

INTRODUCTION

Rackspace aims to deliver a superior OpenStack® cloud with industry-leading reliability, unmatched scalability and innovation-driven agility. Over the past several years, our experience with Swift has led us to uncovering some challenges that have led toward an innovative approach to boosting Swift performance.

SWIFT CHALLENGES IN THE PUBLIC CLOUD

Our Public Cloud team originally wrote Swift in Python, and open-sourced it in 2010. We were the earliest adopter, and we're extremely familiar with it. Using Swift on a massive scale in our Public Cloud, as well as for Private Cloud customers, we're uniquely positioned to address limitations others might not see. And we have a strong track record of working creatively to resolve problems.

For example, as our Swift-based Cloud Files product grew at Rackspace, we encountered scaling issues that weren't easily resolved. Because Cloud Files usage is rising, we took an unusual path to resolving scaling issues. Our team saw an opportunity to gain greater concurrency and better handling of blocking I/O operations by exploring other programming languages. During 2015, our Public Cloud team worked to rewrite the object-server, object-replicator and object-auditor components of the original swift code in Go instead of Python. We call that improvement Hummingbird. These rewritten components are available today as a feature branch in the OpenStack Swift repo on GitHub[1].

Our approach worked. After a methodical introduction to production, the Public Cloud team [2] reported significant gains in reliability and consistency, making Cloud Files one of the safest Swift based object storage clusters available.

SOLVING A PRIVATE CLOUD ISSUE

Early in 2016, Rackspace was tasked to deploy a private, multi-petabyte, geographically dispersed Swift cluster to protect billions of small files. The cluster was distributed across two regions, with primary operations in Ashburn, Virginia and failover to Chicago, Illinois, with multiple copies of objects in both locations for disaster recovery purposes. At first, it seemed as though having multiple copies in each location would serve to enhance performance to acceptable levels. However, during deployment and acceptance testing, we discovered that

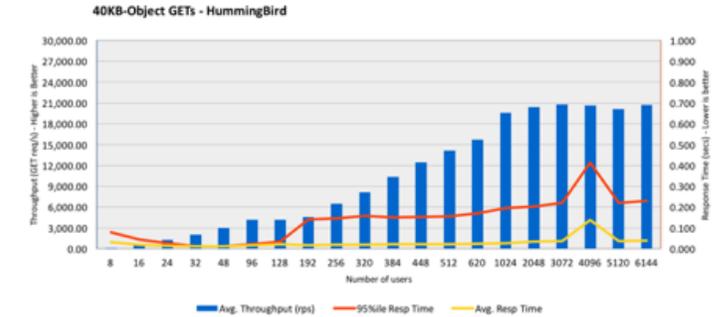
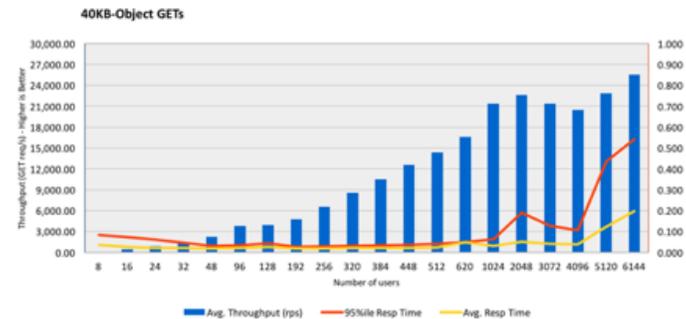
performance was not at a level we felt was acceptable. We saw inconsistent response times and replication passes that took excessive amounts of time to complete.

The enhancements we created for our Public Cloud were exactly what we were looking for in this private cloud deployment. To ensure consistent deployments of Rackspace OpenStack Private Cloud, we avoid carrying any custom patches or deploying code that is not in the master. However, our customer deserved better performance, and we felt that for the success of this environment, we had to leverage the innovation from our own Public Cloud team.

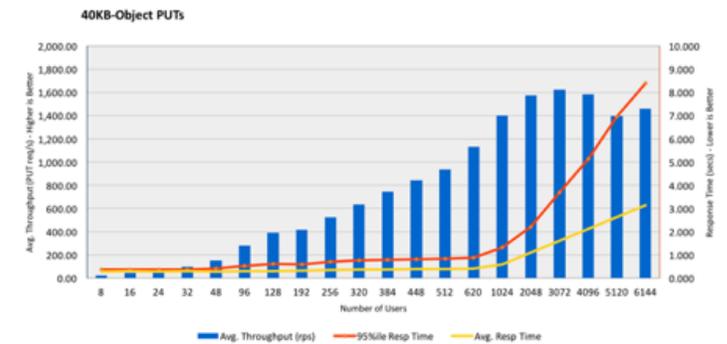
The modular nature of Swift meant that during testing, it was straightforward to switch between the Python and Go object-server, object-replicator and object-auditor, by simply stopping the daemons and starting the alternates. It is even possible to run in a mixed configuration if operations must continue during the switchover, as proven during the Cloud Files team's testing in 2015.

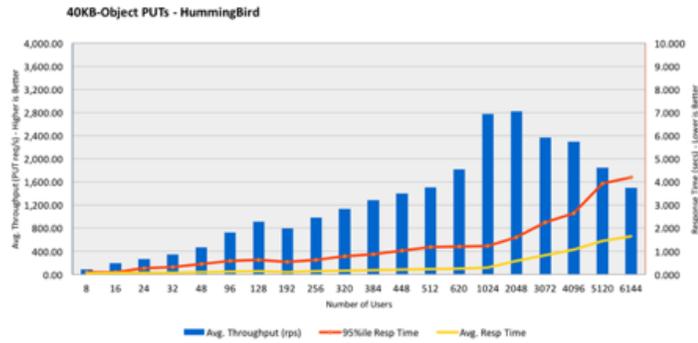
PROVEN RESULTS

If we look at synthetic object GET testing results from this environment, Hummingbird provides similar throughput. However, response times are more predictable as load on the system increases. Most importantly, the throughput remains stable as the load on the system reaches saturation.



Similarly, with synthetic PUT testing, we see some levels of improved system throughput. But most importantly, we see lower response times as the system reaches saturation.





[1] - <https://github.com/OpenStack/swift/tree/feature/hummingbird>

[2] - <http://blog.rackspace.com/making-OpenStack-powered-rackspace-cloud-files-buzz-with-hummingbird>

[3] - <https://review.OpenStack.org/#/c/346243/>

While the OpenStack Technical Committee has recently concluded that the selective use of alternative programming languages is not an option[3] presently, our industry-leading scale and operational experience allowed us to identify and utilize an alternate approach where upstream Swift was failing.

Conclusion

This is a case where we took an innovative approach to resolving a customer issue. Not many organizations would go so far, but at Rackspace, we're here to deliver the best OpenStack experience in the world. We've been able to demonstrate that our approach offers compelling benefits, including:

- Simple implementation
- Easy operation in a mixed configuration
- Increased consistency and reliability.

In short, our approach delivers compelling value.

To learn more about our work to enhance both public and private clouds while delivering industry-leading reliability, unmatched scalability, innovation-driven agility, and a superior approach to OpenStack, sign up for our free OpenStack strategy session at <http://go.rackspace.com/OpenStackExperts>

ABOUT RACKSPACE

Rackspace (NYSE: RAX), the #1 managed cloud company, helps businesses tap the power of cloud computing without the complexity and cost of managing it on their own. Rackspace engineers deliver specialized expertise, easy-to-use tools, and Fanatical Support® for leading technologies developed by AWS, Google, Microsoft, OpenStack, VMware and others. The company serves customers in 120 countries, including more than half of the FORTUNE 100. Rackspace was named a leader in the 2015 Gartner Magic Quadrant for Cloud-Enabled Managed Hosting, and has been honored by Fortune, Forbes, and others as one of the best companies to work for.

Learn more at www.rackspace.com or call us at **1-800-961-2888**.

© 2016 Rackspace US, Inc.

This whitepaper is provided "AS IS" and is a general introduction to the service described. You should not rely solely on this whitepaper to decide whether to purchase the service. Features, benefits and/or pricing presented depend on system configuration and are subject to change without notice. Rackspace disclaims any representation, express or implied warranties, including any implied warranty of merchantability, fitness for a particular purpose, and non-infringement, or other commitment regarding its services except for those expressly stated in a Rackspace services agreement. This document is a general guide and is not legal advice, or an instruction manual. Your implementation of the measures described may not result in your compliance with law or other standard. This document may include examples of solutions that include non-Rackspace products or services. Except as expressly stated in its services agreements, Rackspace does not support, and disclaims all legal responsibility for, third party products and services. Unless otherwise agreed in a Rackspace service agreement, you must work directly with third parties to obtain their products and services and related support under separate legal terms between you and the third party.

Rackspace cannot guarantee the accuracy of any information presented after the date of publication.

Rackspace®, Fanatical Support® and other Rackspace marks are service marks or registered services of Rackspace US, Inc. and are registered in the United States and other countries. Other Rackspace or third party trademarks, service marks, images, products and brands remain the sole property of their respective holders and do not imply endorsement or sponsorship.

