AVADA SOFTWARE

Proactive vs. Forensic – Middleware Health Monitoring

Avada Software's monitoring and management solution **Infrared360** fosters a proactive management approach to your middleware infrastructure.

Though it's difficult to compare the importance of IT monitoring with something that is life threatening, the two disciplines share many techniques and can learn from each other.

WHEN WE THINK ABOUT HEALTH MONITORING, it usually involves calories, blood pressure, weight, and pulse - measurable units that indicate general health. People in generally good health, though, normally have less chance of hidden problems because their system is behaving well - a reflection of their good health metrics.

When we think of the word 'forensic', it's usually associated with the discovery of why or how someone died. Some of this can be attributed to television news and popular series like *CSI* and *Unsolved Mysteries*. The term *forensic*, however, can be applied to many disciplines, like accounting or law. Its origins are actually Latin, translated as *the scientific method of gathering and examining information about the past, which is then used in a court of law.*

It is easy for most people to understand the value of health monitoring. We all want to be notified of issues that can cause serious medical conditions and debilitate us, if not worse. And though it's difficult to compare the importance of IT monitoring with something that is life threatening, the two disciplines share many techniques and can learn from each other.

Medical health awareness has grown because of technology advances. Technology that was formerly the property of few has become accessible by many For instance, a person no longer needs to be in a doctor's office or hospital to have their vital signs checked. The cost and miniaturization of health devices has made them affordable, accessible, and personal. This has been a very good thing, since it has led to proactive determination of problems. You don't have to be lying in a hospital bed to see the blips and graphs that indicate your immediate future.

If you have a health condition, or are starting a new exercise regimen, or are an advanced or ardent athlete, it's likely that you have either a device or any number of Smartphone apps that display and record data like heart rate, respiratory rate, and temperature.

Some gadgets or apps are even more specialized. If you are a runner or cyclist, for instance, you can find out your progress as far as distance, pace, or elevation. You can also send this data to the cloud and access it at your leisure. It may seem a bit over the top to have your vitals up on the cloud, but if used for the right reasons, the information could help indicate where your health is failing! Suppose you suddenly collapsed? If your medical facility could access up to the minute data, they'd have a clear indication of what just occurred by inspecting the data. They could then send it to the correct personnel, and give instructions to help save your life, when every second counts.

The reality is that a serious issue in the IT world can cost way more than what it costs to prevent it.

This reactive approach has some drawbacks. For example, most people learned they had a heart problem only after they had a heart attack! Being reactive is better than nothing, in this case, because having an immediate alert notification and taking corrective measures in a short amount of time can save a person's life. Yet it's still not quite the same as proactive. The ideal situation would be if we could get immediate signals that indicate that unless a change is made to your 'system', you will likely have a heart attack.

Devices are not yet accessible (due to cost) nor miniaturized enough to provide monitoring of deeper issues like 'are my arteries clean?', or 'are my cells healthy?'. Healthcare insurers, moreover, do not normally cover proactive care, so again costs make it difficult to do what's best as far as health monitoring. The world of gadgets is well on the way to helping change this model. Some insurers realize that preventive care costs less in the long run and are changing their models as well.

In the IT world, it's not much different. There are hundreds, if not thousands, of IT executives who perceive proactive care as being too costly or that even reactive care is too costly. The reality is, however, that a serious issue in the IT world can cost way more than what it costs to prevent it. In the current corporate climate of bottom-line spending, nobody wants to show an incremental cost.

In today's business climate of downsizing and cost cutting, IT departments do not have the people to watch each application and asset, like nurses in an intensive care unit. These people have a list of duties to accomplish. Many IT operations people are no longer onsite where the systems reside, so there’s no longer any reason for the flashing red text on the big screen approach. Many people now work from home or are on the staff of an outsourced group, perhaps in another location, state, or country. Many IT sites run what is called 'lights out' operation centers. Yet there is still a need to quickly resolve incidents. The best practice is now to send a notification to the required supporting people as quickly as possible with possible suggestions for tactical actions to correct the problem.

REFLECTING ON REACTIVE

Even in the early days of the mainframe, there were always tools that monitored CPU usage, memory, disk space, or network activity. Finding failure at any of these levels usually meant that the warning or error was displayed on a large monitor or screen. CIOs used to give tours of their operations center; sometimes called a fishbowl because people would be busy behind the glass that housed all the monitors and flashing yellow, orange, or red text that indicated a problem. I remember spending many a day inside a fishbowl. Only thing is they usually didn't allow food in the fishbowl, unlike real fish. At least we all got to go home at the end of the day!

Most of that type of monitoring was considered reactive. When an incident occurred, the operational staff noticed it up on the screen and tried to solve it, if possible. Most often, the issue escalated to a systems specialist in the area of expertise, whether at the OS level, network level, or device level. It was rarely a preventative measure, where the operational staff would notice a trend before it would likely cause the red flashing text.

Clearly, being proactive and avoiding forensic is the better way to go. This is not only from a time, pain, worry, and resource perspective, but cost as well.

Both proactive and forensic problem analysis are needed, depending upon the situation. But clearly, being proactive and avoiding forensic is the better way to go. This is not only from a time, pain, worry, and resource perspective, but cost as well.

The following real-life scenario illustrates this example: A financial firm was clearing transactions for their business partners. These transactions were high value monetarily (think in terms of institutional trades). Without a proactive way to notice environmental problems within all the touch points of a transaction, the company did not notice that they did not completely process some of these transactions after close of day. Completely is the operative word here. The transactions actually were initiated and received. They were processed by the middleware delivery environment.

Unfortunately, the next morning, a partner firm asked why they didn't get a confirmation on those trades. The research showed the transactions were delivered. An indicator showing that the application process was triggered made everyone assume it had indeed worked, but something else was in error.

What had occurred was a bit like the mailman having someone in your home sign for a package and then that person throws it under the bed, the item never to be seen from again – or at least not until somebody stumbles upon it.

At this point the IT department was doing forensic analysis. They were going back to look at each step of the transaction to see where things went wrong. It wasn't until many hours later that they realized the transaction result was never persisted (written) to a database. Yet another application would have used that data as a prompt to send the clearing message back to the originator of the transaction and trigger an electronic transfer of the actual funds.

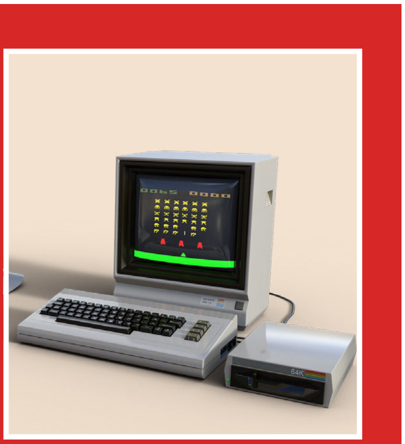
The problem was finally resolved. The people in the transaction chain, though, had major issues. Since the sender still had the funds, who was responsible for the interest on the funds? Who was responsible for any change in price while the trade remained in limbo?

PRECISION ALERTS

We took this approach at Avada Software in the very first version of the Infrared360 middleware monitoring solution. Upon problem identification, Infrared360 gets the information to the responsible parties as quickly as possible, wherever they are, and on whatever device they use – email, SMS, external trouble ticket system, etc. The days of people in the fishbowl staring up at the red flashing text are long gone, or should be. With the proliferation of systems, finding the red graphic on such a screen would be like playing a game of "Where's Waldo". Since most IT people are busy with other duties, a simple SMS text produces the same result. It doesn’t matter if it's 1am and the responsible party is in Asia, text them! And provide relevant and helpful information about the problem. There's nothing worse than getting a notification and it only says 'error' – not a great start! It wouldn't be very helpful if a nurse got a beep at her station and it said 'patient error'. From day one, Infrared360 included a hyperlink in that email or SMS to a display of the problem, not the big screen with a list of all the problems.

Who was going to pay the fines levied by the watchdog organizations for missing the time agreement on completing the transaction? What type of damage to the business relationship occurred? Although the company now knew what the issue was and could set up prevention methods moving forward, the symbolic patient here had a stroke!

This brings up the other issue in being proactive: the holistic approach. It is sometimes difficult to cover all aspects of a system, whether human or otherwise. But if there is good knowledge of the related functions, then you can observe, monitor, and isolate when something might become a problem. For example, if 'Elizabeth's head swells to the size of a melon', it is then easier to analyze the problem if you knew that Elizabeth was allergic to peanuts!



Two such paradigms where most monitoring applications fall behind in modern distributed environments are the Forensic resolution approach and the unsecured siloed worker paradigm.

Much the same is true in our IT example above. If we knew there should be trade notes in that database (there were 'none' btw, so it wasn't just one transaction that was lost) and set an alert if #of transactions <X, then we would have known that the transactions didn't complete and could have corrected the problem. Correcting the problem may involve the ability to securely work across silos, but that's another paper. That would have allowed the transactions to complete before end of day as they normally would.

Hopefully, the aforementioned examples set a good platform for the discussion of proactive vs. forensic monitoring that follows.

Most technical professionals are aware of the need to monitor their IT services, and almost all organizations have multiple software products and tools in place to do so. But according to Gartner, infrastructures have evolved beyond traditional monitoring products and paradigms. Two such paradigms where most monitoring applications fall behind in modern distributed environments are the Forensic resolution approach and the unsecured siloed worker paradigm.

The old, siloed paradigm school of thought uses a large pool of administrators, each for for different types of IT environments that rely on logs and dumps. While these methods are good if you need to forensically analyze a problem, it is the slowest of methods and an inefficient use of skilled resources.

Doing such searching for each and every problem is tedious and time consuming. It's also hard to find 'bugs' that way. It's not like there is an illuminated ERROR 101 sitting in that heap of data. If you are an administrator or support person, this information is not even accessible to you.

See It Live
Get a no-obligation live demo of Infrared360.
Click here or email us at info@avadasoftware.com

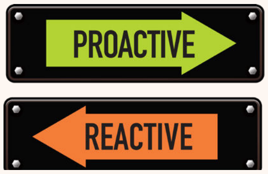
An effective monitoring and management solution under a modern paradigm – one that will proactively prevent these issues – requires the capability to securely collaborate on problem resolution.

And though specific platforms have gotten more mature and helpful with plain English error messages (as opposed to the abstract codes many used for years), this information is gathered after the fact. The world of web services and integrations of applications across multiple tiers has made this much more complicated. Support personnel and administrators can no longer look in one place for information about a failure. Some IT executives believe they have solved this problem by using many point solutions as opposed to an integrated solution platform. In fact, this is an illusion because such solutions lack collaborative features and do not encompass the entirety of the transaction's environment. The biggest reason for this lack of collaborative capability is security. Most solutions can't offer a way to grant or limit access to specific objects based on role and permission levels.

In the transaction example above, the person who monitored the platform would not have seen any errors. The person who monitored the middleware layer would not have seen any errors. A database person likely would not have seen any either because having no data in a table is not unusual. The issue has to do with the context of the problem. The middleware transaction person would have understood that situation, but did NOT see it because they did not have visibility to all of the touch points of the transaction.

An effective monitoring and management solution under a modern paradigm - one that will proactively prevent these issues - requires the capability to securely collaborate on problem resolution. To do that it must be able to permit or limit access and even visibility to objects (see the sidebar for more details on this).

The approach that Avada Software took with its first product release, giving selective visibility to certain people for selective transactional environments, was a major change in the way corporations handled management, administration, and monitoring of these resources. The thought being that 'if people can just SEE all the touch points of the transaction environment for their line of business, and not outside of it, if they are not able to take any action in that environment, it would be a big advantage because now we can place more eyes on the problem. Therefore, more pieces of the problem are available to those eyes without anyone worrying about someone touching it, without the need to provision tools on each desktop, and without the need to change platform specific security access on 'each' platform for 'each' person that needs visibility.



Back to the point of proactive vs. forensic, seeing more is still not enough. Staffs need to get proactive alerts and notifications of these systems. Because transactional systems now span many operating systems and many technologies, a new set of management products arose to try to address these issues. These products rely on capturing and analyzing tons of data in either logs or databases. This approach fits into the buzz about 'big data', but these solutions are still forensic in nature.

See It Live

Get a no-obligation live demo of Infrared360.
Click here or email us at
info@avadasoftware.com

If the intent is to ensure you are more proactive rather than reactive, the "big data" approach is slower... and more costly.

This is not to dismiss other approaches, but there is more than one way to bake a cake. If the intent is to ensure you are more proactive rather than reactive, the above big data approach is slower. Realize that the data needs to be captured and written somewhere. That is an I/O (a write to disk); one of the more performance costing actions a program can do. Then the logs grow, so parsing something large is also a performance cost. Then the logs need to be accessed by something, which entails a security aspect. At this point that found information still needs to be sent to the appropriate parties.

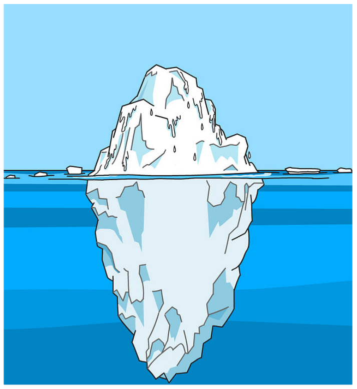
Another technique is to 'capture' every transaction. This, too, is an expensive proposition. While a transaction will most likely span different OS's and different technologies, it is not an easy proposition to tie together an http call → to an application server transaction → to a transformation engine transaction → to an enterprise messaging transaction → back to a transformation system → and out to a web services transaction.

This is especially true if there are hundreds or thousands of such transactions flowing through these environments and they are all being routed differently, depending upon the context of the transaction (for instance coming from supplier A vs. supplier B). These different tiers also do not share any unique signature between them except for the content of the message! It is never a good idea to search through content in order to identify a unique piece of data, because it is a performance issue.

Imagine ALL those transactional flow components each doing an I/O on ALL those transactions each time. There are systems that process 100's of transactions per second! That's a LOT of I/O to impose upon them. Added to that quandary is the fact that there is a foreign entity sitting there in your business transaction chain, and you are left hoping there is no security hole there, or that an error in the entity doesn't cause a transaction problem itself (your pulse monitor should not affect your pulse)!

Even if you decide on that approach, that is an awfully large amount of data to deal with! A conservative estimate of [100 trans/sec * 60 sec * 60 min. * 24 hrs * 7 days * size of data (4mb average)] = 241,920,000,000,000 bytes of data per week to process just one transaction type. Then someone has to figure out what the unique identifier is in order to tie them together, and then do that for each transaction type. Configuration of these scenarios is not a simple task. Once implemented, how much performance overhead is incurred on a transaction system where you are measuring performance for reasons of keeping up with SLAs (service level agreements) and supplier expectations?

See It Live
Get a no-obligation live demo of Infrared360.
Click here or email us at
info@avadasoftware.com



If we can avoid a bigger problem by getting warnings about thresholds and issues for specific environments, then we can avoid the need for the forensic method.

Going back to the healthcare metaphor, this would be analogous to doing every bodily test every day and attempting to tie the results together. There is a general feeling that IT organizations collect data for the sake of collecting data. There is so much of it that nobody has the time to look at it or make any sense of it. The question becomes, is the overhead and resource consumption required to set up, configure, and analyze this data worth the cost?

In a comparative health care example, would anyone personally choose to go through that battery of tests unless it was absolutely necessary? Wouldn't it be preferable to receive a warning notification in advance from a Smartphone stating, "You just consumed 3000 calories of fat, your cholesterol is 230, your blood pressure is 180/110. You are at risk of a stroke!"?

As the world of Internet of Things (IoT) expands, there will be just too many devices, systems, technologies, etc. to collect and analyze each and ever speck of data, and then assemble it all, in order to come up with a reasonable response time to take action. Sifting through tons of captured data is not the way to get proactive health alerts. Proactive alerts warn of an impending issue. We need to have the fastest and most efficient methods possible to relay that information to the proper people or systems. If we can avoid a bigger problem by getting warnings about thresholds and issues for specific environments, then we can avoid the need for the forensic method. Because when it comes to TIME, forensic is usually a bad thing. It means the problem has gotten to the point of deeper analysis or worse-- the transaction is dead and gone. That means we'll need to wait until it happens again and put the right watchdogs in place.

In the transaction world, we *could* decide to trace all the transactions and store all the data and tie it all together. But let's use an alternative example. If I'm alerted to a train track problem that will take that track out of service from 2pm to 3pm, between the Greenwich and Stamford stops, on the Washington DC to Boston line, then I can ASSURE you that no passengers ("trainsactions") will pass through that location between those times. I don't need to wait or all the tracking data and passenger logs and train arrivals, etc. I can take advanced action to hold back my trains or reroute my 'trainsactions'. Just from that warning, I can immediately dispatch a track repair team, and I can inform you which alternative route passengers can take and when each would arrive at their destination. And, with a little skill and the right administration solutions, it can all be automated. I do not need to, nor desire to, inflate costs to analyze a

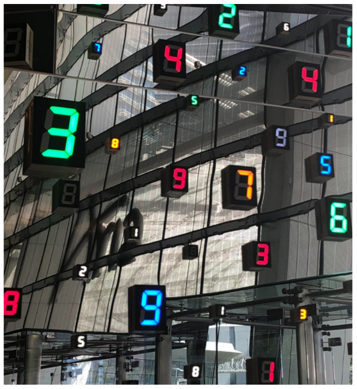
See It Live
Get a no-obligation live demo of Infrared360.
Click here or email us at
info@avadasoftware.com

large repository of information to know that I shouldn't have sent 'transactions through those stops.

Latency Kills Proactive

There is one additional point that really needs to be highlighted here. As mentioned earlier, Gartner has cited that infrastructures have evolved beyond traditional monitoring products and paradigms. We talked about some of those paradigms, but there are several reasons products haven't kept up either. Products that still rely on expensive tracing is one, but another is the approach most solutions take to what they refer to as real-time monitoring. And, this directly impacts the ability to proactively manage, especially in today's low-latency transactional environments.

Most monitoring solutions that claim to provide real-time monitoring are being a bit loose with the term. What they actually do is write data to logs and then monitor those logs and trigger alerts based off of averages in those logs. At Avada Software, we have seen real live scenarios where companies using this type of monitor have had some serious consequences because of relying on these averages inserted latency in the alerts and notifications. In one case even a 15-minute delay in getting alerts on certain critical thresholds caused huge costs in forensic man-hours and missed service level objectives. Clearly this is not good proactive management.



For true proactive management, your monitoring solution needs to have true real-time monitoring and not monitoring and alerting based on logs and averages.

For true proactive management, your monitoring solution needs to have true real-time monitoring and not monitoring and alerting based on logs and averages.

In conclusion, the best-case scenario to deal with an impending problem is to receive a proactive warning of the issue, a description of the issue, along with some steps for handling the situation, in order to mitigate a larger problem. Even better would be if the system can automate all of the notification and steps to take. Then you'd be made aware, and can decide on automated vs. manual intervention, let that process happen, and if you need to log anything, it would just be that situation A occurred at 2:45pm and was remedied at 2:49pm.

- IT executives should take advantage of the proactive approach to save time, risk, resources, and money (TRRM). The overall cost is less both in terms of initial setup as well as TRRM in the long term. The upfront cost of the 'gadgetry' is very affordable and scalable as well.
- For SMB that means they have access to 'professional' gadgetry not previously affordable.
- For large corporations that means they can scale the gadgetry with major savings for time, risk, resources, and money vs. other approaches.

See It Live
Get a no-obligation live demo of Infrared360. Click here or email us at info@avadasoftware.com

A proactive approach to your middleware health management is only possible with the proper monitoring solutions. Check out our **Monitoring Widely Distributed Environments Without Losing Focus** paper to see how an effective middleware monitoring and management solution will help you in your Enterprise Messaging and Middleware environments.

Infrared360®, a single portal providing total administration, monitoring, testing, auditing, analytics dashboards, and self-service for cloud, on-prem, or hybrid environments.

See It Live

Get a no-obligation live demo of Infrared360.

Click here or email us at **info@avadasoftware.com**

ABOUT AVADA

Avada Software specializes in Enterprise Middleware solutions. Founded by some pioneers in SOA, MQ and J2EE technology, Avada Flagship product, Infrared360®, is a holistic & innovative private cloud enabled portal providing administration, monitoring, testing, auditing & statistical reporting for Enterprise Middleware including such as IBM MQ®, Apache Kafka®, and TIBCO EMS®, Application Server providers such as IBM, JBoss, & Apache, and SOAP & REST based web services. Infrared360 is a single web application, yet scales to thousands of endpoints without deploying anything (no agents, no scripts) to those endpoints. Using Secure Collaboration™ and delegated administration, the portal uniquely provides different business units or even different application users delegated virtual environments in which to work.

AVADA SOFTWARE

100 Enterprise Drive. Suite 301

Rockaway, NJ 07866

1 (862) 781-6159

info@avadasoftware.com

+1 973 6971043

www.avadaSoftware.com

For more information: email us at info@avadasoftware.com