

Cloud Computing and SBSE

David R. White

School of Computing Science, University of Glasgow, Scotland, G12 8QQ.
david.r.white@glasgow.ac.uk

Abstract. Global spend on Cloud Computing is estimated to be worth \$131 billion and growing at annual rate of 19% [1]. It represents one of the most disruptive innovations within the computing industry, and offers great opportunities for researchers in Search Based Software Engineering (SBSE).

In the same way as the development of large scale electricity generation, the physical centralisation of computing resources that Cloud Computing involves provides opportunities for economies of scale [2]. Furthermore, it enables more extensive consolidation and optimisation of resource usage than previously possible: whereas in the past we have been resigned to the phenomenon of wasted spare cycles, the Cloud offers opportunities for large-scale consolidation, along with standardised measurement and control of resource consumption.

There are two key stakeholders within the Cloud Computing sector: providers, such as Amazon and Google, and clients, the consumers of Cloud services paying for infrastructure and other utilities by the hour. Both providers and clients are greatly concerned with solving optimisation problems that determine their resource usage, and ultimately their expenditure. Example resources include CPU time, physical RAM usage, storage consumption, and network bandwidth.

From the point of view of the provider, improving utilisation leads to greater return on capital investment – hence the emergence of oversubscription policies [3], where more resources are promised to clients than are physically provisioned. Similarly, clients wish to reduce their resource demands because there is a direct relationship between resource consumption and operating cost. The rise of Cloud Computing has in many cases made explicit the cost of computation that was previously considered only as depreciation of capital investment.

Example optimisation problems include: the configuration of servers, virtualisation, and operating systems to reduce storage and memory usage; transformation of software and architectures to adapt them to a Cloud platform; the intelligent placement of virtual machines to improve consolidation, reduce business risks and manage network usage; extensive, online and continuous testing to improve robustness; policy-level decisions such as management of scalability, demand modelling and spot market participation; and automated online software maintenance and improvement [4].

A relentless focus on efficiency offers researchers in SBSE the opportunity to make an important contribution. The field has an established record in solving similar optimisation problems within software engineering, and treating non-functional concerns as first class objectives.

The way in which software is deployed within Cloud systems also offers exciting possibilities for SBSE researchers, principally due to its centralised nature.

Software is no longer deployed or versioned in the traditional manner, and this consequently enables dynamic, controlled and sometimes partial deployment of new code (a procedure known as “canarying” [5]). A key insight is that many different versions of software can be employed simultaneously, and that manipulation of this multifaceted distribution solves some of the problems traditionally faced by SBSE practitioners.

For example, consider recent work in bug fixing [6] [7]. One of the greatest challenges facing automatically and dynamically applying such work is ensuring that the existing semantics of a program are not disturbed in the process of correcting a bug. By running a form of N-Version programming on a Cloud platform, differences in behaviour caused by automated software repair could be rejected or reported.

Similarly, the extensive use of virtualisation improves the repeatability of execution. Technologies such as Xen [8] provide snapshotting, enabling a virtual machine to be paused, cloned and restarted (with some limitations). This lends itself to improving the robustness of automated repair and optimisation methods: we can rollback servers and examine hypothetical scenarios. Furthermore, due to the elasticity of the Cloud, we can quickly procure extra execution environments to test software on demand.

Applying traditional SBSE work to the Cloud will not be straightforward, and requires further development of both theory and implementation: many problems in the Cloud are real-time and dynamic; there are competing objectives and complex relationships between concerns; the problems faced and the software itself are distributed. One exciting example of how we may begin to adapt to this environment is Yoo’s proposed amortised optimisation [9].

The process of researching Cloud Computing is in itself a challenge, because the nature of the industry is secretive and closed. There is competitive advantage to be gained in proprietary technology, and as a result much detail of how deployed systems function is not available. Thus, we must occasionally re-invent the wheel and in some cases restrict ourselves to synthetic case studies or else recognise the limited reach we can have within certain subdomains.

More pressing is the concern about research evaluation: how can we evaluate our research ideas effectively? This problem is not limited to SBSE alone. There are parallels with difficulties carrying out networking research, and we may adopt that field’s existing methods of modelling, simulation, testbed construction, and instrumentation. A recent and novel alternative is to construct a scale model, such that a low-cost system with similar logical properties to a commercial Cloud can be used for prototyping [10].

In summary, the rise of Cloud Computing provides great opportunities for SBSE research, both in terms of the problems that SBSE may be used to solve, and also the capabilities that Cloud Computing provides, which offer truly new methods of deploying SBSE techniques.

References

1. Columbus, L.: Gartner Predicts Infrastructure Services Will Accelerate Cloud Computing Growth. *Forbes* (19/02/2013.)
2. Carr, N.: *The Big Switch: Rewiring the World, from Edison to Google*. W.W. Norton & Company (2009)
3. Williams, D., Jamjoom, H., Liu, Y., Weatherspoon, H.: Overdriver: Handling Memory Overload in an Oversubscribed Cloud. *SIGPLAN Not.* **46**(7) (2011) 205–216
4. Harman, M., Lakhoria, K., Singer, J., White, D.R., Yoo, S.: Cloud Engineering is Search Based Software Engineering Too. *Systems and Software* (To appear. Available online 23 November 2012)
5. Barroso, L., Hölzle, U.: *The Datacenter as a Computer*. Morgan & Claypool Publishers (2009)
6. Le Goues, C., Nguyen, T., Forrest, S., Weimer, W.: GenProg: A Generic Method for Automatic Software Repair. *IEEE Trans. Software Engineering* **38**(1) (2012) 54–72
7. Arcuri, A., Yao, X.: A Novel Co-evolutionary Approach to Automatic Software Bug Fixing. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*. (2008)
8. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the Art of Virtualization. *SIGOPS Oper. Syst. Rev.* **37**(5) (2003) 164–177
9. Yoo, S.: NIA³CIN: Non-Invasive Autonomous and Amortised Adaptivity Code Injection. Technical Report RN/12/13, Department of Computer Science, University College London (2012)
10. Tso, F., White, D.R., Jouet, S., Singer, J., Pezaros, D.: The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures. In: *International Workshop on Resource Management of Cloud Computing 2013*. (To Appear)