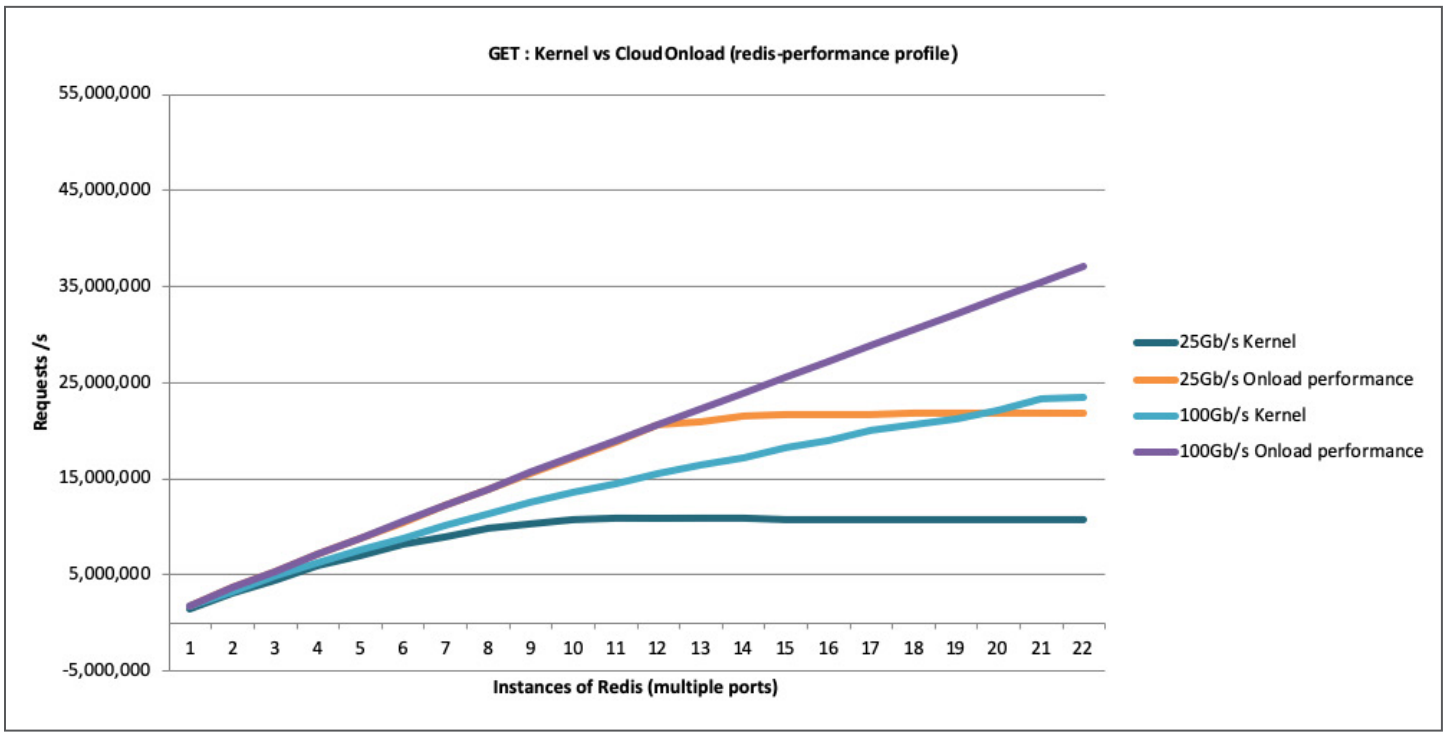
Redis Running with Cloud Onload® Sees a 100% Performance Gain

What is Redis?

Redis is an in-memory data store, often used as a database, cache or message broker. It supports a wide variety of data structures, has built-in replication, scripting, and many other capabilities. By design, Redis' RESP protocol is implemented over TCP and is heavily network dependent, so improvements in TCP packet handling can provide significant overall performance gains.

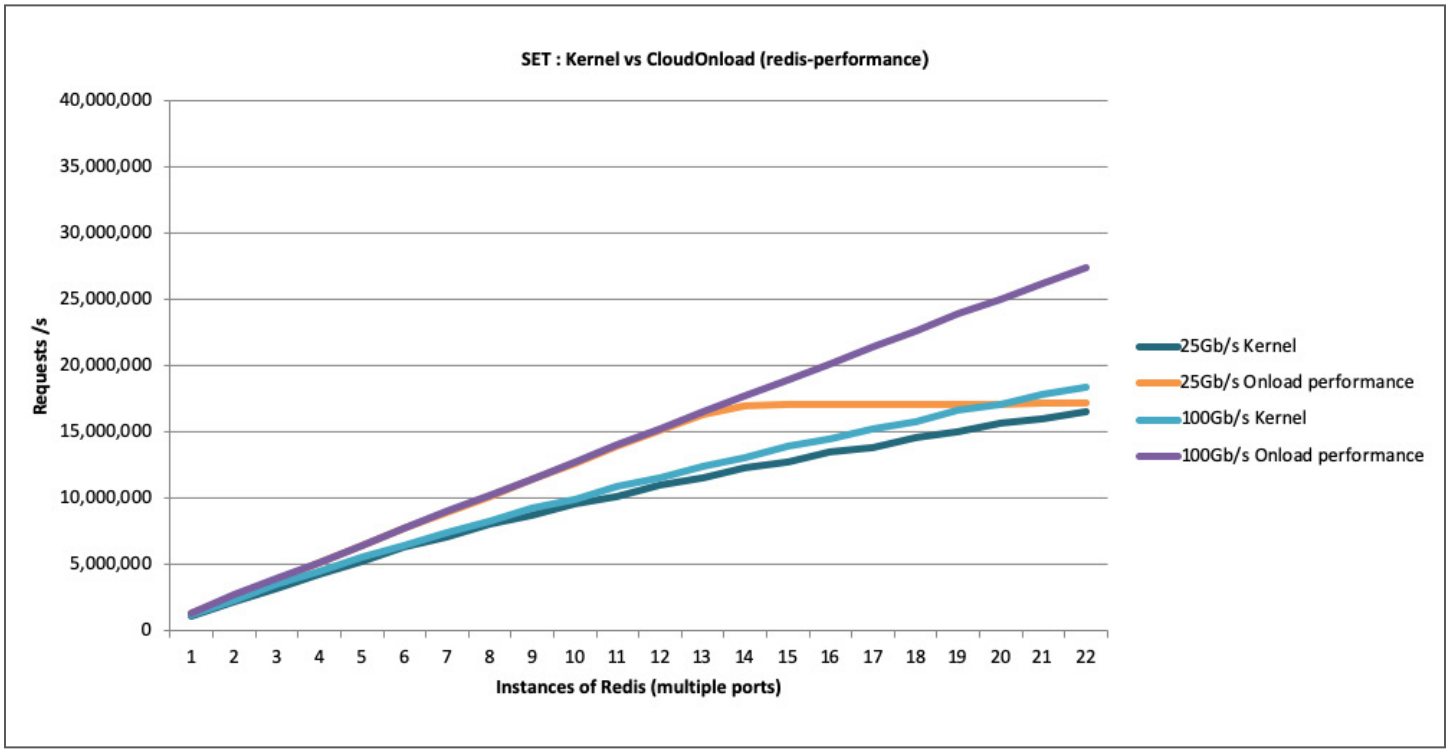


Key Observations from Performance Testing

- Solarflare's Cloud Onload delivers a 100% performance gain for Redis GET transactions when using 25GbE with 10 Redis instances, and at 12 instances an 88% improvement for SETs.
- When processing GET requests using 10 or fewer Redis instances Cloud Onload on 25GbE outperforms Redis utilizing the kernel communications stack with a 100GbE connection.
- We soon reach the limits of the Redis application or the physical system under test. If we look at 25GbE with Cloud Onload we can see that the link reaches a maximum of 10 million requests per second so then by extension a 100GbE link should level off at 40 million requests per second. Our testing only reaches roughly 25 million or 62.5% of what has been predicted.
- Cloud Onload is able to fully utilize either a 10GbE or 25GbE link as shown by the plateaus in the results. The kernel driver is not able to achieve this at 25GbE. Redis using 25GbE and the kernel driver never approaches this plateau, but at 11 instances the slope of the performance gain is nearly flat.

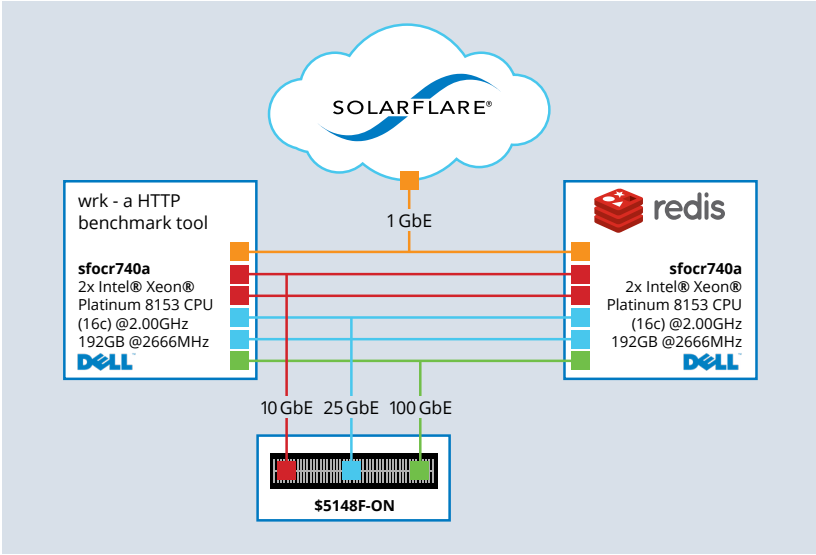
Why Redis Benefits from Kernel Bypass

Since Redis is network intensive, every request includes network processing overhead. Whenever an application like Redis touches hardware, other than the CPU or memory, and in this case the network, it must make at least one, and sometimes several calls to the operating system kernel. Each request is additional overhead that requires both CPU cycles and processing time. Solarflare's Cloud Onload moves the network processing required by Redis from the kernel into Redis's own application space in memory. This single modification improves the performance of Redis by 50% on average and in some cases by as much as 100% as can be seen in these graphs.



Description of Test Platforms

For this testing we used two dual socket Intel Xeon systems labeled "A" and "B", and three production networks, 10GbE, 25GbE and 100GbE, leveraging a single switch dedicated to this testing, no other traffic exists on this switch. The "A" system was used as the server and it had two Platinum 8153 CPUs clocked at 2GHz with 16 cores per processor, the "B" system was used as the client utilizing two Gold 5120 CPUs clocked at 2.2Ghz but with 14 core processors. Both systems contain 192GB of memory, they boot from an NVMe flash drive, and have three Solarflare network cards: SFN8522 dual port 10GbE, X2522-25G dual port 25GbE and an X2541 single port 100GbE. This enables us to test performance against a range of shipping Solarflare adapters as shown above.



Tuning Configuration

Here are the changes we made to the standard install beyond simply leveraging Cloud Onload.

- We increased the maximum number of open files from 1,024 to 10,032.
- Raise somaxconn to 65,535 enabling 65,535 connections versus originally 128.
- Enable Vm.overcommit_memory by setting it to 1 from 0.
- Disable Transparent Huge Pages (THP).

Observations

As an in-memory data store, the performance of Redis is not gated by traditional storage I/O. Redis relies on the operating system's communications stack to process network I/O requests, but in high core count environments, this stack has become the new bottleneck. Here are some additional points to consider:

- Redis with Cloud Onload can service up to 33 million Get requests/second, while Redis using the operating system kernel can only handle 23 million Get requests/second, a 43% improvement.
- Therefore Redis with Cloud Onload can potentially reduce your capex by 43%. In simple terms every two Redis servers leveraging Cloud Onload can service the same number of requests as three Redis servers using the standard Linux kernel.
- Conversely, adding Cloud Onload and a Solarflare 25GbE or 100GbE NIC to existing servers will provide additional headroom for growth or unanticipated business peaks as shown above.

For More Testing Details

Check out Solarflare's **Cloud Onload Cookbook** for the exact installation and testing process along with the specific tuning and tweaking commands executed above.

The above benefit statements are the result of benchmarking designed to focus on the value of optimizing networking through Cloud Onload kernel bypass. Real world use cases are not the same as benchmarks and as such the role that networking plays may vary, so your overall measurable benefits may be different.

For more information please visit:

solarflare.com

Contact Us:

US +1 949 581 6830
UK +44 (0) 1223 477171
HK +852 2624 8868
Email: sales@solarflare.com

SF-121512-CD Issue 3
Redis Benchmark Results 06052019