It all starts with the data

Web Scale IT in the Enterprise

Welcome

Enterprise IT organizations are turning to a portfolio of SaaS, PaaS and hybrid cloud services to quickly respond to the business at a lower price point while improving service levels. Yet a new level of transparency and data precision is required to effectively deliver on this promise. Only AppFirst delivers a miss nothing approach to collecting data across the entire IT ecosystem. With this foundational data set it is now possible for organizations to achieve:

1. A steady-state high-definition view of your end-to-end application infrastructure as compared to the still-picture model that was state-of-the-art with legacy technologies.

2. Significantly drive down costs while ensuring business critical service levels.

3. A deterministic view of the n-tier application environment, eliminating the need to make probabilistic assumptions to manage the system.

4. The correct data view to fully leverage automation and predictive analytics technologies.

This newsletter discusses the key learnings that Claus Moldt, former Global CIO for Salesforce.com took away from addressing these challenges. The need for a new approach is reiterated by the following Gartner research note on Building a Modern APM Architecture for the World of Web Scale IT. If you are looking to solve these challenges, come see how the experts at AppFirst have helped enterprise organizations achieve their vision.
**Q&A With Claus Moldt, Former Global CIO for SalesForce.com and David Roth, CEO of AppFirst**

**David Roth:** What does the concept of web-scale IT mean to you?

**Claus Moldt:** At a high level, it means you're not only running an infrastructure that is targeted to a few users, but you're actually running an infrastructure that enables you to scale to millions of users. That means building in enough logic, automation, management, monitoring, etc. to map the infrastructure as you scale, without accruing a ton of headcount. Amongst other things, you need to have access to metrics that help you understand capacity, monitor the infrastructure when needed, make autocorrections from failures detected, and ensure the system is up and running 24/7. When you build the infrastructure, all of these things need to be more or less automated and built as part of the way you think of the infrastructure. You can't just build a house without the infrastructure, or it becomes impossible to manage. You actually have to create a framework allowing you to reach that web-scale.

**David:** In your experience at SalesForce and eBay, what was your approach in achieving web-scale IT?

**Claus:** It's interesting because with both eBay and SalesForce, I started fairly early on in both companies. When you're starting off, you don't necessarily think about what it will look like when you're ten, twenty or thirty times bigger! You typically only start considering scale when you hit your first issue. A lot of people have the experience of saying “what are we going to do now?” I think that has changed over the last couple of years as the infrastructures and frameworks have matured, along with more cloud players as well. While it's still not where it needs to be, there are certainly frameworks out there that you can use.

First and foremost, it was important to get an understanding of where and how we needed to scale. In order to do that, we set key indicators that were defined by a series of metrics. This told us when we needed to add capacity to the infrastructure or what has gone haywire within the environment.

“A solid set of metrics that act as key indicators for how your system is performing holds the key to managing these web-scale enterprises.”

For example, if I just released code - something we saw very often both at eBay and SalesForce - we assume from the get-go that the code could have some function impact to the systems. If you have a small-scale system and you see a 5% impact - that may not be so bad. But if you see a 5% impact across 10,000 servers, then it's a substantial amount of consumption that is taken away from the system just because this code was released. At this scale, you're impacting potentially millions of dollars of revenue due to this issue.

Three things to consider:

1. You need to figure out what a new release means to you. You need to figure out if this is strictly code-related, or if this is something that you will continue to see because of the newly added features and therefore should assume that it's going to stay like this going forward.

2. Complete data collection across every layer of the technology stack is required to see where in the system things are not performing to expectations. You need to understand how each web server, database server, application, network, etc. is performing in order to identify where the issue actually lies.

3. Time to identification - you need access to normalized and correlated data immediately; not in days or weeks.

“Building an infrastructure with the ability to scale to not just a few users, but to potentially millions of users”
David: Why do metrics matter?

Claus: One of the things that I always used to say is “We want to make sure that we manage the system, and the system doesn’t manage us.” And the only way that we can do that is if we have metrics established - that’s the only way that you can truly understand your system. As you learn more about the system, you may change one or two of the key metrics over time, but you have to start by establishing what you need and then adapt over time.

David: Can you talk about the difference between polling vs. continuous collection?

Claus: I’ll give you an example from eBay, where it was really hard to pinpoint an issue. Pretty much everybody has run into an issue, and called the vendor who then says, “We want you to do more debugging to get more information.” I can’t tell you how many times it’s happened in my career. Getting that kind of information that late in the game is too late in terms of customer impact.

It’s difficult to go back and try to track down information after the end user has discovered the issue. So if the issue occurs a second time, and the vendor tells you they need to dig around a bit more to see what is happening in the OS or in the database, they are just on a continuous quest to find out what the problem is while enabling wider and wider margins. That is far from the ideal scenario. We need to identify problems before they are seen by the users to avoid business downtime and unnecessary support costs.

At eBay, the solution that we ended up with was to try and poll as many things as possible, while also having the ability to increase the frequency of metrics collection when an issue was detected. However, polling was difficult because we only knew that there was the potential for something to go wrong, but we didn’t know where. There are two things that I think are most important:

1. Have enough collection in real-time across as many entities as possible, without impacting your overall system performance.

With the amount of data to comb through and look at you need to make sure that you get access to that information in a timely fashion.

At SalesForce, the first thing that I recommended was implementing a system that would go in and monitor metrics - we had about 8-10 key indicators of how the systems were performing. It was the number of requests on the backend system per second, the utilization that we had on various tiers, etc. and those eight components gave us an initial indication of whether the system was performing well or not. If any of those indicators went red - that’s when we had the ability to go into the next level of metrics and look at what was actually happening in our system. So relevant actionable metrics - making sure that you have the right tools to manage these web scale applications.

“We want to make sure that we manage the system, and the system doesn’t manage us.”
So if you have a very light framework in place that doesn’t consume a lot of system resources (i.e. less than 3%), then you can get enough metrics to have a good idea of what’s going on. Of course, you should become very skeptical if this framework takes 10% or 20% of system resources because like I said earlier, that number becomes significant with 1,000 devices. Polling is not ideal because if the system is already exhausted from a resource standpoint, I can almost guarantee you that with polling you may not be catching the data that you need.

2. Have enough metrics collected to connect an event that’s happening in one area of the system to the impact it has on another area. Only recently has it even been possible to implement a system like AppFirst that will collect data across the board.

It’s hard to do that without having a pretty significant amount of data about the system. Moreover, data from a specific area doesn’t necessarily tell you what the impact is in another area.

David: What strategy would you recommend to someone who is migrating to a public cloud, private cloud or a virtual environment in order to properly size their application for the new environment?

Claus: There are two scenarios that need to be considered when sizing applications:

1. You have a completely new application and you don’t have an understanding of any key metrics.

You should get a rough understanding of what you need in terms of consumption for whatever application you’re deploying. For example:

   a. How many users do you think there will be?
   b. What do you expect them to do?

2. You already have an application on-prem that you’re migrating to the cloud

This is very different because you should already have a fair amount of metrics. If you don’t already have them, I would highly recommend that you use a technology like AppFirst to establish foundational, miss nothing-metric collection across your environment in order to understand your application footprint. Your footprint will provide you with insight into exactly how an application is distributed across its enterprise assets and how it successfully runs before introducing new variables. This allows you to successfully move to new environments, conduct instant root cause analysis and resolve issues as needed.

David: There are a lot of discussions around hybrid application deployment. What challenges do you see stemming from this?

Claus: There are a couple things:

1. You need to make sure you understand the tradeoff between the cost and efficiency of the cloud with the control offered by physical environments.

2. You want to make sure that you still have insight into issues and understand metrics the same way you would if you had ownership solely on-premise.

For example, there are some providers, that will say “don’t worry, we’ll take care of everything.” They don’t allow you to install performance tools within the environment. However, a lot of customers set up an external set of synthetic transactions so they can see those metrics. Even though they didn’t do it at the backend, SalesForce customers were craving some insight into what was going on in the system.

I believe that it is the operations team’s responsibility to understand how the components of the system work together and should have the ability to collect metrics. Having deployments in the cloud is not an excuse to not have this understanding.

David: We hear more enterprises are moving towards an IT as a Service model. Is this hype or are the benefits real?
Claus: In my mind, IT as a Service means IT is no longer a central control group that has to approve everything going on. IT should be establishing best practices and policies in an environment where services enable the business to move fast. To do so, IT needs a minimum understanding of how the business is going to grow and then provide tools to the business to enable this. Only with a detailed set of metrics can you get an idea of costs, consumption and usages.

However, I don’t think that’s the way that most IT departments think about it. They think about building applications and controlling what technology, systems and devices are used.

Modern IT as a Service should provide a framework for allowing business to move fast and not have to wait for IT to deploy systems for new applications, etc. They should build Amazon-like infrastructures with established policies and best practices where people can deploy and go. At the same time IT still retains the responsibility to not throw hardware assets at problems unrelated to application scale, which is why rich system metrics are critical in curtailing unnecessary scaling costs. To make IT as a Service economically feasible it must be controlled the same way you would manage local systems sprawl. If a business needs to be open to growth, dynamic and responsive - IT as a Service is not just hype for those that understand how to deploy and manage it.

David: If you’re moving to the IT as a service model, what risk do you run if you’re not collecting the right data?

Claus: Without the right metrics collection, you’re not going to have all the key points, which includes system metrics as well as business metrics. I don’t think you can separate the two. Most of my system metrics were actually used to calculate my budgets based on what I’ve seen for the last 2-3 years. I can get a very accurate overview of what my budget should be in the following year based on my systems metrics.
David: Any other key thoughts you want to leave with IT leaders?

Claus: One thing I want to talk about is how people think about dashboards. It’s one thing to establish metrics, but it’s important to make them available to everyone. They can’t only belong to the Head of Operations or IT - they need to be everybody’s metrics. If I establish a metric that doesn’t make sense for the people doing the day-to-day work, then that isn’t useful to anyone. If you have to break down the top metric into sub-metrics. Establishing a good dashboard is essential to getting the right metrics, and that is ultimately how you achieve large web-scale IT.

It is critical to build these dashboards with a foundational, miss nothing approach to data collection to ensure decisions are made based on a deterministic view of the system. Partial data leads to decisions based on assumptions, which is no longer required with a complete foundational dataset.

Source: AppFirst

To Accomplish this

- IT needs insight into the business and future growth plans.
- Build Amazon-like infrastructure with policies and best practices where users can deploy and go.
- A detailed set of metrics is required to get an idea of costs, consumption and usage.
- Data quality and completeness is critical in successfully matching operations to business execution.
From the Gartner Files:
Building a Modern APM Architecture for the World of Web-Scale IT

Today’s application (and enterprise) monitoring tools increasingly fail to meet the needs of Web-scale application environments that IT operations teams are deploying to match the needs of modern agile applications. Understand the pitfalls so you do not mistakenly invest in obsolete technology.

Key Challenges
- Traditional application and enterprise monitoring systems do not scale proportionally when resources are added to address the needs of Web-scale environments.
- Web-scale applications’ need for elasticity introduces volatility while users require responsiveness — this places added pressure on systems that are designed to monitor these applications.
- The data collected by many conventional monitoring solutions lacks the granularity and contextual enrichment needed to improve Web-scale decision making.
- Many standard application performance monitoring (APM) tools lack the ability to be part of a lightweight management toolchain.

Recommendations
- IT operations teams need to select Web-scale APM solutions that can grow at the same rate as the systems they monitor without incurring disproportional operations costs.
- When evaluating Web-scale APM alternatives, IT operations and application support teams must prioritize offerings that are in congruence with the characteristics of the systems they are monitoring.
- Application and operations teams must choose Web-scale APM tools that provide enhanced data enrichment and presentation capabilities.
- IT operations and application support teams need to narrow their selection of Web-scale APM offerings to those products that can integrate within a management toolchain.

Analysis
Design Assumptions Have Changed: Ensure Your APM Technologies Look More Like the Applications That They Manage
What are Web-scale applications? They are critical to define in order to provide the needed context to the statement above. Often, the definitions center around the architecture of the application — that is, incorporating elements of statelessness, asynchronous communications and resiliency, often in the form of single-page applications, which often use technologies such as NoSQL databases leveraging eventual consistency concepts. These applications are consumed in an increasing number of methods, including mobile and those without GUIs for consumption within other applications. But while important to consider, the technology elements don’t provide all of the causes for a change in your APM monitoring approach.

Dr. Werner Vogels, vice president and CTO of Amazon, has said that scalable systems are systems where, if we increase the resources, it results in increased performance (that is, units of work) in a manner proportional...
to the resources that are added (see Note 1). Another way to look at this is that scalability suggests that the latency should remain constant while overall throughput improves in relation to the resource added when scaling horizontally to maintain efficiency — as the systems do within the cloud, where the Web-scale IT concept originated. But he also said that resources are sometimes added with a goal to improving resiliency and that, in the process, this should also not harm performance — which it often does with traditional applications and systems.

In addition, Vogels states that scalable systems (including Web-scalable systems) also need to be operationally efficient — that is, that as you add more nodes to the system, it shouldn’t require more operators because they are still performing the same tasks. Ideally, these resources should be homogeneous, but what might be more important is a focus on automation and self-management, things traditional enterprise applications and infrastructures often fail to implement fully with today’s automation tools and approaches. He goes on to say that scalable systems have to be resilient (Gartner would say that today’s scalable systems need to be anti-fragile) with regard to all kinds of variances introduced to the environment and applications. The system is never stable — failure occurs often in Web-scale systems, and thus, applications need to be able to withstand these impacts without impacting the customer or increasing operational costs. Some Web-scale IT organizations forcefully introduce failures to ensure the system is operating in an automated manner. Finally, Dr. Vogels says that the most important feature of a scalable system is that, as these systems grow, costs should not increase at the same rate.

It is Gartner’s opinion that the major reason why traditional application and other monitoring tools don’t meet the needs of most Web-scale environments is that they almost all violate the conditions for scalability set forth by Vogels. Because they are largely inconsistent with the design goals of scalable systems, they would make a poor choice as support tools within a Web-scale context because, if for no other reason, costs would likely rise at a higher rate than the growth of the system they are intended to support. Newer, SaaS-based tools may be a better fit in this regard, but while often built upon an internal Web-scale architecture, not all of their benefits are passed onto customers (such as customization for fit). However, there are some other considerations that potentially sound the death knell for traditional APM and enterprise monitoring tools.

Initial Checklist: Ensure Your APM Technologies Support a New Degree of Dynamism

Web-scale applications have several characteristics that are impacting how we approach data collection, analysis and presentation. These include:

- **Size:** Reside in clusters with large numbers of instances from tens to tens of thousands
- **Volvatility:** Exhibits planned and unplanned instance as well as workload volatility
- **Responsiveness:** Consumers have very limited tolerance for poorly performing systems

Web-scale applications vary in size, although in the common implementations within large cloud services, the number of instances often runs into the thousands. And within each node, there may be thousands of metrics available. A common DevOps/Web-scale IT mantra is to collect data on everything because the dynamic nature of the applications makes it especially difficult to know what metrics to specifically monitor in advance. (Note: While this method is often employed, there may be costs associated with overhead, processing and storage, so care should be used with this approach.) When calculating a potential metric stream using 1-second collection intervals for 100 attribute/value pairs for each of 1,000 servers, more than 100,000 metrics per second would need to be supported. This represents the collection support activity only, not the need to aggregate and perform analytical heuristics against this dataset.

Web-scale applications have highly variant “lifetimes” and workloads. Because these instances typically lack state, it is often easier to discard and start up a new instance when there is an issue in the offending application’s environment. The same can be said during code releases; maintaining state on a specific system is a challenge because applications are often layers of releases and patches. However, to perform an effective postmortem, the monitoring data needs to still be available. This sets up the need for frequent (and granular) data collection in order to understand deviations.
in the environment, thus enabling responses to issues in a rapid manner.

Not causing any increase in monitored application latency is also critical, and there are Web-scale APM architectural changes that are necessary to potentially mitigate this. Most importantly is the need to have separation of components in a pipeline manner. Collection, storage, analysis and notifications must be performed in different threads across separate machines. For example, the “check” and analysis logic should not be in the same component or feature if the monitoring agent is running on the target node. A byproduct of this separation of concerns is that it also lends an element of resiliency while allowing for function-independent scaling.

Make the Data Meaningful: Move Away From Thresholds and Focus on Visualization

The use of static, linear thresholds is something that infrastructure and operations (I&O) organizations have been moving away from for quite some time. More advanced organizations have implemented automated baselining of metrics, and while these continue to have value, they don’t provide a time element that is critical to many of the classic Web-scale IT environments. Understanding (a forward-looking) rate of change within a metric is a more meaningful indicator of a potential issue than just analyzing reactive threshold breaches. Additionally, analyzing singular metric instances often presents limited clarity — especially when operating against large datasets. Thus, it’s important to consider aggregating metrics that result in a new set of synthesized measurements to provide a better sense of the informational context.

Another method to get “better” value from data is information presentation, which is an often-overlooked part of the APM and broader enterprise management industry. Data hierarchies are often represented by tree diagrams using node-link layouts — the classical Windows Explorer interface is an example of this. But they have limitations in terms of information conveyance — especially in Web-scale environments. Newer tree styles such as tree maps and circle-packing diagrams are thought to improve operational sense making because they consolidate large amounts of information in easier to discern geometric shapes. Another way to say this is that they can provide greater salience or make it easier to visually notice important data.

Combine Open-Source Monitoring Technologies With APM to Meet Web-Scale Challenges

APM tools today are focused on traditional needs and use cases, and this forces many users to pair these tools with open-source products such as StatsD, collectd, Graphite, Ganglia and several other open-source collection, storage and graphing products. These tools are often difficult to implement, maintain and scale. Another approach is using commercial products that support collection technologies available in the open-source community but create robust monitoring and collection capabilities such as the hybrid (SaaS or on-premises) products offered by AppFirst and Circonus, as well as SaaS-based products from DataDog, Librato and Server Density.

Another key consideration is that the Web-scale APM solution must be part of an overall management toolchain. Most importantly, it needs to have synergy with configuration management tools to know when changes are taking place and to be able to help correlate those changes to any degradations in performance.

Finally, issues in production must be detected more quickly, often by leveraging APM or custom and dynamically adjustable instrumentation of performance and code quality. This enables teams to understand the impact of a release (“Did I make things faster and better?”) and to take decisive action on these findings to remediate or determine if additional work on a feature is required. Without traditional testing patterns, due to the time between releases, the visibility into production is paramount.
Increasingly, Gartner clients are doing similar collection and adding analysis using IT operations analytics (ITOA) technologies. Many ITOA products handle both structured and unstructured data, facilitating metric analysis with a high degree of granularity, as well as the ability to extract information from log files. These solutions include vendors such as Splunk and Sumo Logic.

APM offerings such as AppDynamics and New Relic pair both APM and metric collection and include a large array of plug-ins supporting standard technologies. Unfortunately, they do not pair this data together strategically within their views, but this will change over time as buyers prefer to have APM and metric collection in a single tool.

Boundary offers network performance monitoring technologies, but they run from server-installed agents rather than by centralized traffic analysis. Boundary’s integrations are tighter in overlaying metric data and flow data over the network topology, but have fewer partnerships and developed metric collections than other APM providers.

Gartner predicts that fewer than 25% of conventional APM products will support pairing high-frequency custom metric collection and application context needed to facilitate Web-scale monitoring by 2016, up from 10% today.

Monitoring technologies that contain the traits required to efficiently operate at Web-scale are built and sold by smaller startups, many of whom have seen substantial growth within the last two years. The demand for specific patterns and characteristics in Web-scale environments is in its infancy within typical IT environments, but there is no question the trends toward operating in a similar manner to those large Web-scale companies will force more technology providers to adopt these traits. The typical situation of startups being acquired is likely to persist, but there is also a growing trend that will include the changing of the old guard (namely the Big 4) as more nimble best-of-breed technologies continue to capture mind share and market share. These nimble players are better able to embrace and extend their technologies to meet the new application and environmental demands, including new languages, new ways to consume applications (such as mobile or wearable technology) and changes within the operations environment (such as big data technologies).

---

**Note 1. Availability and Consistency**

See the presentation “Availability & Consistency.” In it, Vogels specifically calls out the characteristics of scalable systems: autonomy, asynchrony, controlled concurrency, controlled parallelism, decentralize, decompose into small and well-understood building blocks, failure tolerant, local responsibility, recovery built-in, simplicity and symmetry.

Source: Gartner Research, G00257849, Jonah Kowall Cameron Haight, 6 December 2013
## Collecting Metrics for Web Scale Applications

<table>
<thead>
<tr>
<th>Operation Requirements</th>
<th>The Goal</th>
<th>AppFirst Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miss nothing data collection across the entire technology stack.</td>
<td>• Automatically discover and expose all components across the systems that are not performing to expectations.</td>
<td>Collection provides complete and consistent coverage of the full end-to-end system; gathering data from every language, O/S interaction and native/packaged application component without the need to instrument code or install multiple agents.</td>
</tr>
<tr>
<td>Provide immediate access to a normalized and correlated dataset across the n-tier application.</td>
<td>• Reduce time to resolution from days/weeks/months to minutes.</td>
<td>Access terabytes of normalized event data that is aggregated across the system from a comprehensive backend.</td>
</tr>
<tr>
<td>Deliver a complete application footprint to determine how an application is distributed and consumes resources across all enterprise assets.</td>
<td>• Understand baseline behavior and asset requirements to successfully migrate and/or install applications to new environments: physical, virtual, cloud or hybrid.</td>
<td>Complete n-tier data collection of an applications’ interactions with all asset classes across the ecosystem. This includes a baseline of resource consumption and communication over the network.</td>
</tr>
<tr>
<td>Enablement of operational automation for material cost reductions.</td>
<td>• Resolve issue before users experience errors. • Drive down operational costs. • Reduce human error.</td>
<td>An unbiased approach to operational data collection enables deterministic automation for troubleshooting, scale and recovery.</td>
</tr>
<tr>
<td>Collect business metrics to overlay on top of granular performance metrics.</td>
<td>• Transparency into IT's alignment with the business.</td>
<td>Report on and correlate business metrics with all the components that make up a service, including latency, transaction counts and external reliances.</td>
</tr>
<tr>
<td>The fidelity of data fed into dashboards provides a holistic and deterministic view of your n-tier architecture.</td>
<td>• A comprehensive dashboard for a single source of truth into the performance across the entire ecosystem.</td>
<td>Review role-specific dashboards with all data fed from a unified source eliminating manual data correlation and misaligned datasets.</td>
</tr>
<tr>
<td>Introduce zero application latency through web scale collection.</td>
<td>• Ensure the business units are experiencing optimal performance while minimizing wasted compute cycles, ultimately reducing hardware and energy costs.</td>
<td>Collectors introduce zero latency into the execution of any business application: either custom or packaged.</td>
</tr>
</tbody>
</table>
About AppFirst

AppFirst is the world’s only web-scale platform for comprehensive IT data. Through patented technology, AppFirst provides enterprises with complete visibility into all the applications and supporting resources in their IT ecosystem, regardless of the infrastructure. Fortune 500 organizations are leveraging this foundational insight to enable a new IT operating model to significantly drive down costs while ensuring critical service levels.

Learn how this can help your organization at AppFirst.com.