

## Annotated Bibliography

Does "Cloud Computing" improve software quality, performance and reliability?

*Mesbahi, Mohammad Reza, Rahmani, Amir Masoud, and Hosseinzadeh, Mehdi. "Reliability and high availability in cloud computing environments: a reference roadmap." Journal of Cloud Computing: Advances, Systems and Applications, vol. 6, no. 1, 2017, pp. 1-16.*

The paper explains how cloud computing has been used in recent years for providing highly available and reliable services in order to maintain “customer confidence, satisfaction, and prevent revenue losses”. The primary purpose of the paper, however, is to explain how while current methodologies can be useful, there has never been an in-depth study into how cloud computing can be used to cover all of the different aspects of the problem space. Thus, this paper aims to create a sort of roadmap on how one could use cloud computing with their services/products in order to ultimately reach a state of reliability and confidence from their customers.

Despite this, there are still multiple points made in the paper that elaborate on the usefulness of cloud computing in software quality, such as the problem of constant demand and requirement changes. In this regard, cloud computing offers a scalable and flexible infrastructure for software systems, allowing developers of the system to quickly adapt when needed. One of the examples described in the paper mentions how cloud services offer multiple failure detection algorithms across all nodes in a system that can be used to quickly respond to and resolve faults before they turn into errors. These algorithms include components such as fault masking managers designed to hide occurrences of faults from users, and recovery managers in charge of minimizing downtime by resolving error prone nodes.

On a personal note, I have seen such systems in action in my experience working at Best Buy, specifically using a service called SolrCloud which is in charge of managing both the indexing and retrieval of documents. This service similarly offers the use of various nodes in charge of those jobs, and in the event of a fault it's able to switch traffic from faulty nodes to ones still running. In many cases it is also able to resolve the faulty node on its own, however it also offers an alerting system which we use when we need to manually resolve issues. The reality of having to manually resolve these faults when it comes to using cloud computing is something that the paper doesn't mention, emphasizing the imperfectness of failure detection algorithms, and ultimately highlighting that not every problem can be solved by an automated service. Despite this, as a whole, cloud services can still be very useful in the management of faults, providing reliability in any application that implements it.

Continuing on with the idea of fault management systems, the paper also briefly emphasizes how these systems are cost-effective, thus leading to applications that are more easily maintained. This benefit is one that I wish the paper elaborated on more, perhaps even including studies that quantitatively show how implementing cloud services that use fault management systems can lead to decreases in operational costs. While this can be implicitly surmised when thinking about how this can allow developers to spend less time resolving faults and save companies money when it comes to not having to lose customers due to downtime, being able to actually see the numbers would go a long way in proving the relationship between cloud services and operational costs.

*Candea, George, Stefan Bucur, and Cristian Zamfir. "Automated Software Testing as a Service." IEEE Software, vol. 33, no. 6, Nov./Dec. 2016, pp. 32-38, doi: 10.1109/MS.2016.138.*

In this paper, we take a look at how cloud computing can improve quality assurance testing for companies that operate under SaaS through the idea of automated testing as a software (TaaS). They look at how TaaS can be invaluable in maintaining the quality of a company's service throughout the lifespan of that service. The authors first point out how the current state of testing is often slow, labor-intensive and error-prone. They go on to mention that manual testing is no longer a useful method of verification since today's software companies operate in fast-paced environments, requiring the need of automated testing in order "to ensure that software is delivered quickly and with high quality." Through the use of TaaS, developers can improve the quality of their software by having the ability to automatically run tests and receive reports on them within minutes (or occasionally hours). These reports can give developers an in-depth and detailed look into the state of their software, allowing them to resolve defects more accurately and in a quick fashion.

As engineers, I believe we can resonate with the notion of manual testing often being slow and error-prone. One of the many principles of testing is understanding that exhaustive testing is impossible—we will never be able to catch every defect in a piece of software. And while TaaS doesn't solve this issue entirely, it definitely alleviates it by allowing developers to spend more time thinking about how new tests can be implemented (and thus increase coverage) and less time manually running existing tests.

The paper also goes on to mention that one of the primary benefits of TaaS is that it allows developers to easily scale their testing procedures. As an organization becomes more and more popular, one of the most important things to think about next is ensuring that their service can handle load stresses due to increased traffic. By utilizing automated tests during performance/load testing, developers are given the opportunity to run tests at higher and higher scales through various load balancing techniques. This way, developers can more accurately

emulate how their service would function when more people begin to use their service, without being limited by the effects of manual testing.

Utilizing TaaS techniques in my day to day has become an integral part of my teams verification process. Since Best Buy is a large company, we often run performance tests on our systems in order to ensure that we don't run into any failures or issues regarding page crashes and uptime. We've been able to use TaaS to emulate customer searches on the website by sending in thousands of requests a minute in order to put a strain on the system. This has allowed us to accurately determine the reliability of our systems, ensuring that we can handle high traffic, in order to maintain the quality of our software.