTO uses feedback-directed adaptive test scripts to locate most computationally intensive execution profiles and bottlenecks.

Several papers focused on improving the performance of applications deployed in the cloud [17, 27, 5, 29, 14, 6, 12, 33]. Klein et al. [19] defined a self-adaptation programming paradigm to "skip" optional functionality in the cloud-deployed applications. Frey et al. [8] used a simulation-based genetic algorithm for finding optimized cloud deployment options for the software in the cloud. An approach, ATUoCLES, allows collecting execution information for applications, which have all the logic to scale up and down automatically [9]. Spinner et al. proposed a model-based approach to improve AUT performance by adding/removing VMs [33]. However, none of these approaches analyze impact of specific inputs on the performance of deployed programs and efficient resource allocation in the cloud-based environments, which is done in PRESTO.

# 7. CONCLUSION AND FUTURE WORK

Our novel solution for *Provisioning Resources with Experimental SofTware mOdeling (PRESTO)* enhances cloud elasticity by learning and refining models of under-constrained applications throughout performance testing and using these models stakeholders can craft resource provisioning strategies for the cloud that are highly tailored for specific applications. Experimental results suggest that PRESTO is effective and efficient - up to 40% better response in provisioning resources on average when the AUT throughput worsened significantly. In summary, we extend the theory of cloud computing by utilizing performance testing in its load balancing and resource provisioning. We believe that our work is a successful attempt of using software engineering artifacts to guide cloud deployment of software. The future work will involve automatically searching for scaling operators to (de)allocate different resources to VMs and determining the provisioning strategies to maintain AUT's performance at an acceptable level.

## Acknowledgements

This work is supported in part by the NSF IIP-1547597 and NSF CCF-1217928 grants. Any opinions, findings, and conclusions expressed herein are the authors' and do not necessarily reflect those of the sponsors.

# 8. REFERENCES

[1] A. Arcuri and L. Briand. A practical guide for using statistical tests to assess randomized algorithms in software engineering. In *ICSE '11*.

[2] A. Arcuri and L. Briand. A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering. *STVR*, 2012.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Commun. ACM*.

[4] K. Birman, G. Chockler, and R. van Renesse. Toward a cloud computing research agenda. *SIGACT News'09*.

[5] P. C. Brebner. Is your cloud elastic enough?: Performance modelling the elasticity of infrastructure as a service (iaas) cloud applications. In *ICPE '12*.

[6] F. Brosig, N. Huber, and S. Kounev. Automated extraction of architecture-level performance models of distributed component-based systems. In *ASE'11*.

[7] Cloudstack. http://cloudstack.apache.org/.

[8] S. Frey, F. Fittkau, and W. Hasselbring. Search-based genetic optimization for deployment and reconfiguration of software in the cloud. In *ICSE '13*.

[9] A. Gambi, W. Hummer, and S. Dustdar. Automated testing of cloud-based elastic systems with autocles. In *ASE '13*.

[10] Google. Auto scaling on the google cloud platform. https://cloud.google.com/resources/articles/auto-scaling-on-the-google-cloud-platform.

[11] M. Grechanik, C. Fu, and Q. Xie. Automatically finding performance problems with feedback-directed learning software testing. In *ICSE'12*.

[12] N. Huber, A. van Hoorn, A. Koziolek, F. Brosig, and S. Kounev. Modeling run-time adaptation at the system architecture level in dynamic service-oriented environments. *SOCA'14*.

[13] S. Islam, K. Lee, A. Fekete, and A. Liu. How a consumer can measure elasticity for cloud platforms. In *ICPE '12*.

[14] P. Jamshidi, A. Ahmad, and C. Pahl. Autonomic resource provisioning for cloud-based software. In *SEAMS '14*.

[15] Z. M. Jiang, A. E. Hassan, G. Hamann, and P. Flora. Automated performance analysis of load tests. In *ICSM '09*.

[16] JMeter. https://jmeter.apache.org.

[17] K. Johnson, S. Reed, and R. Calinescu. Specification and quantitative analysis of probabilistic cloud deployment patterns. In *HVC'11*.

[18] JPetStore. http://sourceforge.net/projects/ibatisjpetstore.

[19] C. Klein, M. Maggio, K.-E. ρ Arzén, and F. Hernández-Rodriguez. Brownout: Building more robust cloud applications. In *ICSE '14*.

[20] D. Lo and S. Maoz. Scenario-based and value-based specification mining: Better together. In *ASE '10*.

[21] D. Lorenzoli, L. Mariani, and M. Pezzè. Automatic generation of software behavioral models. In *ICSE '08*.

[22] Q. Luo, A. Nair, M. Grechanik, and D. Poshyvanyk. Forepost: Finding performance problems automatically with feedback-directed learning software testing. *EMSE*, 2016.

[23] H. Mendelson. Economies of scale in computing: Grosch's law revisited. *Commun. ACM*.

[24] NetScaler. http://www.citrix.com/netscalervpx.

[25] T. Ohmann, M. Herzberg, S. Fiss, A. Halbert, M. Palyart, I. Beschastnikh, and Y. Brun. Behavioral resource-aware model inference. In *ASE '14*.

[26] T. Ohmann, K. Thai, I. Beschastnikh, and Y. Brun. Mining precise performance-aware behavioral models from existing instrumentation. In *ICSE Companion '14*.

[27] D. Perez-Palacin, R. Calinescu, and J. Merseguer. Log2cloud: Log-based prediction of cost-performance trade-offs for cloud deployments. In *SAC '13*.

[28] PRESTO. http://www.cs.wm.edu/semeru/data/ICPE16-PRESTO/.

[29] M. Sedaghat, F. Hernandez-Rodriguez, and E. Elmroth. A virtual machine re-packing approach to the horizontal vs. vertical elasticity trade-off for cloud autoscaling. In *CAC '13*.

[30] A. Shankar, M. Arnold, and R. Bodik. Jolt: Lightweight dynamic analysis and removal of object churn. In *OOPSLA '08*.

[31] D. Shen, Q. Luo, D. Poshyvanyk, and M. Grechanik. Automating performance bottleneck detection using search-based application profiling. In *ISSTA '15*.

[32] J. E. Smith and R. Nair. The architecture of virtual machines. *Computer'05*.

[33] S. Spinner, S. Kounev, X. Zhu, L. Lu, M. Uysal, A. Holler, and R. Griffith. Runtime vertical scaling of virtualized applications via online model estimation. In *SASO'14*.

[34] J. Wildstrom, P. Stone, E. Witchel, and M. Dahlin. Machine learning for on-line hardware reconfiguration. In *IJCAI'07*.