Data Center Monitoring Guide

Guide to Selecting a Data Center Monitoring System

While the process of selection of a monitoring system is necessarily unique to every enterprise, this document provides some guidance as to issues to consider when making that decision. Selecting the best monitoring system for your enterprise boils down to a single selection criteria: Pick the monitoring system that adds the most value to your business.

A monitoring system adds value if the benefits of the system are greater than the acquisition, implementation and operational costs.

Generally, the benefits an enterprise will obtain from a monitoring system fall into the following categories:

- reducing the cost of outages and service degrading events
- reducing staff cost (time) of investigations into performance and availability issues
- improved information efficiency

Note that the focus of assessing a monitoring system’s positives should always be on the business benefits, not the features.

Balanced against these benefits will be the costs of the monitoring system:

- acquisition cost
- implementation costs
- operational costs

Assessing the Benefits of Monitoring

A monitoring system is an efficiency tool – it allows enterprises to avoid and minimize expenses and revenue loss, rather than contributing directly to increased revenue. (Managed Service Providers that sell monitoring and value-added response services are an obvious exception.) Thus in order to assess the business value of a monitoring system, and to compare possible systems, one must have an idea of the possible expenses the tools will mitigate.

Minimizing the Cost of Outages and Service degrading events

Quantifying Outage Costs

Avoiding outage costs is a common justification of monitoring, but is often hard to quantify, and is different for every enterprise. For some enterprises (although increasingly few), downtime may matter very little, and only the simplest of monitoring is justified.

Each enterprise should consider both the immediate impacts of outages and the brand impacts, but both cases will require thought and discussion specific to the enterprise.

Consider the case of online retailers with directly measurable dollar/minute metrics attributable to web site sales. Does an outage mean that revenue for the duration of the outage is lost? Perhaps customers will simply purchase later, when the site is online. Perhaps the outage means customers lose trust in the brand, and not only make their immediate purchases at a competitor, but also make all future purchases at the competitor. In this case, the outage cost for a small but growing site could be much greater than at an established brand, despite a much lower sales volume. The established brand may impact $1million in sales during an hour long outage – but those sales will likely be made up later. A similar outage on a smaller, growing site may only directly impact $2,000 in sales – but the sales are likely to be permanently lost, and worse, the loss of
goodwill by early evangelists of the site can significantly affect growth.

An outage on a site that provides a subscription service may have less impact on longer term customers, but customers are more likely to churn if they experience an outage before they have internalized the value of the service – new customers, or those in trial. In this case, the outage costs not the customers subscription fees for a month, but the lifetime customer value of those that did not convert.

An outage of an internal IT virtualization infrastructure that idles the workstations of 150 engineers (at $150 an hour fully loaded salary) is superficially an obvious direct cost – but as exempt employees, the engineers may complete their work anyway, perhaps by staying late. Then the cost becomes one of employee satisfaction – and if it results in employee turnover, the cost becomes much higher. If an outage of IT systems affect sales people at the end of the quarter, preventing them from accessing their CRM, or perhaps their phone systems, there can be a very large cost – in sales staff dissatisfaction, revenue for the quarter, and even corporate stock price.

There are non-market driven costs too – downtime in a business unit may be valued disproportionately to its revenue contribution due to political clout of its executives. Thus determining the cost of an outage is not a simple matter of entering data into a formula, but requires knowledge of the revenue models of the enterprise.

**Quantifying Service Degradation Costs**

Service degradation issues can often cost more than outages. With an outage, there is a clear, identifiable situation – a service is down. With a degradation, there is often a lag before the issue is reported, another before it is acknowledged, and further complications with identifying the systems and personnel responsible (networking staff, server staff, and storage staff may each insist their respective systems are working correctly). This longer duration of the issue (compared to an outage) can result in larger costs. The costs may be lower sales revenue on an ecommerce site (slower site performance directly correlates with less conversions.1) For internal systems, costs may be inefficient use of engineers time as they wait for compilations or other resources; or less effective sales staff if their CRM system is slow. Given the high fully loaded cost of personnel, any system impact that detracts from productivity can quickly become a large drain.


**Analysis of past Outages**

Each organization will have to rely on its own experience to assess the historical frequency of outages, whether the outage would have been averted given ideal monitoring, the direct costs of the outage and the indirect, brand costs of the outage.

Some questions to discuss that can help guide this assessment:

- Why do you want a monitoring system?
- What do you want the monitoring system to do? What benefits do you anticipate getting from it?
- How many outages or adverse performance events occurred over the last month? 6 months?

For each historical incident, as best can be determined:

- What were the direct costs of this outage or performance issue?
- What were the 'brand' costs of this event?
- How many hours of staff time were involved in determining the cause of the outage?
- What is the fully loaded cost of staff time for the staff involved?
- What capabilities would a monitoring system have required in order to alert on the issue and identify the cause *during* the event?
- What capabilities would a monitoring system have required in order to alert on the impending issue *before* the event?

A question that is always useful to ask is “So what?” If some devices went down, and there was no monitoring – so what? Why does it matter? This is a good way to flush out who cares about the issue.
Reduction of staff cost for investigations into performance and availability issues

With increased complexity of applications and infrastructure, the time spent to determine the root cause of performance or availability issues can be a substantial expense that good monitoring can significantly reduce.

Consider the example of a performance issue on an e-commerce web site. Troubleshooting the issue could involve bringing in staff resources to look at the network, the web server operating systems, the front end application, the load balancers, the back end database, the virtualization platform that runs the database virtual machine, fiber channel systems that connect the virtualization platform to the storage, and the storage system. Any one of these areas could reasonably be the cause of the issue. Further, silos of information can exacerbate the time required to determine a system is not contributing to the poor performance. For example, the database server operating system may be observed to be running slowly, leading to troubleshooting efforts to focus on OS level tuning and issues – but the issue may be the underlying virtualization platform being memory starved, and transparently swapping out memory from the virtualized OS. In such a case, if the monitoring system alerted that the virtualization layer was low on memory and that swapping of virtual machines was occurring, and this information was available to all team members, troubleshooting would be much quicker, involve fewer resources, and the issue would be resolved sooner.

Of course, not every situation is going to be alerted on by monitoring, but even in such cases monitoring can still greatly reduce the time to resolution of the issue. This will only be true if the monitoring is collecting a wide variety of information, from a wide variety of systems, and making this information visible in chart form, so that trends and changes can be spotted by human intelligence, and the issue correlated with these changes. A simple example: after a software release, the performance of an application is worse. A quick examination of charts can show if there are differences in request load. If this is the same as recent historical levels, the monitoring can show if the database is performing significantly more table scans after the release, perhaps because a needed index was not created. Charts will also show that the increase in sequential scans was attributable to the release, and not a gradual increase over time with load; and also show how much extra Disk IO is being put on the storage system as a result, and how this is affecting request latency. Without historical charts, resolution of such an issue would take much longer – translating to a significant expense.

Improved information efficiency

By providing accurate data as to where resource bottlenecks are, and by aggregating data from multiple systems, monitoring systems can provide actionable data about costs and performance that improve enterprise efficiency. A simple example is that in the fact of performance issues and inadequate monitoring and analysis, it is not uncommon for organizations to purchase new capital infrastructure that does not address the root issue. (For example, upgrading front end CPU capacity when the issue is the storage system IO operations per second capacity.)

Another example where monitoring can optimize capital expenditures is to ensure equipment purchases meet current and future needs, but avoid overspending on overcapacity. (“Buying out of fear”, as one customer calls it – spending $80,000 on storage, in case the $50,000 storage is not performant – without knowing exactly what the requirements are.) It also allows purchases to be planned – trends can clearly show when circuit or equipment upgrades will be required, giving months of warning with commensurate negotiation power, rather than requiring immediate outlays to maintain service levels.

Monitoring systems collect a lot of information about a lot of systems, and this data can, if presented efficiently, allow new insights into the enterprise’s operations, that can realize better planning and expense control. Aggregating all the ISP bandwidth used per ISP, or per datacenter, can reveal opportunities for contract negotiation savings. Being able to track storage usage by business unit across all storage assets in an enterprise may not fall under the traditional rubric of monitoring, but given that monitoring systems collect the data underlying this information (storage capacity of every volume on every storage system), it is a reasonable item to extract from them. Being able to track real time and historical trends of a variety of performance and utilization metrics can provide unanticipated benefits to enterprises.

Costs

Acquisition cost

A typical period to assess the cost of a system is three years. Thus the acquisition cost should include initial purchase cost, plus 2 years maintenance, for a premise based system. A hosted system’s cost should reflect the cost over the three years (which is typically based on some usage metric – number of monitored systems, or datapoints, or end users.)
Cost to Implement

There are several components to this cost:

- **hardware.** Some monitoring systems require expensive hardware (particularly with regard to disk subsystem requirements) to scale to support a high monitoring load. Others can run on a low resource virtual machine, but typically trade off the amount of metrics tracked. SaaS based systems often have low resource requirements without the trade-off (as the demanding storage/processing is done on the provider’s systems.)
- **costs to meet availability requirements.** At a minimum, the monitoring system will require backups (tape costs, backup agent installation, load on tape drives, etc). There may also be a requirement for high availability – duplicate hardware, clustering, monitoring of the monitoring system, etc.
- **time to install the system to be ready for use.** Will the installation of the monitoring system software be done in an hour? Three weeks? By a professional services team?
- **training costs,** covering not just any cost of training programs, but the staff time to attend training, or to self-learn the system.
- **cost of staff time to implement initial configuration.** How long does it take to define what to monitor? To enter all the systems and their attributes into the monitoring system? To define escalation chains, or tune alert thresholds?
- **what is the cost to realise improved information efficiency?** Is it even possible? e.g. if the monitoring system is monitoring disk usage of all storage arrays, can that information be delivered in a way that represents the usage of storage across the enterprise by business unit? Does it require an external reporting package? Programming involving the monitoring system’s API? Or is such capability built in? What value does such a use provide to the enterprise?

Ongoing operation costs

Despite many enterprise’s concern with initial cost, ongoing operational costs tends to be the largest cost component of monitoring systems. The staff time required to reflect datacenter changes in the monitoring system can easily consume a full time employee. It’s a rare enterprise where the data center systems are provisioned, deployed, then left unchanged. Each enterprise should consider the associated costs (in staff time) and their historical and expected rate of change of the following classes of events:

- **adding another device of an existing type you’re already monitoring** (e.g. deploying another windows server – with increased adoption of virtualization, such deployments tend to accelerate.)
- **changing the configuration of a device being monitored** (e.g.changing a Mysql database to a slave, or adding another volume on a NetApp, or defining a new IIS web site instance on a windows server.)
- **start monitoring a completely new application** (e.g. deploying memcached)
- **changes to information behind the custom presentation of business data.** e.g. if there is a dashboard graph showing the total of production apache requests served per datacenter, what work is required when a new apache server is deployed? Does code need rewriting? Or does the monitoring automatically construct the appropriate graph?

Translating business requirements to features

Features required for Proactive Warning of Outages

Certainly one of the business goals is to proactively warn about, and hopefully prevent, impending outages. This is one of the easier business drivers to convert to a feature list, as it is driven largely by technical requirements. While any monitoring system should be able to alert of an outage on a system, and thus speed time to resolution, being able to proactively provide warnings of impending failures and performance issue requires different capabilities. It may require a monitoring system that can alert when a load balancer detects that a Virtual IP has less than the desired level of server redundancy; or when request latency is increasing on a storage array, or when database replication is lagging more than the desired time offset, or when the number of server threads on a Java application is approaching a limit. Being able to prevent outages requires a much more capable monitoring system – but the capabilities must match the infrastructure deployed.

Converting other business requirements to features

As noted above, the process for selecting a monitoring system should care less about features and more about evaluating how the system will impact business, positively or negatively. To align features with business value, an enterprise should detail the way their organization works (or how they want it to work), and translate that into capabilities that help meet their business goals. The important issue to remember is that except for specific technical goals as mentioned in the above section, the feature list should detail business goals and capabilities, not specific ways of achieving the goals.

For example, an organization may operate with the following operational constraints: they run east and west coast datacenters,
with staff at both locations, and applications run at both. They have infrastructure from 3 business units at each location, and some infrastructure is shared. They employ virtualization technology, and have little staff time to devote to their monitoring. Their custom applications are a mix of java and windows .NET, and they also use Tomcat, IIS-, MySQL and SQL Server. They want alerts to be routed to the appropriate teams, differentiating between roles even within the same host (e.g. Storage and DB groups may both be paged for different reasons for the same host), and escalated to people to ensure coverage. They want morning alerts handled by their east coast staff, and later switch to the west coast staff. There is frequent change in their datacenter in terms of reconfiguring or adding devices or applications, but not all the devices are production, warranting production alerting. They plan to grow some infrastructure into Amazon's EC2 cloud in the future.

Their business goals are to allow the growth of service revenue, which will require additional infrastructure to handle the load. They wish to target their capital expenditures for this growth correctly; avoid headcount growth; minimize downtime and its impact on revenue and get better information for cost allocation among business units.

Translating these needs to features with their associated business drivers, they can best meet their business goals by finding a system with the following features:

- they need to monitor using APIs specific to their virtualization platform, and also monitoring for JMX, WMI, MySQL, SQL server, and snmp devices, in order to provide proactive monitoring for their infrastructure and minimize downtime.
- knowledge within the monitoring system of what to monitor and chart, and when to trigger alerts, for all their devices and platforms. They have estimated that it would take 200 staff hours to define the initial monitoring profile of their applications and systems, with further costs for each software upgrade or firmware update.
- the monitoring should automatically track changes in each device that may require changes in monitoring. The absence of this will cost them 12 staff hours a week to keep up with changes in devices and applications.
- they need the ability to monitor within EC2, and track the changes in machine instances in EC2 as machines are added/removed. The absence of this feature will preclude the use of EC2 infrastructure, necessitating $200,000 in extra colocation costs for further cages and infrastructure.
- the ability to manage multiple locations from a single console. This will minimize monitoring system deployment costs and ongoing operational costs by allowing cross site issues to be managed in a unified manner.
- they need alert routing and escalations that can be managed by device group, type of alert, and time of alert. Due to the number of systems, it is not feasible to have all possible alert recipients receive all alerts (and retain employees), so the absence of this feature would necessitate creating a first level NOC system purely to route alerts manually, at substantial cost.
- multiple business units imply there is likely a need for role based access control – but whether this feature adds any business value depends on the degree of openness and interaction between business units.

Each feature should be prioritized in terms of how much value each feature brings to the enterprise. This value will vary by enterprise – an organization with a fairly static infrastructure may decide that relying on manual workflow is sufficient for ensuring changes to infrastructure are reflected in monitoring (although I would suggest that processes done rarely are also rarely done when needed!) One enterprise may initially desire role based access control, but on reflection find that it adds no business value. Another may determine it is essential, as it allows them to unify monitoring while meeting contractual requirements of confidentiality for their customers.

Having determined the list of features and their relative value to enterprise, an organization can then narrow down a list of proposed solutions that meets the most important of these features, in order to accurately assess the value to the enterprise.

Evaluating Candidate Software

Each candidate solution should be evaluated for the prioritized list of features – as they relate to business value – weighted as appropriate for the typical actions of the enterprise.

Typical areas to evaluate solutions against will be:

- The amount of each of the cost's identified above under Implementation and Operational costs. Operational costs are likely to be the larger over the life of the system.

- Will the system cover all devices/applications, or will point solutions still be required for some areas? What is the business cost if multiple
monitoring systems are employed? (Typically duplicate alerts, difficulty in scheduling planned downtime for systems; in setting alert
escalations; in correlating performance issues across devices)

- Does the system provide monitoring sufficiently comprehensive that it will alert proactively, even for issues staff didn’t know they should
  be monitoring, that will reduce the likelihood of an outage? (e.g. is it monitoring for failures in a redundant supervisor module? Failed power
  supply? Lack of spare disks? Queuing in a load balancer?) The value of this is directly related to the costs of downtime that can be
  eliminated.
- Can the system monitor and trend the metrics that matter? (e.g. For a NAS or SAN storage array, the performance directly impacts all the
  applications and systems that use it – so the read/write request latency should be a baseline metric. Yet many systems cannot collect this.)
- How capable is the system of being extended? Is there an API available that allows integration into provisioning systems? Does that matter
to the enterprise?
- Does the system allow my staff to manage more systems? Or will the time to manage the monitoring eat into their time that could be
  spent creating more strategic value for the business?

With a trial deployment, the realistic costs and benefits of a system can be assessed, always keeping a focus on business value
comparison, not feature comparison. There will likely be multiple ways to deliver the same business value, that may not fall
into the same “feature” check box.

A simple example is system security. The business goal is to prevent the disclosure of information that may be embarrassing to
the enterprise or provide intelligence to competitors or vendors. Yet this goal may be translated to a feature checklist as “all
data stored locally in corporate datacenter.” This is one way of achieving the goal (although it makes many assumptions about
the deployment.) But the goal may be better achieved through a SaaS model, even though it would not meet the checklist
requirement. A SaaS system is likely to be delivered from audited, tested datacenters with 24 hour manned guards, biometrics,
cameras, external penetration tests, and from a system designed explicitly with security in mind and encryption used at many
levels (transmission and storage of data, etc). A premise based system, even if operated behind the corporate firewall, is likely
to be deficient in many of these areas – so while it would meet the checkbox, it would not deliver the business value as
efficiently. This illustrates why it is important to detail the business drivers for each feature (“maintain security of data”) rather
than just the feature as the end users expect it to be delivered (“all data stored locally in corporate datacenter”) – no one will be
able to predict the ways in which all the business drivers can be delivered, so listing the driver makes the assessment far more
likely to based on the business driver, rather than the anticipated way of delivery.

Conclusion

We hope this whitepaper illustrates some of issues involved in selecting a data center monitoring system. Selection of such a
system will always require a good knowledge of the enterprise to be monitored, so that business value can be accurately
aligned with the benefits of the systems. Selection lists should be driven by business values, except for specific technical
requirements such as the ability to monitor a specific protocol. Some of the questions above should help bring out the
expected benefits and costs of a monitoring system. After all the discussions and dialog has occurred, the selection of a
monitoring system comes down to the simple statement made at the beginning of this paper:

Forget about features. Pick the monitoring system that adds the most value to your business.

© 2011, LogicMonitor.com

LogicMonitor provides SaaS based datacenter monitoring that delivers automation,

improves uptime and frees staff resources. (But what matters is if we add value to your
business. Find out with a free trial deployment, at www.LogicMonitor.com.)