VIRTUALIZATION & Cloud Review

EBOOK

IT MONITORING IN A HYBRID CLOUD WORLD

The changes faced by systems administrators today aren't simply about the technologies we implement, but about the scale at which they're implemented. How we manage, monitor, and maintain the technology in our care needs to change as well. **by Trevor Pott**

Forward

Like most people who will read this, I have spent the majority of my life working with computers. By this, I don't mean simply answering emails or writing documents, but building, fixing, administering, testing, or architecting systems, services, and entire data centers. In the decades that I have tended to my electronic herd, IT has changed—is changing. Will continue to change in the future. Staying on top of that change is a constant challenge.

Today, I make my money as a writer. Some call me a journalist, some call me an analyst, and my current publisher calls me a contributing editor. The title is irrelevant. The writing is just another tool, and none of it changes what I truly am.

I am a systems administrator. No matter what I do, no matter what others call me, a lifetime at the coalface of IT has ingrained in me certain things that cannot be easily expunged from my personality.

I am risk averse, especially when talking about something that will be end-user facing. I like der blinkenlights, but new and shiny goes through the testlab first. The tics and mannerisms of a sysadmin constantly being berated by layers 8-10 of the OSI model are simply part of who I am.

Keep your head down. Don't remind them that you exist. Printers were sent from hell to make us miserable. Uptime is everything.

The pace of change in the IT industry is accelerating. Changes brought about by cloud computing, composable workloads, hybrid IT, the internet of things, and the evolution of security from something comprehensible into something only an appropriately complex algorithm can parse.

Today's changes are different than the simple evolution of technologies we've experienced in the past. Today's changes are more fundamental than the switch from mainframes to PCs or from RAID cards to external storage.

The changes faced by systems administrators today aren't simply about the technologies we implement, but about the scale at which they're implemented. How we manage, monitor, and maintain the technology in our care needs to change as well.

Cover: VectorFusionArt/Shutterstock.com
iDEAR Replay/Shutterstock.com

IT MONITORING IN A HYBRID CLOUD WORLD

One of the websites I am responsible for is down. Determining why it's down, however, is a bit of a journey. Ten years ago, figuring out what went wrong, fixing it, and altering procedures to prevent a recurrence would have been relatively easy. But today, hybrid IT is the new normal, and solving these sorts of problems can be quite complex.

A decade ago, I had all my clients hosting their websites on their own servers. On behalf of my clients, we ran email servers, DNS servers, caching, load balancing, intrusion detection, front-end, database, a box full of crazed squirrels, you name it. None of the data centers I've overseen are large, but at their peak, several of them ran a few thousand workloads each.

This was in the days before desired state configuration and the "pets vs. cattle" debate. There were a lot of pets in these data centers.

Pets and Cattle

A typical data center for me had some fiber, a backup VDSL connection, and only a handful of publicly facing workloads. Not a lot of upstream, a whole lot of downstream. We'd have redundant air conditioning with outside air, UPSes, compute, storage, and networking, and the really rich folks would have a generator. Five or six of these would end up being a lot of work for two sysadmins.

As you can imagine, workloads were sent into the public cloud. Web-facing stuff first, because it had a lot of infrastructure "baggage." Ever more mission-critical workloads moved until—seemingly without anyone noticing—the on-premises data center, the hosted solutions at our local service provider, and the public cloud workloads were scattered about the continent.

Despite the geographic dispersal of workloads amongst various providers, however, any given client's workloads remained critically conjoined. What was out in the public cloud fed into the on-premises systems, and everything had to be synchronized to the hosted systems for backups. If the wrong bit fell over, everything could go sideways.

Joe Techapanupreeda/Shutterstock.com

Tracing a Website Outage

The first problem with my website outage was that I didn't notice the outage, a customer did. Which is embarrassing, but also an important bit of information. Either my monitoring software couldn't detect the problem, couldn't alert me of the problem, or was also down.

This could be useful additional diagnostic information for me, or a separate fire to put out. I won't know until I'm a little further down the rabbit hole, but it is troubling.

Having spent years with pre-virtualized, one-application-per-metal-box workloads, whenever something stops working, my first instinct is to look for hardware failure. Today, that would mean seeing if the virtual servers, hosting provider, or public cloud had fallen over.

A quick look shows that I can connect to all the relevant management portals and the various management portals claim all the workloads are up and running. Unfortunately, I can't seem to log in to any of these workloads using SSH®. This is alarming.

The hosting provider gives me console access to workloads—

something that, sadly, my public cloud provider does not—and I am able to quickly assess that the various website related workloads are up, running, have internet access, and otherwise seem healthy, happy, and are enjoying life. They're not currently handing customers, which means that the switchover mechanism believes the primary workloads are still active.

I get an email on my phone, so something has to be working with the public cloud hosted workloads; part of the mobile email service chain lives there. I hop on Slack® and ask a few of my sysadmin buddies to test my website. Some of them can get there, some of them can't.

While I pour coffee into my face and curse the very concept of six o'clock in the morning, a phone call comes in from a panicked sales manager: only orders from one specific website have shown up in the point of sales system overnight. Five other websites haven't logged a single order.

Rather than drag you through each troubleshooting stage, I'll jump right to the end: the answer was DNS. More specifically, the outsourced DNS provider had a really interesting oopsie where half of their resolvers wouldn't resolve half of our domain names and the other half worked perfectly. This broke nearly everything, and we weren't prepared for it.

Old Monitoring in a New World

For most of our customers, the monitoring suite lives with the hosted provider. There, it's just one more VM on a box full of various VMs. By living in what amounts to the backup location, it's also not part of either of the primary production infrastructures (the public cloud and the on-premises data center), so it seems like a reasonable place to put a widget that would check to make sure both of those sets of workloads were up and public-facing.

In the case of my early-morning outage, because there was not actually anything wrong with the website, and the hosting provider provides a caching DNS server, the monitoring solution didn't see anything wrong. It could resolve domain names, get to the relevant websites, see email passing, and so forth.

Back in the day when everything ran from a single site, this was fine. Either things worked, or they didn't. If they didn't work, wait a given number of minutes, then flip over to the disaster recovery site. Life was simple.

Today, however, there are so many links in the chain that we have to change how we monitor them. DNS, for example, clearly needs to be monitored from multiple points around the world so that we can ensure that resolution doesn't become split-brained. Currently, none of our customers use geo-DNS-based content delivery network-based regional website delivery, but it's been discussed. That would add yet another layer of monitoring complexity, but this sort of design work can't be ignored.

Hybrid IT Is the New Normal

None of my clients are doing anything that, by today's standards, is particularly novel or difficult. Websites hosted in the cloud update the point of sales software on-premises. They receive updates from that same software. Orders for some customers arrive via FTP or SSH and are funneled to the on-premises servers for processing.

> The first problem with my website outage was that I didn't notice the outage, a customer did. Which is embarrassing, but also an important bit of information.

There is middleware that collects order tracking information from manufacturing, invoicing from point of sales, information from the e-stores, and logistics information from the couriers. All of this is wrapped up and sent to customers in various forms; there are emails, desktop and mobile websites, SMS pushes, and I think one client even has a mobile app. The middleware also tracks some advertising data from ad networks and generates reports.

Somewhere in there is email. Inbound email goes through some hosted anti-spam and security solutions. Outbound email comes from dozens of different pieces of software that will forward through smart hosts at various points until they are funneled through the main server located in the cloud. Email can originate from end-users or from office printers, manufacturing equipment, the SIP phone system, or any of dozens of other bits of machinery.

None of the clients for whom I act as sysadmin currently have more than 200 users. Most are in the 50-user range. None of the technology they have deployed is even as complicated as a hybrid Exchange™ setup or hybrid Active Directory®.

Despite this, these small businesses are thoroughly enmeshed in hybrid IT. This multi-site, multi-provider technological interconnectivity means changing how we think about monitoring.

Hybrid IT is not a novelty. It's not tomorrow's technology. It's the everyday business of everyday companies, right now, today. Are you ready?

Aside: Infrequent Administration

The smallest of organizations can be found using some combination of organization-managed and as-a-service IT infrastructure. We may already be at the point where large organizations that don't utilize both no longer exist. What's also becoming clear is that the days of IT vendor monoculture are coming to a rapid end.

Once, it was common to have all of one's storage be EMC®, and all of one's switching be Cisco®. All desktops were Microsoft®. All mobiles were Blackberry®. These things were a given, especially in enterprise IT. For the most part, this has already changed.

Heterogeneity in everything from operating systems to switch vendors is now the new normal. Where we once had one vendor's worth of management interfaces and CLIs to master, now we have many. Complicating matters, the number of domain specialties to be managed under the umbrella of IT has ballooned. More IT niches total, and more vendors per niche.

All of this is a drawn-out way of saying that our data centers have become far more complex. For me, managing that complexity has become increasingly difficult. As technologies become more mature, dynamic, and automated, I need to interface with them less and less frequently, and this is a problem in and of itself.

Months can go by between my interactions with a particular vendor's products. Keeping the quirks of one particular command line in mind when you only need to use it every six months isn't exactly easy. As such, I find I need to rely on an increasing amount of software and services to keep it all straight. What I have, what it's doing, and how it is currently configured.

TROUBLESHOOTING IN COMPLEX IT ENVIRONMENTS

In the data center, everything is connected. With hybrid IT being the new normal, the number of technologies underlying a seemingly simple support ticket can be staging. Solving problems and closing out the day's tickets can involve troubleshooting both on-premises and off.

It's not enough to talk about tech problems in the abstract. Real-world examples make problems more concrete and easier to understand. Being a systems administrator has furnished me with enough examples of doing it wrong to last a lifetime, but the most important lesson I've learned about troubleshooting is that nothing beats understanding how all the moving parts interact.

Business Processes, File Shares, and You

Let's consider a fairly normal web serving setup that caused a fun problem. The website in question serves up some custom middleware that allows both staff and customers to track orders through a manufacturing process, and one day it developed a problem serving up some images.

VectorFusionArt/Shutterstock.com

The website is a combination of PHP scripts, generated HTML, a couple of databases, and an astonishing number of images. For the website to "work"—that is, for it to deliver content to users in a timely manner—a number of different systems across the organization need to function.

The website isn't static; at any given point in time, a dozen different systems and at least as many staff could be generating files or making changes. There are thousands of generated images, static pages, and database entries on a slow day.

One day, a ticket showed up that said the website was having trouble serving images. Help desk staff went to the site; it seemed to be serving images just fine. The ticket was bounced back to the user, requesting examples of images that wouldn't load. The user provided them, and the images would load for some of the help desk staff but not others. Being an intermittent problem, it was kicked upstairs to the systems administrators.

Intermittent problems are the most frustrating problems to diagnose, and this particular issue proved no exception. Despite there being umpteen log files on the web server, none of them actually logged a relevant issue. The file server that served up the files to the web server was similarly unhelpful.

Back To Basics

In order to solve the problem, we had to go back to basics and tease apart each element of the stack. The images are hosted on a Windows® file server. This is due to both the Windows applications that generate the images needing to store images on an SMB share and business process needs that require that these images be easily available and editable by staff members. Con-

necting to an FTP to pull down images to edit, then pushing them back up to a web server just wasn't going to fly.

Under basic testing, each piece of the puzzle was working. The Linux® server mounted the SMB share just fine. The Apache® web server could see the relevant directory and serve images from it. It turns out, however, that a specific bug in Apache means that when serving files from mounted file-based network storage, one needs to make sure that EnabledSendfile and EnableMMAP are set to off in the httpd.conf file.

This bug only shows up under specific circumstances and is hard to catch if you can only test components one at a time. The ability to test the whole stack at once, gather real-time information on performance and loading, and then compare this with logs and system events would have turned two days of troubleshooting into less than an hour's worth of effort.

Today, this same website solution now has tentacles in Amazon® public cloud and in four separate service provider hosted setups. It not only ties together information from on-premises manufacturing processes, but several outsourced manufacturing chains, as well as numerous logistics companies. This makes the ability to see the whole stack of interconnecting elements all the more important.

Another example of things gone wrong concerns an increasingly important element in a modern data center's infrastructure: the reverse proxy.

Limits That Change

Another example of things gone wrong concerns an increasingly important element in a modern data center's infrastructure: the reverse proxy. As part of ongoing modernization and security efforts, network isolation had been taking place for several years. Anything that posted a management website, for example, was placed behind a reverse proxy that contained various intrusion detection features.

For the most part, this worked remarkably well. Everything from VMware® host websites to Dell® iDRAC or Supermicro® IPMI websites worked from behind the proxy. All management website access worked just fine for over a year before we ran into a problem.

By the time we did finally run into a problem, almost everyone had forgotten the reverse proxy was even there. We had gotten used to using domain names to access management websites instead of IP addresses, and the reverse proxy just kept on doing its job, invisibly.

Eventually the day came when some virtual machine migrations were needed. At some point during the migration process, we needed to download virtual machines using VMware Web-Based Datastore Browser. Downloads would start, everything would go well, and then several minutes in, downloads would fail.

We crawled all over the error logs on the VMware hosts, the storage servers, and the vSphere® server to no avail. There were no error logs, nor any reason that these downloads should fail. And then we remembered the reverse proxy.

It turns out that the reverse nginx setup on the reverse proxy had a 2GB file limit. This was a simple issue that cost hours, in large part because we didn't have the ability to even pin down which system was causing the problem.

Today, workload management consists of wrangling virtual machines on VMware and Scale Computing infrastructures, both on-premises and on hosted service providers. Very soon, it will also consist of handling VMware on Amazon public cloud (via VMware on AWS®). Each location has its own security defences, reverse proxies, and so forth. Each location is another collection of moving parts that we as systems administrators need to keep in mind, not in order to serve workloads to our customers—both internal and external—but simply to access the management tools that let us diagnose the actual customer-facing workloads.

Having the Right Tools is Important

At some point, it all gets too big. If my job were to coddle one application stack, I could easily remember all the layers in the stack.

Apache is a docker container that sits on top of CentOS® Linux. That Linux instance is presenting files to its containers, which are mounted using SMB and NFS shares for X and Y files, and via iSCSI for Z files. The Linux instance is inside a VM running on a Scale cluster, and the reverse proxy is nginx on CentOS on a VMware cluster. Internet is delivered via VLAN 1000 on the second NIC. SAN and NAS

are VLAN 2000 on the third NIC. Customer access is VLAN on the first NIC.

On and on and on; configuration details both minute and gross. Easy if your area of responsibility is one workload. Crushing if it's thousands.

Somewhere along the way, we need a way to document all of this information. What's connected to what, and how. We need more than Visio® diagrams, spreadsheets, and notes in a text file. We need to be able to enter this information into a system that will monitor each connection and offer up details of how the interconnected stack of technologies works when something goes sideways.

In essence, modern systems administration requires a means to augment the memory of systems administrators. A Google® for data center infrastructure, both virtual and physical, on-premises and off. We need a means to visualize these issues, and to test each step in the chain, both individually and together, because the whole chain may behave differently than when testing individual links.

Data centers aren't going to get less complex, and hybrid IT isn't going away. The question now is which vendors will deliver a solution to this increasingly important systems administration problem.

Aside: Learning to Social is Probably Important

We all have character flaws. Myself probably more than most. Notable amongst my flaws is that I only really get along well with people quite a bit like myself. Operations nerds. People who wake up in the middle of the night in a cold sweat—not because of the dream of turning up to work naked, but because they dreamt of a triple disk failure on a RAID 6.

I don't really get along with developers. Developers write the applications that never quite work right and I have to band-aid in production. I certainly don't get along well with management. When things run smoothly, they ask why they pay me and when things cease functioning, they ask the same thing.

My drinking buddies down at the pub don't come from the world of unlimited budgets, but from places where each penny is counted, every upgrade begrudged. When the finger of blame and the hairy eyeball of shame are your constant companions, it is easy to feel constantly under siege. The temptation to respond to the increasing pressures of professional complexity with a closed door and a closed mind can, at times, be overwhelming.

Unfortunately, this alluring path of isolationism is toxic, both personally and professionally. We are social animals, and we need the expertise, insight, and emotional support of others to solve the challenges with which we are faced. No man is an island, and as our areas of responsibility grow ever more intertwined with those of others, finding ways to bridge the gaps between us becomes a survival mechanism.

THERE IS NO "SILO" IN TEAM

A management approach that treats all IT infrastructure, software, and services as a single entity—holistic IT—is the goal of many organizations. The diametric opposite of traditional IT, holistic IT is often associated with modern buzzwords like DevOps. Buried underneath layers of marketing and hype are some logical and rational reasons to embrace holistic IT that are grounded in solving real world problems faced by everyday practitioners.

Traditional IT is siloed. Each group of warm bodies is their own world. Developers, designers, project managers, QA techs, and systems administrators all have roles to play in IT. In a traditional IT shop, however, each group lives on the other side of the wall from one another.

Even within these departments, there are silos. Consider how systems administrators have become specialized. In a single organization, you may find an administrator or multiple administrators focusing on individual applications, operating systems, websites, or specific aspects of physical infrastructure.

The communications barriers, rivalries, and mutual distrust fostered by this approach are so universal they have become a permanent part of our popular culture. There are internet memes and entire vocabularies of thinly coded speech dedicated to pointing the finger of blame at the next room over. That's where the black magic happens, and when problems arise, throwing a rival group under the bus is an easy out.

It's not efficient, but it is very human. Unfortunately, our competition these days is from shell scripts and robots, so us puny fleshbags need to up our game.

Thinking Holistically

The first steps along the path towards holistic IT don't involve drum circles, talking sticks, or even magic crystals. There's no sharing of feelings, or even a need to play nice. Starting down the long road to joined-up thinking requires a little more than buying the right software and ensuring everyone uses it.

Of course, finding the right software for your particular organization's circumstances has a difficulty level of "reality bending," but let's put a pin in that for the moment. The software is important because it not only forms the foundation for cross-team interactions, but it gives everyone something to hate that isn't one another.

Shifting the focus of emotion is important. Any attempt at holistic IT will repeatedly touch the raw nerve of job security for all involved. When you are a specialist working in a silo, it's easy to make yourself indispensable. When you work in a cross-discipline fashion with a large number of other intelligent people, it's easy to feel like you're under constant scrutiny and/or being turned into a disposable cog in a vast machine.

It's also important to recognize that the push towards holistic IT is happening alongside some other important transitions within our industry. Hybrid IT is the new normal and it makes troubleshooting all the more difficult.

Cloud-in-a-can is a thing that we can now buy, which will ultimately end up transforming every aspect of how organizations deliver IT. All of this cloud talk is changing how we think about everything in IT, including economics and supply chains, and it has brought both systems integrators and trust back into every serious discussion about IT's future.

All of this is happening against a backdrop that includes a history of organizations treating IT staff terribly and genera-

TW Stock/Shutterstock.com

tions of conflict between end-users/customers and IT practitioners. Holistic IT is about people, and that is always a delicate balancing act.

Software Feeds, Speeds, and Needs

Helping different silos work together requires—at a minimum—infrastructure discovery, e-discovery, secure archiving, reporting, auditing, monitoring, ticketing, testing, orchestration, backup, and recovery software. All of this has to cross disciplines. At the core of each, rests concepts like profiles, recipes, Role-Based Access Control (RBAC), and a huge dollop of CYA.

Holistic IT is about people, and nobody wants to be target practice for the blame cannon. Taken together, infrastructure discovery, e-discovery, secure archiving, reporting, and auditing are tools focused on figuring out what state everything is in and then archiving it in a secure fashion so that one can prove what happened in the past.

Helping different silos work together requires—at a minimum—infrastructure discovery, e-discovery, secure archiving, reporting, auditing, monitoring, ticketing, testing, orchestration, backup, and recovery software.

These are tools management doesn't often see the value in, or may even consider to have negative value, as their primary purpose in the real world is for IT practitioners to say "yes, I did in fact warn you about this horrible thing, you did nothing, and guess what happened?" Bad for penny pinchers. Great for morale.

Monitoring and ticketing software are likely to be the most difficult software purchases any IT organization makes. Both need to work together to draw all stakeholders in a problem together without assigning blame or giving anyone any easy out to ignore the problem, but still requiring someone to take charge and shepherd the ticket to resolution.

Keep the Problem Inside the Walls

Problems in a modern data center can be complex. Issues experienced by users can involve dozens or even hundreds of different technologies, crossing potentially dozens of individual silos within an organization and possibly involving external suppliers or vendors.

Ticketing and monitoring software needs to be able to help practitioners narrow the scope of the problem to a few likely technologies and make it easy to engage the individuals responsible, all while providing a solid trail of indisputable evidence that doesn't allow anyone to "throw the problem over the wall" in order to get rid of it.

Each organization's internal business processes and IT mix will determine which solutions are required to pull this off. Lashing those solutions together into a workable whole is the hard part, and in many situations, will require custom code.

Testing and orchestration software comes in a few different flavours. It can be as simple as copy data management or private cloud software. Both should allow infrastructure administrators to define templates or recipes for workloads, often including backup regimens, storage profiles, and more. Developers and application administrators can use a self-service portal to generate new workloads, copy existing ones, move workloads, and so forth.

Ideally, all of this is backed by robust RBAC which controls who has access to what, and what they can do with the data. This, in turn, feeds into the CYA software stack, making audits—and ultimately compliance—easier on everyone.

Testing and orchestration software can also be much more robust. It can go down the route of desired state configs, with a much more programmatic approach. Automated regression testing, chaos monkeys, versioned configurations, and more are part of bleeding edge holistic IT.

Backup and disaster recovery software gives everyone the ability to sleep at night. Far from being simple holes into which all of the day's changes are poured, backups and disaster recovery have almost no value if nobody has ever bothered testing them to see if they can restore an organization to a working state.

From recovering a single file to an entire data center, there is often more red tape around backups and disaster recovery software than any other part of the organization. In part, this is simply because it isn't regularly used. Some aspects, such as SMB file recovery, have evolved end-user self-service portals as a basic expectation. Others—such as the restoration of active workloads—are still very much guarded by a priestly case of

paranoid and testy administrators wielding forms to be signed in triplicate, which exist primarily because the restore process lacks any semblance of formal testing.

Putting It Together

The end goal of holistic IT is to quantify not only what each workload consists of, but all of its interactions. What happens when a customer pushes "submit" on a website form? How many systems are involved in making that service available to the customer? Is the entire stack instrumented? Is each part of the form tested as part of ensuring service availability? Are those tests built into the monitoring and ticketing system, and will they survive a disaster recovery event?

The finger of blame makes a huge difference in how we perceive the world. If you can shrug off blame for something not working as someone else's problem, a different attitude is taken towards problem resolution than when it's your fingers in the vice.

If IT is a single entity—one giant interwoven machine—then all parts must act in harmony for it to function. Harmony is enabled by ease of use, and ease of use is very, very difficult. Documentation, adherence to standards and procedures, and nearly obsessive attention to automation are all required.

Holistic IT is not about job elimination. It isn't about turning us into faceless drones. It is instead about the simple admission that today's IT—even in the smallest shops—is composed of so many moving parts that it's simply too much for any one human mind to hold. We are, with very rare exceptions, simply not capable of knowing everything that relies on the piece of IT we are responsible for, nor our chunk of the data center relies on.

We need software to help bridge the gap. We need to augment our very human memory with documentation and discovery, history and testing. We need to be able to prove to ourselves where the problem lies so that we can have rational, professional conversations with our peers and solve problems for our customers.

Above all, those of us championing this future of IT need to be aware that these changes are big, and big change is scary. Holistic IT makes life easier, but it is ultimately all about people. Keep the feelings of your human compatriots in mind and make the transition slowly. One piece of software at a time.

Aside: No, I Don't Know Everything About Every Application

Communication difficulties with my colleagues are not my only character flaw. Like many systems administrators, I don't speak "user" particularly well. There's an irony to this: with the increase in Software as a Service (SaaS) adoption amongst businesses of all sizes, today even systems administrators are users to someone else.

The finger of blame makes a huge difference in how we perceive the world. If you can shrug off blame for something not working as someone else's problem, a different attitude is taken towards problem resolution than when it's your fingers in the vice.

Perhaps the most frustrating manifestation of this that I've encountered is the degree to which users will agree to help a systems administrator solve a problem. I've lost track of how many times a problem could have been solved in five minutes, but took hours because of a user's recalcitrance in providing even the most basic information.

Barring hardware failure, computers do as they're told. It's the people part of the equation that's hard. As the whole of our society moves at a breakneck pace towards automation, the burdens that rest on each of us, regardless of occupation, increase. As important as communication is, so too is removing the necessity for end-users to provide the mundane details that make troubleshooting easier.

We, as systems administrators, need tools that will enable us to solve problems with the minimum possible user interaction. Like us, they have better things to do with their time, and they want their computers to "just work" without having to put the effort into communicating with strange people, namely us.

LOOKING AT IT MONITORING FROM BOTH SIDES NOW

For the user, everything is about the app and their experience of using it. Sysadmins have to look at layers or dependencies when they are troubleshooting. And the problem that prompted the user to call support may not be occurring when a sysadmin goes looking for it, making time-stamped records are important.

This difference between how a user sees a problem and how a sysadmin sees a problem can sometimes be vast, leading to problems in translating between the two. Sysadmins the world over make jokes about tickets submitted by users. Conversely, users the world over think sysadmins are unhelpful, arrogant pains in the neck. Both views are born from a frustration in being unable to bridge the gap in individual experiences, and a lot of that boils down to the tools we use.

For a user, the physical device they use is just a tool. The network is something they may not even know exists. Anything on the other side of the network from their endpoint is a black box that they have no interest in understanding.

This is a perfectly rational approach to using a computer. I am not, for example, particularly interested the nitty gritty details of fuel distribution in the planes I fly in, nor do I care about the details of instrument landing system protocols. I get on the plane, it flies to my destination, I get off. It is a tool to me and nothing more.

I choose the plane instead of the ubiquitous car metaphor for a reason. For many of us, planes are a not-quite-necessary evil. There are almost always alternatives to taking a plane, but those are slower, less efficient and rarely an option for serious business use.

Perhaps more to the point, a lot of users feel about computer support the way most of us feel about traveling by air.

Bureaucracy

Often, sysadmins get a bad rap. They can come off as suspicious, overworked, underpaid, undertrained, and ultimately capricious. Computers are bizarre, poorly designed devices all on their own. Having to deal with sysadmins just makes everything about computing that much worse.

Thinking about people as data packets helps. Sysadmins exist to make the lives of individuals easier. They exist to protect the system as a whole. Individual complaints are diagnostic, but what matters is that the system as a whole keeps running.

iDEAR Replay/Shutterstock.com

If there's a problem, you isolate and remediate. You don't want to turn off the whole data center unless you absolutely have to. By the same token, it's better to down an individual workload than to start bogging down the entire system for everyone.

From the sysadmin's point of view, order is required. There is no room for exceptions. You force all workloads to behave according to pre-defined rules and everything works smoothly. Anything that breaks the rules is quarantined and forensically

examined. Systems administration is much easier when one can emit rigid fiats and ruthlessly enforce compliance.

Over the past 20 years, however, a few things have changed.

Resistance

There's cloud computing out there. Everyone has smartphones. You can connect laptops to the internet using the cellular network. In other words: there are options available to work around demanding systems administrators, and when pushed, that's exactly what people do.

While the user has options or a work around, it's not so simple for systems administrators. The sysadmin doesn't get to take that easy path of building a system where everything follows the rules and you just look for what's out of bounds if something stops working. In order to keep users using systems as intended, they have to provide systems that users actually want to use.

Reverse the Polarity

In the 90s, things involving computers occurred largely consecutively. I wrote a file in a word processor. I saved the file. I then transferred the file somewhere. I opened it on another computer. Backups were regularly done to an attached tape drive.

Unless there was a lot of scripted batch processing occurring in the background, each was a distinct action that I had to consciously perform. This has radically changed today.

When I open my word processor, it checks with my operating system to see what my identity is. This is then verified against a cloud server to make sure that I'm allowed to use that word processor, and may even involve authenticating against that cloud service. As I type, my document is spell and grammar-checked in real time, with missed words and word selections streamed back to the mothership's big data system. At some point, my word processor updates its checkers with the results of the big data crunching.

My document is also auto-saved regularly. This is kept not on my computer, but on the company server. A copy of each save is sent up to my personal cloud storage, which performs versioning. The server storing the master copy also regularly versions any manual changes to the master file made when I push the "save" button. The master file is backed up every night, and those backups are versioned, with copies of the backups also kept in the cloud.

All of this busywork happens behind the scenes, but for me as an end user, the experience hasn't changed much in decades. I still open a word processor, type my document, save it, and close the word processor. The bit that I don't have to worry about is moving it over to a floppy disk: today, my smartphone and my notebooks, all of which are attached to my cloud account, automatically download the latest master copy of the documents I am working on.

Increasing Complexity

The user experience may not have changed much, but the underlying technological components have. In the 90s, creating a document took only my PC. Today, the creation of that document also involves several servers and networks, many of which I don't control.

At a minimum, there are my files, directory, DNS, and backup servers, the cloud servers for authentication, application distribution and patching, big data analytics, results distribution, DNS, and backup for all of those. My network, the network of the cloud and DNS providers, internet service providers, and all the backbone providers in between are also involved.

Twenty years ago, all of this worked in computers; today it doesn't. Systems administrators can't even see all aspects of the system anymore, and it grows increasingly more dynamic and complex by the day. Continuous delivery, A/B testing, and other modern application development and service delivery models mean that it is entirely possible two users sitting side by side aren't being delivered the same version of an application.

The dynamicity of network routing and DNS, as well as the ubiquity of load balancers, reverse proxies, and so forth also mean that the electronic pathways taken by a user to access an application—or parts of an application—change. Not only can these vary between users, or between instances of application launch, but they can change while a user is in the middle of using an application.

Imagine trying to keep control of an airport if Gate 23 was Flight 4438 to San Francisco for some passengers, but for others it was Gate 28, for others still Gate 45. All gates somehow got the passengers to the same flight, but they have to take alternate routes to reach the gate. Oh, and the

tickets in passengers' hands can change on the whim of the airline. Then, what?

This is IT today. We can no longer reasonably create a model of how things "should" work based on some static blueprint and then find out what's anomalous. This means we need tools that let us see the world from the user's perspective and work backward.

Profile, Capture, and Analysis

The perfect tool would allow systems administrators to profile applications from launch to closure and tie this in to log file capture systems at the device, network, and cloud provider level, spitting out a single analysis. Imagine launching your word processor using this tool and having that tool capture every network request, every local system I/O, and every library call made.

Ultimately, monitoring software has to evolve. It has to be not only application-aware, but also able to analyse the whole chain of causality from application to cloud and back again.

That capture data could then be lined up with logs from switches, servers and data made available via APIs from cloud providers (and, in a perfect world, ISPs and backhaul providers). The analysis software could then look for failures. The application tried to look up authentication.cloudapp.com, but the DNS server couldn't find it? Well, that's interesting…

Perhaps more importantly, all our software needs to be aware of dynamicity. From asset management to network discovery, log file analysis to auditing. We need to collect information on how the applications we have behave, when they change, what sorts of changes are normal, and what sorts of changes aren't.

Essentially, we need to be able to detect the difference between A/B testing in a delivered application, a cloud outage, and the malicious injection of code. We need to be able to figure out if something is user error, the network (our fault), or a cloud provider (not our fault, but we have to take the blame for it anyway).

No vendor can build software that can infer what's normal and what's not simply from application behaviour. No sysadmin can divine this from log files. Ultimately, we need the participation of application vendors in this process. We need those vendors to provide, via API, information about how all versions of their applications (including test versions) should behave so that our analysis software can compare it to the data we gather.

Making Do

To my knowledge, no such application yet exists. More's the pity. We're all stuck trying to assemble something like it using the best of the tools we have available.

What's clear is that getting as many logs into the same place and lined up precisely by time is critical. We need to see what applications are trying to do so that we can make rational inferences about where it's gone wrong.

Ultimately, monitoring software has to evolve. It has to be not only application-aware, but also able to analyse the whole chain of causality from application to cloud and back again. In the meantime, let's try to keep the users from digging tunnels to their destination from the back of the bathroom.

Aside: Only So Many Things Fit In My Brain At Once

If our abilities to monitor complex environments and cope with the changes wrought by hybrid IT, increasing expectations, and constant budget pressure need some work, then imagine what life is like for those who measure their compute capacity in acres. Scale changes everything.

One workload is easy to manage. Ten or even a hundred workloads can be handled by a single person without burden. The game changes when we talk about a thousand, ten thousand, or hundreds of thousands or workloads.

There are tools and techniques, technologies, and business practices, all designed to help us cope with scale. But there is a point where we can no longer truly understand what our responsibilities are. Where it will no longer all fit in our minds and we must apply abstractions and metaphors, even when thinking about it only with ourselves.

At this scale, the software matters. All the more so when what we're dealing with at that scale is diverse and complex, involving multiple teams and more users than would fit in the local arena. Size matters; not just to the business, but to our ability to easily understand and manage it.

WHEN IT COMES TO IT MONITORING, SIZE MATTERS

Different-sized organizations have different IT needs. They may be trying to juggle information between silos of specialists, or they may be a small business admin forced by necessity to develop their IT staff into generalists. One problem that all organizations face, however, is getting the right information in front of the right people so that they can make decisions.

I have talked about the differences between how users and administrators see problems and communications problems inherent to the siloed structure of IT teams. I've prattled on about IT monitoring and mean time to innocence.

All of these topics have, in one way or another, touched upon the importance of making sure IT staffs have the right tools for the job. These tools include monitoring software, log file collection, analytics software, and so forth. There is another issue that is tightly coupled with these topics, and that is information overload.

Information overload affects all systems administrators, regardless of organization size. Different organizations, however, are affected differently and there is no one solution that solves all problems.

Case Study

When we're talking about information overload, size is everything. Scale matters in a way that's difficult to comprehend unless you've worked in a really large shop.

In an SMB, one server doing something strange can flood an inbox with a few thousand irrelevant messages that take a systems administrator half a cup of coffee to isolate, fix, and clean up. In an enterprise, a bad patch can cause hundreds of thousands of systems to spontaneously start emitting an endless stream of error messages, flooding all available communications channels, and even bringing organizations to their knees.

I have seen this with my own eyes. A bad patch hit three data centers worth of render nodes: 150,000 machines. Nobody noticed because the patch didn't trigger anything odd in the canary group. However, when the New Year arrived and the leap second was to be applied, everything went nuts.

150,000 servers lost their minds. They started flooding logs with error messages from a service that had fallen over because it didn't understand the leap second. This, in turn, caused regular bleating from the monitoring agent, which considered the loss of the service critical.

The resultant mess crashed the organization's email servers, flooded the local telco's SMS capabilities, and racked up thousands of dollars in SMS charges (this was back in the days when you paid per text). It even resulted in at least one

Information overload affects all systems administrators, regardless of organization size. Different organizations, however, are affected differently and there is no one solution that solves all problems.

administrator's child smashing the company phone repeatedly with a frying pan because he was absolutely convinced that after 15 minutes of constant vibrating, falling on the floor, and slowly creeping towards the family dog, that the phone was possessed.

It was hard to argue he was wrong.

Consolidating Identical Alerts

While such a scenario is an obvious example of information overload brought about by an exceptional circumstance, information overload happens in far more mundane and operating-as-designed circumstances. Again, scale changes everything.

For a small business with 40 servers, there is benefit in having each of those servers notify the administrator when there are patches ready. For an organization with 150,000 servers, that's insane. Most of those are likely to be identical, so is there really a point in having 80,000 servers all say

"patch me" at the same time?

In the case of patches, we have patch management software. System Center (paid) or WSUS (free) for Windows and Satellite (paid) or Spacewalk® (free) for Linux. Servers connected to these systems will have their patches managed centrally. Ideally, instead of emitting complaints about their patch status to an administrator on an individual basis, the administrator receives a summary report on a regular basis from the patch management server.

Even after consolidating the obvious alert spam, administrators are left with the task of separating housekeeping messages from genuine problems.

This idea of consolidation is important. One summary report talking about a similar problem in aggregate, instead of a seemingly unlimited stream of noise is something our poor old primate brains require to do something useful.

If you saw an inbox full of 150,000 seemingly identical emails, it would be rational to highlight them all and delete them. Amidst the inundation of identical alerts, however, there might very well have been something interesting and problematic that went unnoticed. Therein lies the problem.

Signal-To-Noise Ratio

Even after consolidating the obvious alert spam, administrators are left with the task of separating housekeeping messages from genuine problems. Small shops used to be able to do this without any software to help sort one from the other. Large shops have been working on this problem for decades, to varying degrees of success.

Even in small shops, however, the number of workloads under management is increasing to the point where this may no longer be feasible. This is in large part due to the fact that smaller organizations likely only have one or two technical staff members. These generalists must manage everything from hardware to software and even cloud services; there may be fewer workloads total than in a large enterprise, but individuals in the SMB are responsible for such a diversity of management tasks that they can be quickly overwhelmed.

One of the big problems that any administrator can face is the monotony of such messages. Even if what turns up in our inbox is a summary, instead of individual alerts, seeing the same summary every single morning when we log in quickly reduces the relevance to us.

We are either going to log in to the patch management server, release patches to a canary group, pursue feedback, and then release to production, or we're not. Seeing that summary email in our inbox isn't really going to change our routine any, and so we mentally filter it out.

The danger here is that in getting used to mentally filtering out certain classes of email, whether based on sender, subject, or what-have-you, we might miss a genuine call for help. Our patch management summary might, for example, contain information that a group of servers had failed a critical patch, something we might not see until we go looking for it. That can go to bad places pretty quickly.

A New Hope

All is not lost. There is hope. Two similar and closely linked categories of software exist to help us solve the above problem. They are called Alert Correlation and Event Correlation. The canonical startup in this area is BigPanda™. They have made a name for themselves solving exactly the problems described above.

Their success, combined with increased demand from existing customers, has led the big names in monitoring software to build solutions into their suites as well. These range from BigPanda-like alert/event Correlation solutions to more innovative solutions that veer into big data analysis.

Monitoring solutions that monitor multiple applications, hardware devices, and services can not only pull together like alerts and suppress stuff you don't want to see, but they can also be set up to do service-level correlation.

This service relies on these applications, which rely on these servers, and these OSes, on this hardware. Don't send me 15 alerts when it all goes squirrely, but give me a summary of what's wrong with the whole related stack.

Alert Correlation and Event correlation are rapidly evolving elements of the monitoring and analysis landscape, but they

have very quickly become absolutely indispensable pieces of the systems administration toolchain. They are one more item we all need to invest in, but they promise to help us regain some semblance of our sanity.

Aside: Why Am I Not Sleeping Right Now?

For all that, there are difficulties in coping with diversity, communication, and even scale in IT. Squaring the circle requires picking a place to start chipping away at our problems and getting to work. Whether the scale and scope of our responsibilities are large or small, understanding what is occurring is a great place to start.

In IT, monitoring solutions are the basic building block of situational awareness. All other solutions we might engage in proceed from this point. You cannot fix something unless you know what to fix, and that means finding software that works for you.

For me, personally, I look for tools that will augment my memory. I need to not only know what is happening, but what has happened. I need the software to keep track of how things are configured and detect changes in the environment.

As I get older, my interest in all-nighters decreases. I want to spend time with my wife, my cats, and my garden. I want a life outside the data center and that begins with moving on from putting out fires to having the right tools to prevent those fires in the first place.

THE ART AND SCIENCE OF WORKLOAD MONITORING

Traditional IT monitoring solutions provide a variety of raw metrics, often with the ability to set thresholds. When a metric moves above or below the desired threshold, an alert is generated. This is a fairly simplistic system with two critical flaws.

The first flaw in traditional monitoring is that the cause of IT problems is usually complex. The alert we receive could properly indicate the source of the problem, or it could be occurring because something in another part of the IT stack has gone wrong, and we're left to unpack it. This is something previously discussed here at length.

The second flaw in traditional monitoring is that administrators don't always understand raw metrics. Misunderstanding can lead to monitoring the wrong thing. In turn, this can lead to inadequate or inappropriate alert generation. Misunderstanding raw metrics can also lead to setting inappropriate thresholds for alerts, or to attempting to fix the wrong elements of the IT stack during troubleshooting.

Hunting for Bottlenecks

Consider for a moment an application that appears to be responding to requests more slowly than normal. We'll put the inadequacy of human interpretation of average response times to one side, and assume that we have some empirical means to measure application responsiveness. Many things could be the bottleneck that is slowing down the application.

The easiest bottleneck to spot and understand is a CPU bottleneck. If the application consumes all available CPU capacity, it's pretty clear that's the bottleneck. If the application is heavily single-threaded, and thus unable to make adequate use of additional processor cores, that will also be fairly self-evident.

In the case of the CPU being the bottleneck, either you can provide additional CPU resources or you can't. In most cases, if additional CPU power can be provided, that's the cheapest route. If the hardware wall has been hit, however, then to solve CPU problems, one needs to dig into the code of the application and start optimizing things.

Customers deploying commercial, off-the-shelf applications don't have this luxury. They can submit feature requests, perhaps even pay for some custom coding, but that's usually where it ends. If the code they are using is open source, they might be able to contribute to the project, but that isn't guaranteed.

Network as a bottleneck is remarkably similar to CPU. There isn't a whole lot that a customer can do to resolve it, short of providing bigger network pipes or putting money into rewriting the code. This latter is highly unlikely to produce any real-world benefits, because when applications are network-bound, it is rarely because they are inefficiently filling the pipes with unnecessary overhead.

So, CPU and networking are the easy bottlenecks to spot and understand. That means the rest are hard.

Storage Headaches

Storage is a complex beast, therefore most people get it wrong. This is because so many different things can impact storage. Because there isn't just one thing that causes performance issues, there isn't just one solution.

Without trying to teach a master class on storage and related problems, there are four primary things to consider when talking about storage: Input/Output Operations Per Second (IOPS), latency, disk queues, and throughput. Each of these things tells us different things about our storage. It is affected by different things, and means different things to different applications.

There are two basic storage request types: lots of little requests and great big long requests. Lots of little requests will drive high IOPS, while big long requests will drive high throughput. Low latency is always good, but it is absolutely critical for heavily random workloads, typical of those applications that have lots of little requests.

Disk queues are an important measure of the underlying hardware. From a hardware point of view, most of the time,

we want storage devices and controllers with the largest possible queue depth. There is no point in our storage sitting idle, and large queue depths make sure that when there is lots to do, operating systems and applications can transact with storage with minimal waiting.

Operating systems don't report queue depth directly. There are many layers where queues can exist, and not all are directly visible to the operating system, especially if one is using network attached storage. This leads us to monitoring disk queue length, which is a measure of how many outstanding I/O operations there are.

Spinning magnetic drives use an elevator algorithm to look at all the pending I/Os in its visible queue and find the optimal pattern for moving the magnetic arm that reads and writes from the media. Flash drives similarly prefer to be able to write entire blocks, because write operations on flash drives require an erase and the block, which contains multiple pages, is the smallest unit of a flash drive that can be erased.

Relying on administrators to understand what all the raw metrics mean, and how they can affect a given application, is asking for trouble.

So, keeping the queues on individual storage media is good. Filling up the queues on the storage controllers all the way could be bad. And what's reported by the operating system as disk queue length is only tangentially related to either of them.

A disk queue length higher than one means that your application is making more requests than your storage can actually deliver. Queue depth is more nebulous. Deep queue filled with pending I/Os are not actually a bad thing, if they are at the right place in the storage stack.

And that's just the basic hardware.

Bottlenecks Come in Layers

Much of the storage in a data center is network-attached, and network-attached storage is affected by network performance. If an application starts to slow down, and it looks like it is storage related, it might actually be the storage hardware and it might not. It all depends on what else you've got sharing that same network, and/or how much network aggregation and contention there is between hosts that might be asking for storage services and the actual devices delivering them.

RAM utilization can also impinge upon storage performance. RAM as a bottleneck is a tricky thing. Applications and operating systems store vital bits of themselves in RAM. RAM is also used by applications, operating systems and hypervisors for caching.

A system that performed beautifully yesterday might today be slowed to a crawl, not because there is anything inherently wrong with the underlying storage layer, but simply because some change reduced the amount of RAM used for some layer of cache. Without that frequently accessed data living in lightning-quick RAM, the application is forced to go to the disk for every request. Every request for data that used to be cached is time that could have been spent by the storage system doing something else.

The important concept here is that of the consequence cascade. A minor and seemingly irrelevant change to something way over here can cascade through multiple layers of technology to make it seem like the problem is over there.

Change Tracking

All of this means that threshold-based monitoring is outdated to the point of pointlessness. In the past, we could investigate and troubleshoot consequence cascades when an organization had just a few dozen applications under management. Today they can easily have thousands.

Relying on administrators to understand what all the raw metrics mean, and how they can affect a given application, is asking for trouble. None of us can keep track of all of that, especially if we don't work with a given application on a regular basis.

When setting up monitoring for an application, it helps to be able to discover, identify, and log all the other elements in a data center that service depends on. That way, when another administrator—or our future selves—comes across it, we don't have reverse engineer the problem every single time to fix it.

Not all monitoring software is up to the task, especially

given how dynamic the modern data center can be. We need monitoring software that tells us when something is out of bounds. We also need monitoring software that snapshots entire interrelated stacks of parameters.

Most importantly, we need software that detects changes to environments and provides us the ability to easily traverse the logs of multiple services, applications, operations systems, hypervisors, and physical devices that each hold some piece of the puzzle.

Conclusion

A document that says "monitoring is important, maybe you should care about that" is as obvious as a document proselytising the importance of backups. Sysadmins don't need to be convinced that monitoring is important. We've all of us earned that knowledge the hard way.

What is a more nuanced discussion – and one that we all too often avoid – is the bit where monitoring is no longer merely of importance to operations teams. Monitoring is increasingly part of feedback mechanisms for composable infrastructure. It can tie in to middleware, be used in analytics, drive purchasing decisions and even be an important part of job performance reviews.

IT monitoring feeds into analytics that gets seen by more than just the nerds. It is almost always part of cross-departmental communications and feedback mechanisms in which operations teams are involved. In short, monitoring is evolving from being a tactical tool used by the IT teams to keep the lights on into part of the economic, social and political fabrics that hold organizations together.

None of this is news to most sysadmins. What we should all take away from this discussion is that there is no one solution, no one approach, no one combination of software or services that will meet the needs of every organization.

Even within a single industry there is a huge amount of variation in how IT is implemented. This makes sense: IT exists to support the business and the business is constantly seeking ways to do things different from their competitors. That's how one gains an edge, entices customers and grows the business.

Throw in the amount variety of choice we now have to choose from thanks to cloud computing and hybrid IT and it's safe to say that no one can know exactly how to meet your needs without taking the time to study exactly what those needs are.

The challenge for IT is that the very nature of IT is always changing. Our data center designs are in constant flux.

The challenge for IT is that the very nature of IT is always changing. Our data center designs are in constant flux. Not only do our monitor solutions need to be regularly reviewed and altered to keep up with these changes, but in most cases our approach to assessing the adequacy of our monitoring capabilities and even how we interact with the tools available needs to change.

We need to be part of driving the roadmaps of the tools we rely on. This can mean participating in IT communities, and otherwise being proactive about ensuring that the third-party solutions upon which we have become dependent are aware of our needs and able to respond.

This widens the web of interaction to include vendors, and possibly other community members. It increases the need for more of those dreaded "soft skills", like negotiation and schmoozing. Unfortunately, this is inevitable.

As IT automation and Software Defined X start to handle more and more of the low-level work of keeping the lights on, the attention of the systems administrator is pushed higher up the stack. Perversely, this makes us ever more reliant on monitoring, as we interact with the nuts and bolts of our own datacenters less and less.

Systems administration is a profession; one as critical to smooth functioning of today's society as that of doctor, accountant or lawyer. As with any profession the right tools often make the difference between a task accomplished and an accomplishment worthy of pride.

SOLARWINDS SOLUTIONS

Managing and maintaining IT infrastructure isn't easy. Doing so successfully begins with effective monitoring, with an increasing scale quickly driving requirements for more in-depth troubleshooting capabilities. Having the fastest hardware or the right application to solve a business need isn't enough—administrators must also have the visibility into their infrastructure to solve problems before and as they happen. Fortunately, SolarWinds has solutions to these problems.

Exploring solutions to today's IT challenges begins with understanding the SolarWinds® Orion® Platform. The Orion Platform consists of the Orion Console, which installs on Windows Server® and serves as the management nexus for many of SolarWinds products. Individual products attach to the console like modules to a centralized management application.

The Orion Console stores and processes data from the individual applications, offering more insight into IT infrastructure than would be possible from standalone solutions. The console includes a sophisticated analytics engine that enables real-time monitoring, scheduled reporting, and historical viewing.

The Orion analytics engine has been optimized to handle data at significant scale. SolarWinds applications collect large amounts of data. Taking advantage of this for real-time usage, the Orion Console can serve as the center of Network Operations Center (NOC) display, with various SolarWinds applications having modes customized for exactly this purpose.

NOC dashboards are customizable and interactive. They can be configured with multiple tabs, each with several columns capable of showing resources, analyses, and other information at a glance. They can also serve as launching points for interacting with infrastructure to remediate issues.

Real-time monitoring is helpful to any IT operations team. However, any good IT infrastructure solution needs to offer more than that. Monitoring all collectors from even one SolarWinds application in real-time would be nearly impossible; there's too much data collected for humans to understand. This is where the Orion Platform's reporting and analytics capabilities shine.

Scheduled reports can offer an aggregate view of collectors that is easy to consume and can run at off-peak hours. Many report categories can be quite resource intensive, making this sort of scheduling a necessity. Regular reports are very useful when doing trend line analysis or if you're in a large environment. Being able to look at historic data is useful for solving recurring issues. It is also useful for solving transient issues, when you need to examine information from a great number of data sources in order to isolate the problem.

Data collection demands are not going to diminish. Clearly, businesses are always looking for ways to improve efficiency. For the foreseeable future, hybrid IT will—from a regulatory compliance, collection, and analysis perspective—impact day-to-day operations and retention requirements. All of this will be amplified by upcoming IT security automation technologies that are just now entering the market.

Sight Across Sites

Perhaps the most vexing challenge facing today's systems administrators is the rise of hybrid IT. A hybrid environment removes some portion of the infrastructure from the direct control of the operations team. The best they can do is monitor the managed environment and respond to outages or performance issues with failovers, redirects, load balancing, or contacting the provider's support.

Having the right information at hand before choosing which path to take towards resolution is critical, no matter which service is causing the problem. It is doubly important when one is going to confront a provider's support team and ask them to resolve a problem.

SolarWinds NetPath™, a feature in Network Performance Monitor, is a critical technology for helping administrators solve hybrid IT challenges. It allows for the monitoring and analysis of public cloud solutions side by side with on-premises or service provider-hosted solutions. Identical workloads can be tested in multiple environments in order to quickly determine which environment is underperforming, and why.

PerfStack™ Performance Analysis, a feature of the core Orion Platform, is another useful tool in this situation.

PerfStack allows systems administrators to drill down through data provided by multiple collectors at the same time to see what might have caused a problem.

Administrators could, for example, line up storage latency and throughput alongside network latency and throughput with CPU and RAM utilization for multiple different workloads that make up a single service, and at multiple different points along the network chain between the service components and the end-user. This allows administrators to not only to see where performance issues lie, but quickly see performance cascades, allowing administrators to address the root cause of issues instead of ineffectually trying to alleviate symptoms.

There are many other SolarWinds products that are useful in solving hybrid IT problems. These include Network Performance Monitor, Server & Application Monitor, and Web Performance Monitor, which do what they say on the tin. Librato® can also be useful, as it collects data on multiple end-user experience metrics and provides an easy way to share and annotate this data between different IT teams.

PerfStack allows systems administrators to drill down through data provided by multiple collectors at the same time to see what might have caused a problem.

Heterogeneity

As our IT environments become more complex, the ability to see all the different components that make up our IT stack becomes increasingly critical. In order to properly diagnose what's going on, we need multiple administration tools with collectors—and sometimes management capabilities—located on every site where our IT operations are in use. This includes premises under our control, service provider hosted infrastructures, as well as the public cloud.

Our tools need to talk to the multiple hypervisors, OSes, applications. They need to understand networking and be able to tease out performance issues from those networks, even when they can't the kind of in-depth reporting that you'd get from on-premises switches directly under IT's control.

Virtualization Manager allows IT teams to go beyond monitoring and take direct control of their virtualization environments, enabling alert-to-resolution problem solving in a few simple clicks.

Scheduled reporting and automated documentation are important, but most of all, we need to create a database of what is where, what it's up to, and how things are changing. IT has to also keep track of many things across too many sites for any one administrator to remember it all.

In addition to the SolarWinds tools mentioned above, Storage Resource Monitor is worth a look for complex environments. It offers insight into storage solutions from multiple vendors, acknowledging the reality of today's heterogeneous data centers.

Virtualization Manager allows IT teams to go beyond monitoring and take direct control of their virtualization environments, enabling alert-to-resolution problem solving in a few simple clicks.

Those organizations with in-house developers coding custom applications should look to TraceView™, a SolarWinds application performance monitoring solution. TraceView allows developers to trace performance issues down to individual blocks of code, and supports Java®, .NET, Node.js®, PHP, Python®, Ruby, Scala, and Go.

SolarWinds Orion Console really starts to shine in complex environments. While the various applications offered by SolarWinds can be licenced individually, the real value comes from their integration. Combining SolarWinds applications allows an IT team to progress from a firefighting effort that views each problem in isolation to a fire prevention effort that considers the whole of today's dynamic, heterogeneous environments.

The People Problem

Of course, without the right communication and documentation tools, all IT management efforts will eventually fail.

In addition to enabling different IT groups to work together, IT management solutions need to integrate with ticketing software and provide easy access to the complete chain of evidence for a given problem.

Web Help Desk® does what the name implies, and any IT department of appreciable size will need the ticketing and asset management it provides. SolarWinds Security Information and Event Management solution digs into logs to helps users solve security and compliance issues across numerous vendors' products.

Of course, not every IT team has the same needs, nor should everyone within an organization have access to the same information. In many cases, these restrictions are mandated by regulatory regimes. Fortunately, many of SolarWinds products incorporate robust Role-Based Access Controls (RBAC). Where SolarWinds products can affect infrastructure directly, restrictions can be enabled and fine-tuned such that different IT groups can access the information they need without compromising security.

SolarWinds also has offerings that can exist independent of the main Orion Console—for example, if you don't want to host your logs on-site, look into Papertrail™. This is SolarWinds' cloud-hosted log management product and offers a number of ways to analyse data and create alerts. This flexibility allows SolarWinds to serve IT teams and organizations of all sizes.

Regardless of your IT team's size, remote support is critical. SolarWinds Dameware® Remote Support, Dameware Mini Remote Control, and Mobile Admin® products help administrators resolve issues on end-user devices, remotely hosted services, and even manage critical tasks from their mobile device.

SolarWinds is focused on enabling communication between IT teams. These teams may need to communicate with one another from sites located around the world or with technicians in the field. IT teams may even have to reach out to a domain expert after hours, or who is on vacation, making mobile access—and the ability to prove it's worth interrupting them—an important feature.

Living Large While Staying in Charge

If SolarWinds' product lineup seems expansive, it is because the scope and scale of IT itself has grown. Where once the IT architecture of even the largest enterprises could be held in the mind of a single individual, today even the best of us can only understand the scope of our deployed technology in aggregate. Small organizations can get by with simple mapping and asset tracking solutions, while large organizations can only describe the number of endpoints and workloads under management by using sufficiently large exponents.

It's easy to monitor products and solutions one at a time. IT infrastructure, viewed piecemeal, makes a deceptive sort of sense. Unfortunately, what happens with one piece of the puzzle affects the next and the next, and before you know it, integrated management and analysis solutions go from "nice to have" to being absolute requirements.

Analytics isn't easy. There's more to it than getting as many logs into the same place and lined up precisely by time. Extracting signal from noise is a constant battle; no sooner do we collectively get a handle on things, only for a whole new market of IT products to come out, and we all need to play catch up one more time.

This is where SolarWinds comes in: as a solution whose whole is of greater value than the sum of its parts. One application can solve one pain point. Multiple applications can work together through the Orion Console's alerting customization to enable alert correlation and event correlation that differentiates between minor annoyances that can wait and real-world issues that that need to rouse someone from bed.

Combined, SolarWinds applications can identify bottlenecks, allow in-depth examination of consequence cascades, and make change tracking and trendline analysis possible. All of this is useful, even in small businesses. It's absolutely vital in larger organizations.

With the Orion Platform, SolarWinds can help organizations manage their journey and enable organization growth without having to stop and retool IT every few years when it spirals out of control.

Check out SolarWinds today at www.solarwinds.com.

solarwinds