ESG Lab Review

Robin Systems: Container-based Virtualization for Databases and Data-centric applications

Date: January 2017 **Authors:** Evan Marcus, Lab Analyst; and Michael Leone, Senior Lab Analyst

Abstract

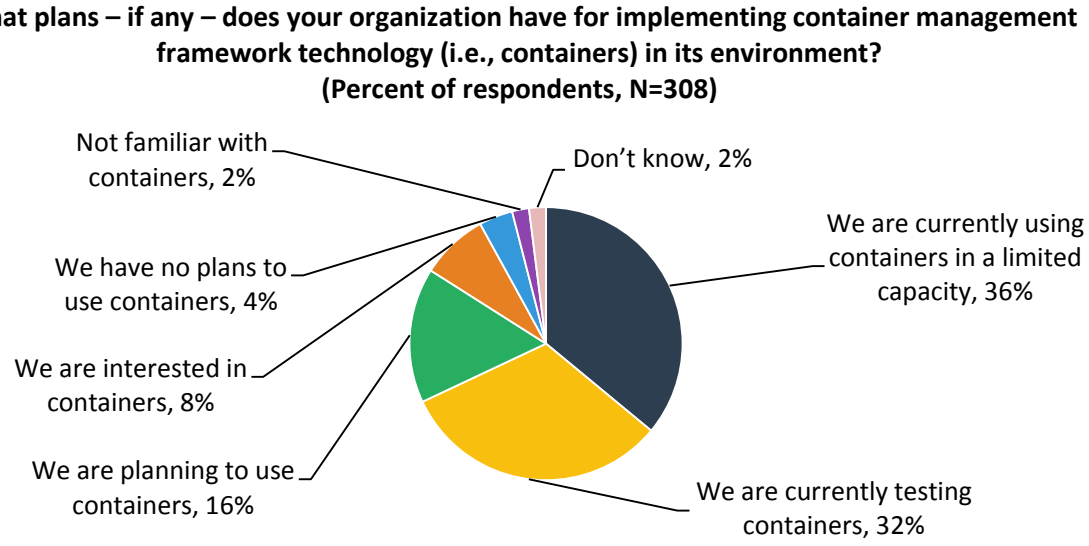
This ESG Lab Report highlights the recent testing of Robin Container-Based Virtualization Platform. Using a combination of guided demos and audited performance results, ESG Lab validated the ease of use, performance, scalability, and efficiency of Robin Systems' container-based architecture.

The Challenges

Containers optimize application deployment by bundling all of the application's required components into a single package, including supporting libraries and configuration files. Containers only require a supported Linux kernel to operate, making it easy to move them between environments, e.g., between hosts, from dev to test, or from test to production.

Organizations are discovering that existing data center infrastructure is not capable of dealing with large number of containerized applications since a single modern microservices-based web application can easily span hundreds or more containers. Organizations run many applications and often find their systems administration teams overwhelmed attempting to match resources with containers. Containers improve server utilization by allowing multiple applications to run on the same server. But since all applications share the same storage, storage performance can be erratic, which impacts overall application performance. To combat this, some organizations deploy critical applications on siloed infrastructure to ensure good performance, which leads to overprovisioned hardware and poor resource utilization.

Figure 1. Plans for Deploying Container Management Framework Technology



Source: Enterprise Strategy Group, 2017

As shown in Figure 1, recent ESG research indicates that 68% of organizations are testing or using containers today, and another 16% are planning to start using them soon.[1] The benefits of containers—including easy, consistent application deployment and light overhead when compared with virtual machines and hypervisors—make them appealing for a variety of applications. However, because they can require close monitoring of resource consumption and quick response time

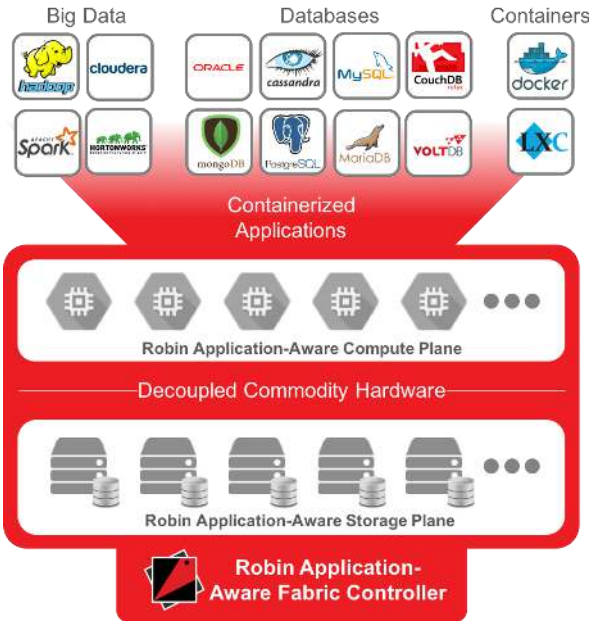
[1] Source: ESG Research Report, *The Cloud Computing Spectrum, from Private to Hybrid*, March 2016.

when those resources run short, containers still need a significant amount of administration to ensure they remain efficient over time.

The Solution: Robin Systems Containerization Platform

The Robin Systems Containerization Platform is a container-based, application-aware compute and storage platform. The software effectively abstracts underlying server, VM, network, and storage boundaries to produce a compute, storage, and data continuum. Many different containerized applications can run in this continuum without impacting one another's performance. Application portability and scalability are increased because compute and storage are decoupled; applications can be freely moved around the continuum without moving or copying data. Complex distributed applications like NoSQL, Hadoop, Cassandra, and Mongo can be deployed quickly and easily. Robin's application-level snapshots that include the entire application environment make it easy to quickly create a copy of a production environment without impacting production performance, and to quickly roll back to a previous point in time to correct a problem.

Figure 2. Robin Systems Architecture



Source: Enterprise Strategy Group, 2017

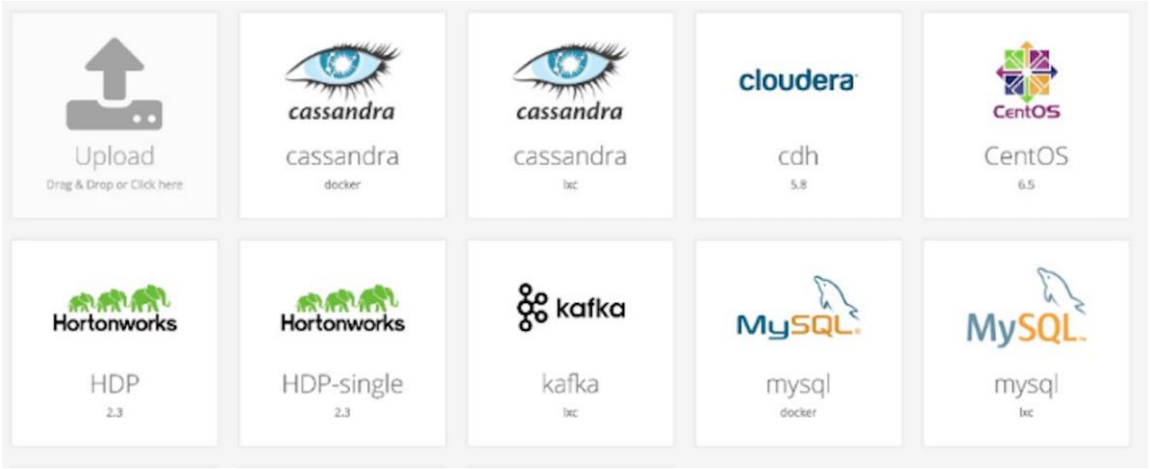
The specific advantages of Robin for organizations looking to simplify the use and administration of containers include:

- *Ease of Use* – Deploy applications in a few clicks, and create data volumes in less than a second.
- *Scalability* – Expand or shrink compute resources without service interruption or data movement.
- *Consistent Performance* – Consolidate database applications, Hadoop clusters, and other distributed applications onto a single platform without impacting performance.
- *Guaranteed QoS* – Ensure that applications receive designated guaranteed IOPS and consistent levels of CPU and memory, regardless of "noisy neighbors."
- *Reliability* – Recover instantly from underlying system failures.
- *Capacity* – Support more than 100,000 co-existing volumes with varying data protection needs.
- *Data Protection* – Assign stateful applications to protected volumes that use replication or erasure-coding.
- *Efficiency* – Provide storage protection at the storage layer rather than at the application layer, reducing storage requirements for data protection.

The key to Robin's technology is its Application-Aware Fabric Controller, which serves as the management layer for all application deployment and data movement. It controls and manages two primary assets, the Application-Aware Compute Plane and the Application-Aware Storage Plane, which virtualize the compute and storage separately, eliminating silos of capacity. The controller enables the intelligent provisioning of containers and storage based on individual application requirements as well as the topology of the environment, and it configures the application to make the best use of those components. The result is one-click deployments of complex, multi-container applications like Cassandra and Hadoop. Further, the Fabric Controller ensures that all applications get sufficient compute, storage, and network resources to meet user-defined quality of service requirements; the result is predictable performance for all applications. Since Robin controls the entire I/O path, it manages the priorities of read/write requests and so can provide guaranteed minimum and maximum levels of IOPs to ensure that application performance is maintained, despite any noisy neighbors.

Robin Systems also offers data lifecycle management through one-click granular snapshots and thin clones, which can be created in seconds. Unlike other implementations of clones and snapshots, Robin clones the entire application environment: the storage, the OS, the application configuration, and the topology. If something goes wrong in an application, the data can be rolled back to a known-working snapshot in seconds. To ensure high levels of availability, the Fabric Controller monitors the infrastructure and automatically recovers failed nodes and disks. With Robin's ability to seamlessly move workloads between servers, hardware can be used more efficiently, and less hardware is required in reserve for inevitable performance spikes.

Figure 3. Containerized Image Catalog



Source: Enterprise Strategy Group, 2017

Simplified Deployment and Protection of Complex Distributed Applications

The Robin UI includes a catalog screen that shows all of the containerized applications and data pipelines that have been configured for Robin, as shown in Figure 3. Images for these applications can be downloaded from Docker Hub or a private Docker repository directly into Robin. Linux Container (LXC) images require minimal one-time preparation before they can be used. Each image must also be accompanied by a manifest file that sets parameters for the image to be deployed.

To demonstrate the ease of deploying a complex containerized application with Robin, ESG Lab deployed a Cloudera instance that included multiple containers and ZooKeeper, a centralized service that maintains naming and configuration information, and manages several other services. We clicked on our selection of Cloudera in the Robin catalog, gave the new instance a name, reviewed some prearranged settings, and clicked the ***Create Application*** button. That was the end of ESG Lab's active involvement. The Robin Platform built a complex Cloudera configuration including 12 containers and dozens of storage resources. The entire application was created, configured, deployed, and ready for use without further interaction in just 24 minutes and 14 seconds. Table 1 compares estimated deployment times for several other commonly containerized applications.

Table 1. Theoretical Application Deployment Time Estimates With and Without Robin

Application	With Robin	Without Robin
Hadoop	< 30 Minutes	2 Days
Cassandra	5 Minutes	1-2 Days
Oracle	5 Minutes	2-3 Hours
Elasticsearch	1-2 Minutes	2-3 Hours
Postgres	Seconds	1 Hour
MySQL	Seconds	1 Hour

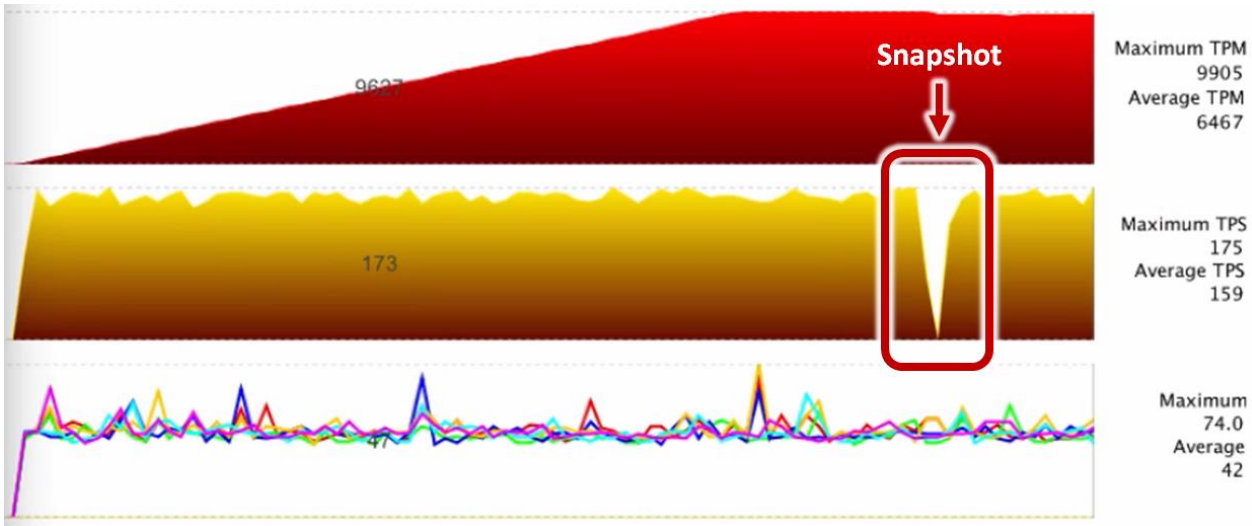
Throughout the process, we were able to monitor progress through the GUI and the command-line interface. Once the deployment completed, ESG Lab was able to connect to Cloudera's management portal and individual application consoles directly from the UI. Robin Systems' software enabled ESG Lab to take a process that typically requires hours or even days to complete, and execute it flawlessly in less than 25 minutes through its orchestration and automation.

Robin also provides a consistent failover mechanism for all applications, which improves application availability, and reduces operating expenses and software license costs. Many popular database applications require expensive additional licensing for HA and DR solutions, and dedicated standby servers and storage; Robin eliminates many of those expenses.

Once a containerized application has been deployed, it must be protected with backups. Robin offers transactionally consistent snapshots, which can be taken with a single click and cover all aspects of the container, including the OS, database schema, application configuration, and resource topology. Robin manages snapshots in a space-efficient manner, primarily consuming disk space to manage changed blocks. This complete snapshot supports Robin's Time Travel feature, which allows a container to be quickly and easily rolled back to a previous snapshot of that container. This can be done in place without the need to reapply logs or roll back database changes, even if there have been changes made to the database schema or the application's topology. A fully functional application clone can then be made from a snapshot in less than a minute with minimal storage overhead and no further impact to production.

To demonstrate the ease and speed of creating a snapshot and clone, ESG Lab leveraged an existing container with a preinstalled Oracle instance. We used Swingbench to generate OLTP traffic by adding additional rows to the database at a rate of about 160 transactions per second, brought up the snapshot interface, gave the snapshot a name, and then clicked the ***Create*** button. At this point, a brief, milliseconds-long interruption in workload performance occurred, as depicted in Figure 4. It should be noted that the interruption was imperceptible in the top graph, which displayed the transactions per minute performance metric. The application perceives this interruption as a temporary seven to ten millisecond increase in I/O wait, the result of quiescing the database just long enough to take a transactionally consistent snapshot. The bottom graph shows database response in milliseconds; note that during the snapshot, there is no apparent change in response time.

Figure 4. Impact of Snapshots with Robin Systems Containerization Platform



Source: Enterprise Strategy Group, 2017

After the snapshot completed, ESG Lab created a clone from it by clicking the ***Thin Clone*** button, giving the clone a name, and clicking the ***Create*** button. Less than ten seconds later, a new thin clone in a new container with its own IP address was ready to use. ESG Lab logged into the clone, started the database, and verified that it contained a fully functional copy of the original database. The entire snapshot and clone process took about three minutes to complete.

Why This Matters

Enterprises are eager to take the best advantage of their data resources, but the complexity of distributed applications often gets in the way. When organizations try to integrate containers into traditional storage models, complexities related to scale, agility, and traffic diversity come into play. Protecting container storage with snapshots and clones can require hours or days and a full-sized copy of the original container. Increasing the challenge, organizations seldom implement just one or two containerized applications—more often it's dozens or hundreds. Further, the amount of time and effort required to manually monitor and manage not just the containers, but also the underlying resources they consume, can quickly become too much for a team of system administrators.

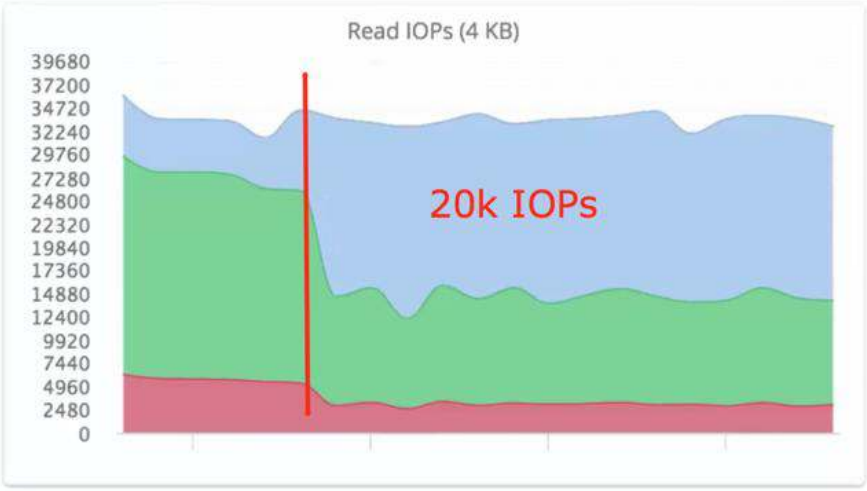
ESG Lab validated the simplicity of deploying and protecting a complex containerized application environment with the Robin Systems Containerization Platform. Robin leverages containers and provides a complete solution to enable enterprises to explore data without being gated by skill or resource availability. One-click deployment enabled deployment of a complete Cloudera infrastructure in less than 25 minutes, and efficient protection of the entire deployment via the snapshot functionality in less than a minute. In a just a few minutes more, ESG Lab created a ready-to-use, fully functional clone of the production environment. With Robin, organizations gain an efficient platform with automation and orchestration at its core to not only eliminate guesswork, which drastically improves IT productivity, but also enable faster time to value and reduce the burden placed on system administrators.

Quality of Service and Performance with Robin Systems

The Robin Systems Containerization Platform provides granular QoS control. The UI is equipped with sliders to enable an administrator to set the number of cores, amount of memory, and an IOPs threshold on a per-container basis. Each of these settings can be increased or decreased live, with all changes taking effect immediately and without any interruption to the running system. The only exception is memory allocation, which requires a reboot to decrease. Robin achieves this granular level of performance tuning through what it calls Application to Disk Performance Isolation. Robin ensures that each application gets the resources it needs to execute properly, all the way from the application through the server to the disk.

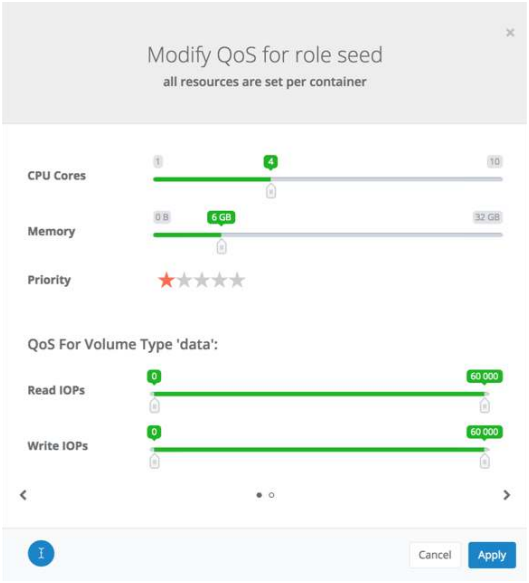
Granular performance tuning helps organizations avoid performance degradations common to complex IT environments. One example is a noisy neighbor scenario, which occurs when one containerized application on a server consumes so many resources that it pulls them from other applications on the same server, resulting in unpredictable performance. By setting minimums and maximums for each container's resources, IT departments no longer need to defend against unplanned performance spikes by overprovisioning hardware. This functionality also ensure predictability in performance regardless of the resources consumed by other applications. ESG Lab audited Robin's QoS functionality on a live system. Figure 5 depicts a graph that shows read IOPS for three color-coded applications over several seconds. The application represented in blue is Cassandra, which is initially struggling to obtain the I/O it requires.

Figure 5. Guaranteed Levels of IOPS



Source: Enterprise Strategy Group, 2017

Figure 6. QoS Adjustment Screen



Source: Enterprise Strategy Group, 2017

To resolve this issue, ESG Lab brought up the QoS modification screen, similar to the one shown in Figure 6, and adjusted the ***Read IOPs*** slider to ensure Cassandra received a minimum of 20,000 IOPS. That change, which occurred instantly and without interruption to the application, took place at the point in time marked by the vertical red line in Figure 5. No other action was required to achieve this performance change. The value of these QoS settings in a production environment is clear: users can have production, test, and development all sharing the same resource pool without concern for one application affecting another's performance, as long as they are being managed by Robin Systems' Application-Aware Fabric Controller.

QoS can be particularly valuable during maintenance or backup activities. When a compaction of tables is required on a Cassandra cluster, instead of sacrificing I/O assigned for regular operations, the administrator can temporarily increase the IOPS limits and ensure that the compaction can be completed without impacting the cluster performance.

With all the functionality that Robin adds, including snapshot management and elimination of the need to clean up after snapshots and rearranging blocks to increase contiguity ("garbage collection"), one might think that an environment running Robin would suffer significantly slower performance. Robin's stated goal is to impose no performance penalty for using its product when compared to a bare metal architecture.

ESG Lab audited a series of performance tests using EZFIO that baselined random, sequential, read, and write performance. Across block sizes ranging from 4Kb to 128Kb, most tests yielded little to no difference in performance between architectures. One exception was the small block (4Kb to 16Kb), multi-threaded read tests, where Robin actually outperformed the bare metal system by 20-58%.

Figure 7. Robin's YCSB Test Configuration

Three identical servers each with

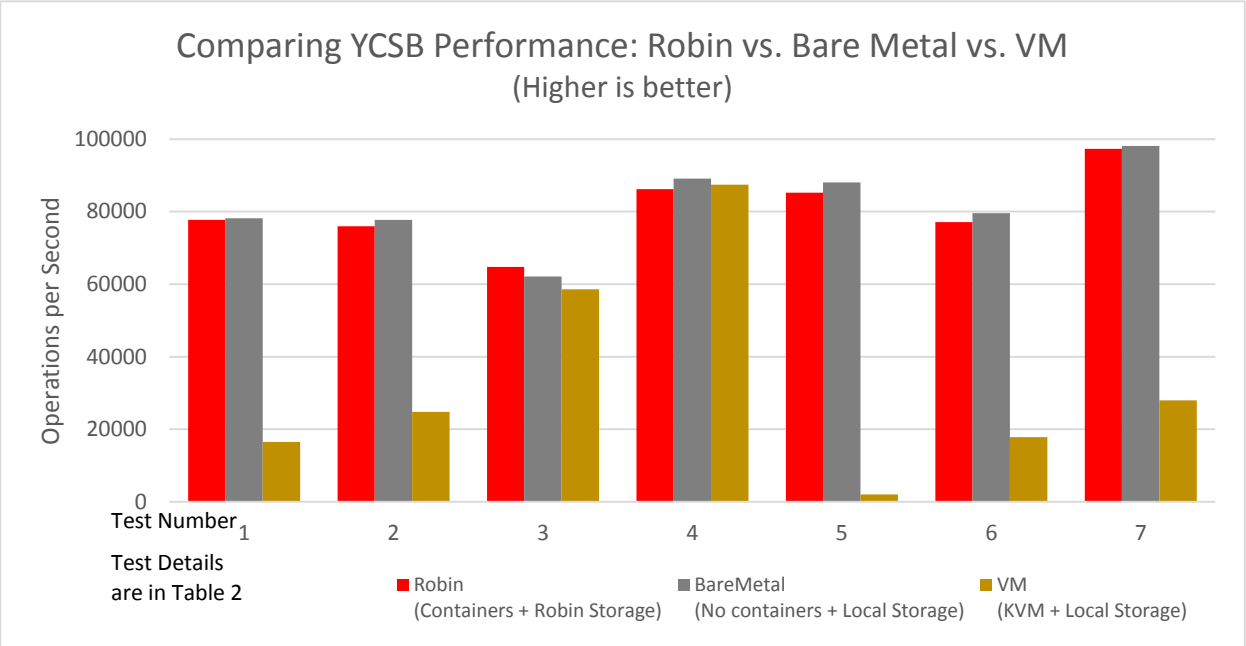
- 252GB of RAM
- 1.4TB on 5 SSD drives
- One 10Gb Network Card
- CentOS 7 (Linux kernel 3.10.0.327.36.2.el7.x86_64)
- Robin Software Version 3.0.0
- Cassandra Version 3.7

Source: Enterprise Strategy Group, 2017

Next, ESG Lab audited seven different executions of the Yahoo Cloud Serving Benchmark (YCSB) against three different configurations of a three-node Cassandra database: a container-based configuration with Robin storage, a bare metal configuration with local storage, and KVM using local storage. YCSB is an open-source framework developed at Yahoo! Labs for evaluating and comparing the performance of systems that serve data, including NoSQL databases like Cassandra. YCSB simulates a real world workload, as compared to EZFIO, which models a very specific data stream that is not as representative of real world scenarios. A description of the YCSB test hardware can be found in Figure 7.

The results of the seven sets of YCSB tests are shown in Figure 8 and Table 2, with a description of each test as either Load or Run; Load workloads spend most of their time loading data into the database, while Run workloads are running a more complex workloads of reads, writes, updates, scans, and inserts against the database.

Figure 8. Comparing YCSB Performance - Bare Metal vs. Robin



Source: Enterprise Strategy Group, 2017

The introduction of Robin into the test environment had a negligible impact to the overall performance of the underlying Cassandra database running on a traditional bare metal configuration. The worst case comparing Robin and bare metal demonstrated a slowdown of just 3.25%, while the best case showed a less than 1% improvement. Across each of these scenarios, the average penalty that Robin introduced was just 1.6%, well within what is considered noise. On load tests, Robin yielded an average performance gain of 4.5x when compared to the VM test bed, while run tests averaged a more than 2x improvement (excluding test 5, which proved to be an outlier). Details about the tests can be found in Table 2.

Table 2. Comparing Robin Performance in YCSB Benchmark vs Bare Metal and VM (in Operations per second; more is better)

Test Number	Test Details	Test Type	Robin (Containers & Robin)	Bare metal (No containers & Local Storage)	VM (KVM & Local Storage)
1	DB Updates, heavy workload (50/50 read/write mix)	Load	77,762	78,150	16,466
2	DB Updates, heavy workload (50/50 read/write mix)	Run	75,954	77,727	24,815
3	Read-mostly Workload (95%/5% read/write mix)	Run	60,701	62,074	58,592
4	Read Only (100% Read)	Run	86,217	89,114	87,462
5	Read Latest Workload (95%/5% read/update mix)	Run	85,264	88,049	2,000
6	Short Ranges Workload (95%/5% scan/insert mix)	Load	77,819	77,104	17,846
7	Read-Modify-Write (50/50 read/modify/write)	Run	97,353	98,087	27,999

Source: Enterprise Strategy Group, 2017

Why This Matters

Delivering consistent application performance is one of the most challenging aspects of running a complex containerized production environment, especially for databases and data-heavy applications. Applications are constantly fighting one another for limited, expensive resources, and the result is inconsistent performance that always seems to slow down at the most critical times. Manually reallocating resources is a complicated, time-consuming, and error-prone operation that can introduce significant downtime, even if the operation goes well. Organizations face a choice between the performance of bare metal and the convenience of virtualization. As a result, many organizations choose to suffer with inadequate performance or over-allocate expensive hardware resources to try to protect themselves against poor performance. Even with extra hardware, there are no guarantees.

ESG Lab validated Robin's ability to eliminate the trade-off and deliver the benefits of virtualization while maintaining bare-metal performance and guaranteed QoS. Robin allowed for the adjustment of QoS settings on a per-container basis on a live system without interrupting running applications. A live application struggling to perform correctly due to a lack of IOPS was retuned, and demonstrated an immediate performance improvement. Live, real-time performance tuning makes it possible for organizations to respond quickly to performance issues by reallocating expensive resources, and it means that they don't have to overbuild their systems just in case performance issues emerge later. ESG Lab also validated that users can be confident that introducing Robin to a previously bare metal environment has little to no impact in application performance as witnessed by real-world performance test results of a Cassandra environment.

The Bigger Truth

Containers and containerized applications continue to gain acceptance in part because they help organizations with the traditionally difficult choice between the ease of use and unpredictable performance that VMs bring and the complex management but more reliable performance of bare metal. More and more organizations are standardizing on container-based architectures to simplify the deployment and management of applications, including the migration of those applications from dev/test environments to production. Time-to-market is reduced because applications can migrate with their complete environment; the complex effort of ensuring alignment of the right libraries and other dependences is avoided since containers package all of those components into a single bundle.

The problem is that those bundles aren't as complete as the organization may think they are, and they lack isolation from other containers that rely on shared underlying compute and storage resources. Deployments can be complex, requiring coordination of many different components and resources. Once they are deployed, it is challenging to back them up without causing an impact to production applications. Making an independent copy of a production system for testing or development can take days to complete. Containers can easily interfere with one another when they are forced to share storage and compute resources.

Robin Systems has developed a solution to improve the way containerized applications work in the real world. The company has taken a key feature of virtualization—the ability to decouple shared pools of compute and storage—and applied it to containers. By introducing Robin into a containerized environment, organizations gain one-click deployment of complex, multi-part applications, which can dramatically improve time to market and reduce administrator overhead. They can protect those applications and their data with milliseconds-long and effectively interruption-free coordinated snapshots that can be restored quickly and easily. Developers can clone an entire environment, including the application, data, schemas, and application topology from a snapshot in minutes without requiring administrator involvement. Application uptime can be improved via automatic fault tolerance for the entire application stack. Finally, performance issues are avoided because administrators can set resource minimums and maximums to guarantee quality of service for each containerized application.

As organizations look to improve time to market and agility, containers become an increasingly important piece of their strategy. If you are looking for a way to overcome some of the challenges of containers, thereby speeding time to value and easing the burden on system administrators, ESG Lab recommends taking a look at Robin Systems Containerization Platform.