Reinventing Monitoring for the Private and Hybrid Cloud

Bernd Harzog CEO, OpsDataStore

Monitoring of systems, networks and applications has been around since the inception of computing. However, in the last 5 years there have been dramatic changes and innovations in the computing, networking, and applications development landscapes. These dramatic changes are:

- Infrastructure is shared and dynamically allocated – often by continuous automation. Server virtualization by VMware and DRS (Dynamic Resource Scheduling) caused virtual servers to move around automatically between hosts based upon CPU and memory constraints. Automation vendors like VMTurbo and Cirba have taken this one step further by incorporating constraints like networking and storage congestion, as well as allowing for the prioritization of workloads.
- Virtualization is now extended to the network and storage. In some cases virtual networks are established in software between every two things that have to talk to each other using products like VMware NSX. Storage is being virtualized so that there is only a faint resemblance between the storage layer accessed by the operating system and the applications, and the actual physical disks or flash arrays.
- There are ever more layers of abstraction between the actual application and the physical hardware. These might include Docker, the JVM, a PaaS stack like Cloud Foundry, and the layers of virtualization listed above.
- The application layer is now characterized by both diversity and rapid change. It is not just a Java world any more with other languages being popular and new ones arriving every day. The pressure on the part of the business for more functionality to be put into production more quickly has led to both agile development and DevOps – both of which result in the applications being modified in production more frequently.
- Private and Hybrid clouds lead to layers of infrastructure, layers of management and teams, and layers of tools. This creates the same problem that siloes of management created in the past. Each of the IaaS, PaaS, and SaaS layers in the cloud architecture have their own teams to manage them and their own tools to manage that layer.

This layer of technologies, products, and teams is shown in the diagram below. The tools to monitor these layers are:

- For the IaaS layer – storage tools, SAN tools, networking tools, server tools, virtualization (hypervisor tools), and tools to monitor the wide area network.
- For the PaaS layer – tools to monitor all of the software services that support the applications which include database servers, web servers, Java application servers, PaaS frameworks and of course now Docker.
- For the SaaS layer – tools to monitor the applications themselves which include Application Performance Management products, as well as tools to monitor actual end user experience.
There several core problems with these layers of tools that must be addressed in order for Private Clouds to be successful:

- The data from each of these tools at each of these layers is not shared with the owners or operators of the other layers of the private cloud stack
- A consolidated set of data or metrics is not available to business constituents or users of the private cloud giving them no end-to-end visibility into the behavior of their applications and the workloads that support them
- When there are performance problems (as there always are), the layering of the toolset and lack of collaboration across the teams that own the three layers make problem resolution a time consuming and frustrating process
- The lack of end-to-end visibility to the application owner of how their application is performing, and how it is operating in the shared infrastructure that supports it impedes adoption of the private cloud by application owners. People who live in houses simply do not trust apartments unless they know who their neighbors are and how they are going to behave.
- Despite the vast amount of monitoring data that is being collected by all of these tools, in many cases the data necessary to ensure that the private cloud is actually working correctly for its hosted applications and business constituents is not be collected at all or in the required manner.
The New Types of Monitoring Data Needed

Highly dynamic and interactive systems a new approach to collecting and processing monitoring data. The following concepts must be incorporated in a modern monitoring data pipeline:

- **Streams of Metrics** – A modern online application system produces streams of metrics starting with the response time (user experience) data, and including data about how all layers of the software and hardware that support that interaction are behaving.

- **Streams of Relationships** – In a modern online application system, a transaction of interest is related to the Java virtual machine in which it runs, the operating system upon which the JVM runs, the virtualized hardware where the operating system runs and then the entire virtualized and physical infrastructure (down to the spindle on the hard disk) that support that transaction.

- **Streams of State** – Most enterprises have Configuration Management Databases (CMDB’s). These are supposed to store the current state of the entire software and hardware environment of the enterprise. But now online systems change too rapidly for CMDB’s to be kept up to date – relegating them to the junkyard of worse than useless legacy technologies. Private clouds and their modern applications are simply too dynamic for the CMDB and the CMDB must be replaced with something that is automatically and continuously kept up to date.

New Data Collection Approaches Needed

The existing approaches to big data, respectively focused upon log analysis and batch processing of business data with technologies like Hadoop are woefully inadequate when it comes to being able to process streams of metrics, streams of relationships, and streams of state in real time. Process in these cases means the ability to ingest these streams as they arrive, continuously perform operations on these streams to add value to the data, and then transform the data and the relationships into forms useful to people using market leading query and visualization tools.

This leads to the need for the following new approaches to the collection of monitoring data:

- **Data “Pushed” not “Pulled”** – Today most of the data that is “collected” into big data back ends is collected by having someone or something query the data from its source. Smart phones broke this paradigm as it is impossible to query billions of smart phones and ask them for their data. In this new world it must become the responsibility of each end device to provide or push its data into the new real time back end for these metrics, relationships and state. All systems management software that relies upon querying things for data is now legacy software unsuited for the modern world. The modern data collection paradigm needs to be based upon streams of data pushed from each device via a message bus like Kafka to a back end that is capable of ingesting them all in real time and processing them all in real time.

- **Data Collection In Real-Time** – Legacy systems management products collect data at best every 5 minutes and many collect data as infrequently as hourly. This leaves too much time in between when something bad happens and the system knows about it. Modern data collection needs to be real-time and continuous.

- **Comprehensive Data Collection** – Because legacy management systems cannot deal with volumes of data, they pursue “sparse” approaches to data collection that sample and that fail to collect data comprehensively across all of the aspects of the software and hardware.
systems that support an end user, a device or an application. Modern data collection needs to comprehensively collect data from every layer of the hardware and software ecosystem that supports an interaction or a transaction.

- **Open Multi-Vendor Data Collection** – The diversity in the sources of new management data is too great for any single vendor to stand a chance of collecting them all or even of collecting all of the metrics, relationships and states the pertain to one set of interactions or transaction. The pace of innovation in this industry is simply too fast for any one vendor to be able to keep up. Therefore only an approach that recognizes that there will be many sources of data and many vendors who specialize in collecting various types of data will succeed.

- **Relationships Established at Ingest Time** – It is impossible to know ahead of time, and to be able to plan ahead of time for how future streams of data will be related to existing streams of data. Therefore these relationships must be established at the time that each new stream of metrics and state are added to the system. All previous attempts to pre-define a model of an environment, like the Common Information Model (CIM) of computing are now invalid since anything defined by a committee cannot keep up with the pace of innovation in these new environments.

- **No More Extract, Transform and Load (ETL)** – Data needs to arrive in real time (no extract) and be stored in a useful form on a continuous basis (no more Load). Instead streaming ingest, coupled with continuous and real time transformation, and streaming writes of the resulting useful data needs to replace ETL.

- **A Real-Time Data Store** – all of this data and the relationships between these streams of data need to be stored in a real time data store that can keep up with the ingest rates from these new environments and crucially, make this data instantly and continuously available to modern analytics and BI tools like Tableau and Qlik and modern real time visualization systems like Grafana.

**The Crucial Metrics and Their Sources**

The single most broken thing about monitoring is the mistaken notion that monitoring for resource utilization across the set of devices and stack of software that comprises an application or a private cloud can ensure acceptable service quality.

To fix this misconception we have to redefine performance as not having available resources, but rather in terms of how long it is taking to get work done (response time and latency) and how much work is getting done per unit of time (throughput). Response time, throughput, and of course errors need to become the crucial metrics by which service quality is judged and delivered.

If we accept that response time, throughput, and errors are the crucial categories of metrics the next steps are to identify what these metrics are at each layer of the private cloud stack and then identify the sources of those metrics. The most important thing to realize is that the standard interfaces that exist at each layer of the stack (SMIS, SNMP, Netflow, JMX, etc.) do not measure response time, throughput and errors and that we have to rely upon vendors who have invested heavily in the custom instrumentation required to collect these metrics.

We also have to recognize that the investment in collecting these metrics at a particular layer is significant as is the level of technical expertise required to collect these metrics more than what can be assembled by an single company.
Finally, we have to recognize that the industry is changing too quickly for any single vendor to be able to cover the waterfront or the entire cloud stack. Network Virtualization, Storage Virtualization and Containers (Docker) are just the most recent examples of innovations that require advances and innovations in instrumentation and these advances are going to continue to occur at an accelerating pace.

For the reasons listed above the only monitoring strategy that can work for the private/hybrid cloud is a focus upon best of breed vendors who collect the required data at a layer in the cloud stack, and to then feed these metrics into a common big data back end for processing or analysis. A diagram that depicts this monitoring architecture is below.

**The Monitoring Architecture for the Private/Hybrid Cloud**

![Monitoring Architecture Diagram]

**Summary**

Both Private and Hybrid Clouds require a completely different approach to collecting data, processing data, and making this data useful to users and analysts. Real-time streaming combined with continuous transformation needs to replace existing batch processes. Single vendor approaches needs to be replaced by an ecosystem of vendors that can collectively keep pace with the pace of innovation backed by a high performance big data back end. Enterprise IT Operations organizations must adopt these modern approaches to managing their private clouds or else those clouds will not be cost competitive and functionally competitive with public cloud offerings leading to ever more outsourcing of workloads to public cloud providers.

**About Bernd Harzog**

Bernd Harzog is the CEO and Founder of OpsDataStore the real-time big data back end for all IT Operations Management data and vendors. OpsDataStore’s open big data back end consumes and relates data from multiple sources and immediately makes that data useful to decision makers using market leading BI and visualization tools. Learn more at [www.opsdatastore.com](http://www.opsdatastore.com).