Chapter 1 : OS Monitoring Tools | Operating System Monitoring Tools | Opsview

Efficient SQL Server performance monitoring includes monitoring of operating system, SQL Server, and database performance. In How to monitor your SQL Server instances and databases, we presented tools for monitoring the latter two performance metrics, In this article, we will present tools that provide operating system performance monitoring.

Obviously, one of the most important components is the operating system itself. Lync Server supports x64 editions of: By detecting, alerting, and automatically responding to important events and performance indicators, these management packs reduce resolution times for issues and increase the overall availability and performance of systems that are running the Windows Server operating systems. Apart from relevant Windows Server management packs for System Center Operations Manager, the following system tools and resources can be used to monitor the health of the operating system depending on the operating system version. Windows Resource Monitor is a powerful tool for understanding how system resources are used by processes and services. In addition to monitoring resource usage in real time, a Resource Monitor can help analyze unresponsive processes, identify which applications are using files, and control processes and services. Reliability Analysis Component is an in-box agent that provides detailed experience information on system usage and reliability. By exposing Reliability Analysis Component through a WMI interface, developers can monitor and analyze applications, increasing reliability and performance. The Reliability Analysis Component also keeps track of any important changes to the system that are likely to influence stability, such as Windows updates and application installations. It provides a graphical interface for customizing performance data collection and Event Trace Sessions. It also includes Reliability Monitor, an MMC Snap-in that tracks changes to the system and compares them to changes in system stability, and it provides a graphical view of their relationship. Data Collector Sets Reliability Monitor Wizards and templates for creating logs Data Collector Set groups data collectors into reusable elements for use with different performance monitoring scenarios. After a group of data collectors are stored as a Data Collector Set, operations such as scheduling can be applied to the whole set through a single property change. Windows Reliability and Performance Monitor includes default Data Collector Set templates to help system administrators begin immediately collecting performance data that is specific to a server role or monitoring scenario. The home page of Windows Reliability and Performance Monitor is the new Resource View screen, which provides a real-time graphical overview of CPU, disk, network, and memory usage. By expanding each of these monitored elements, system administrators can identify which processes are using which resources. In earlier versions of Windows, this real-time, process-specific data was available only in limited form in Task Manager. Reliability Monitor calculates a System Stability Index that reflects whether unexpected issues reduced the reliability of the system. A graph of the Stability Index over time quickly identifies dates when issues began to occur. The accompanying System Stability Report provides details to help troubleshoot the cause of reduced reliability. By viewing changes to the system installation or removal of applications, updates to the operating system, or addition or modification of drivers side by side with failures application failures, operating system crashes, or hardware failures , a strategy for addressing the issues can be developed quickly. Adding counters to log files and scheduling their start, stop, and duration can now be performed through a Wizard interface. In addition, saving this configuration as a template enables system administrators to collect the same log on other computers without repeating the data collector selection and scheduling processes.

Chapter 2 : Monitor Resource Usage (System Monitor) | Microsoft Docs

Performance Monitor (perfmon) Performance Monitor is the second most common operating system performance monitoring tool for Windows. Performance Monitor acts as both a real time and log-based performance monitoring tool for operating systems, so only the real time portion of the tool will be discussed in detail in this section, and the logging portion will be discussed later.

Introduction Computer operating systems monitor their resources constantly. In a system, processes are the main resource owners, and as such, most monitoring is done at the process level. This information is used by operating systems while they are running to perform effective memory management, scheduling, multiprogramming, and many other important decisions. In addition, performance monitoring is useful while developing and refining systems, and it provides user support during everyday operation. Records of operating system and process performance can be used to quantify changes to the system and allow accurate comparisons to other systems. They can also be used to predict the performance of similar systems and what type of performance gains may be expected in the future. There are a large number of operating system and process monitoring tools available. This paper will present a survey of some of the most common tools used for operating system and process monitoring. These tools are primarily divided into two main categories: Real time monitoring tools are concerned with measuring the current system state and provide up to date information about the system performance. Log-based monitoring tools record system performance information for post-processing and analysis and to find trends in the system performance. In addition to these categories, this survey will look at Windows-based tools and Unix-based tools, since the tools are very different for these two common operating system types. Back to Table of Contents 2. They sum up the performance for a particular factor with a single number. Typically, these tools rely on system calls that are built into the operating system to extract the performance readings. Because these calls are built into the operating system, they do affect the system performance, sometimes significantly. They are also very difficult to change since the operating system source code is not usually readily available. I will describe a number of the most common operating system and process performance monitoring tools for Windows and Unix systems in the following sections. Task Manager was introduced with Windows NT and provides a fast look into the current system state [Moore]. It shows all applications one or more processes running within a single application context and their state, all processes and some of their most frequently used performance measures, and some general system performance measurements. Newer versions also display networking performance measurements. All measurements are made by directly calling functions in the operating system to retrieve system counters [Moore]. In addition, Task Manager gives users the ability to control the system by affecting the running process. This is the function general computer users typically use Task Manager for when an application or process enters a "hung" state due to errors in the code and cannot be exited normally or when a process is hogging the CPU. The first is the Processes tab, shown in Figure 1. The Processes tab shows the current memory and percentage of CPU usage of every process running on the computer as well as the total CPU and memory usage of the system. While this is not a lot of information, it is a very good first indicator when a process is taking too much of the CPU or has a memory leak. It is easy to use and even allows sorting by name, user name, CPU usage, or memory usage. The second Task Manager tab that presents operating system performance data is, not surprisingly, the Performance tab, shown in Figure 2. A short history is shown on a graph, but once again, none of this data is logged for comprehensive analysis. Task Manager - Processes Tab Figure 2: It is highly integrated into the operating system, and it is not designed to log any of its performance measures for performance analysis or system evaluation [Moore]. However, it is an invaluable tool in monitoring and adjusting the processes running on a computer. Task Manager is also the standard for real time operating system and process monitoring tools for Windows systems that all other tools must be evaluated against, because another tool must provide some features not available in Task Manager for it to be considered useful. Performance Monitor acts as both a real time and log-based performance monitoring tool for operating systems, so only the real time portion of the tool will be discussed in detail in this section, and the logging portion will be discussed later. Like Task Manager, Performance Monitor measures performance by making system calls to retrieve system counters, but Performance Monitor makes these calls via a performance library that also supports logging of the counters. Unlike Task Manager, Performance Monitor provides an interface to monitor any selection of a huge set of system counters on a graph in real time, rather than just the limited set Task Manager uses. Counters include things like percentage of processor time, thread count, page fault rate, memory size, and elapsed time for processes. Similarly, there are counters that provide state for threads, the processor, the system, network interfaces, memory, physical disks, and many others. This level of detailed information available for monitoring from Performance Monitor is very extensive and makes Performance Monitor ideal for monitoring resource usage and performance of almost all pieces of a Windows system. The main window for Performance Monitor, shown in Figure 3, is a graph of all selected system counters updated in real time at a specified rate. Almost everything on this display is customizable from whether it is a graph or a histogram to the colors assigned to certain counters. The current reading and statistical information is displayed for the selected counter as the graph updates. Only one counter can be selected at a time for displaying numerical data, but any number of counters can be included on the graph at once. This example shows the counters and options available for a process. The list on the left lists all of the available counters for a process that can be selected for monitoring. The list on the right shows all instances of a process object that can be selected for monitoring. Performance objects that can be monitored, detailed in Figure 5, include many objects other than processes. Performance Monitor - Counters Figure 5: Performance Monitor - Performance Objects In addition to detailed monitors of a large set of system counters, Performance Monitor also allows the user to set up alerts that will cause an action to occur when a specified counter exceeds a defined threshold [Aubley01]. Alerts can be set up to log an entry, send a message, or run any command or program whenever a counter exceeds a threshold set by the user. It provides functionality similar to that of Task Manager but without any control over processes. It simply outputs process performance measurements to the command prompt windows. Process Monitor is not very configurable and only has one update rate: The advantage of Process Monitor over Task Manager is more in user preference than anything else. Some users who are more comfortable with command line interfaces or who prefer the simplest view into the state of the processes running on a machine may prefer Process Monitor over Task Manager. However, there is not any data provided by Process Monitor that is not also displayed in Task Manager. Like Performance Monitor, Process Explode provides a vast amount of performance data of processes, threads, memory, and the system in general. Most of the measurements are the same, but Process Explode provides access to a few additional measurements of these four system objects [Moore]. Like all of the performance measurement tools discussed thus far, measurements are made by getting counts for various system objects. The main differences of Process Explode compared to Performance Monitor are: Process Explode requires no setup to see all of the performance measurement information all on one window. However, this can be a little overwhelming upon initial use of this tool, since there is so much data displayed in a single window. Process Explode does not update automatically, but instead only updates when the user refreshes it manually. Process Explode does not include any alert or logging capabilities. Process Explode can change process priorities and permissions and can stop running processes like Task Manager. The main differences of Process Explode compared to Task Manager are: Process Explode shows many more performance measurements than Task Manager. Process Explode cannot be used to get a quick summary of the CPU and memory usage of all running processes Back to Table of Contents 2. Process Viewer is very similar to Process Explode, but it shows a subset of the process performance measurements shown by Process Explode [Moore]. Like Process Explode, Process Viewer does not update its information automatically but instead only updates manually when a user presses the Refresh button. Process Viewer is most useful for examining process memory usage. It has a separate Memory Details window that shows all of the user address space counts for a process along with the virtual memory counts. The ps command provides detailed information and performance statistics about running processes such as the names and process IDs as well as the current CPU usage [Ezolt05]. The ps command is one of the oldest performance measurement tools for Unix systems that is still used widely today, and it has a large number of options and parameters that it can display. Some of the parameters for the ps command that are most useful for process performance monitoring include: Top has been one of the most used tool for monitoring real time performance of processes in Unix systems ever since its introduction 22 years ago in BSD Unix 4. Top produces a list of all the currently running processes listed in order of CPU usage. The top CPU users appear at the top of the list, leading to the name of this command. The list is continuously updated at five second intervals by default, and there are options to shorten or lengthen the update period [Fink02]. Top also provides some general system information and performance measures including: In addition to displaying all of this information, Top also allows manipulation of processes in a limited interactive mode. In interactive mode processes can be terminated, their priorities can be changed, and the display can be configured for filtering and desired formatting. It is very simplistic compared to the Windows tools described thus far because it does not monitor individual process, but instead only monitors the system as a whole. The display for xosview shows histograms of a number of useful system parameters including CPU usage split up by process type and the load average of processes [Fink02]. The complete list of system parameters that can be monitored includes load average, CPU usage, memory usage, swap space usage, page swapping, disk usage, and interrupt rate. Each histogram plot is color coded to represent different regions. For example, CPU usage is split into four regions: Each of these shows up as a different colored bar representing what percentage of the total belongs to that region. Xosview is highly configurable, with options to turn on or off every meter and change other detailed settings like separately controlling the sample rate for each parameter. It is one of the easiest ways to quickly visualize the current system state in a graphical format. The treeps application dynamically updates it measurements at an adaptive rate based on process activity, sampling data for inactive processes less frequently than active ones. Treeps displays all the fields from the procinfo structure including CPU time, process status, current memory used, and a number of static properties like the process name and Id. In addition to showing process status, treeps also displays process hierarchy, so it is easy to see process parents and children. All of the detailed information can be displayed by selecting a process from the hierarchy tree [MacDonald]. Treeps does allow limited process control similar to Task Manager for Windows. From treeps, processes can be killed or signaled. Treeps is for real time monitoring only, and it does not include any logging capabilities. The five features listed in the table indicate whether or not each tool exhibits five common features of real time monitoring tools.

Chapter 3 : Zen and the Art of System Monitoring | Scalyr

Monitor your IBM Spectrum Protectâ„¢ solution so that you know when you must investigate performance changes. Operating systems have different tools that are available for monitoring performance. Simulating workloads to test performance is another useful task to learn. To monitor system processors.

Use your operating system diagnostic tools and utilities to monitor system activity. Before adjusting IBM Content Search Services parameters that are referred to below, review the documentation for each parameter mentioned below and follow the recommendations for optimal settings. Use the information in the following table to troubleshoot system performance. Documents are not submitted fast enough for indexing. To validate, monitor queues and check whether the input queue size is low, relative to the configured input queue size. If so, ensure that the client is pushing documents fast enough to the server. Monitor memory consumption on the server computer. Consider reserving memory or increasing the maximum heap size. Monitor disk activity and see the next section "Too much disk activity during indexing" for recommendations. For example, the number of preprocessing threads, tokenizers, or indexing threads might be too low. See the documentation for each parameter for recommendations on optimal settings. The queue size is too small. Monitor queues and ensure that indexing progresses smoothly. Also ensure that the queue size is appropriate for your heap memory allocation and the size of documents that you are processing. Note that increasing the queue size too high can have a negative impact on performance. Some IBM Content Search Services files such as program files, configuration files, logs, or collection data might be stored on a remote disk, a disk with high latency, or disk with limited throughput. For the temporary folder, consider using a RAM drive, increasing the disk block size, or using a faster disk. To determine whether this is a problem, shut down other applications. There is insufficient available memory. Multiple collections are being indexed simultaneously. If this is the cause, consider storing the data for different collections on different disks. Index merge operations occur too frequently. Consider increasing the values of the following indexing parameters: When IBM Content Search Services memory consumption is high even without out-of-memory errors , performance can degrade due to frequent JVM garbage collection calls and more expensive memory management. Consider increasing the heap memory by increasing the value of the maxHeapSize parameter. Queue size setting might be too high. Check the queue size settings. If necessary, reduce the queue sizes. Too many preprocessing and indexing threads are defined. Check how many preprocessing and indexing threads are defined. If necessary, reduce the number of threads. Too many collections are indexed concurrently. If possible, limit the number of concurrently indexed collections. Overall memory consumption on the IBM Content Search Services computer is very high, or higher than the physical memory. Operating system memory management using virtual memory, garbage collection, and so on starts to have a noticeable effect on the overall system performance, even when out-of-memory errors do not occur. Too many applications are running. Consider shutting down applications or distributing applications to different computers. Physical memory is too small. Increase the physical memory or reserve more physical memory if applicable. CPU consumption is very high and search times are very slow when indexing and searching on the same computer.

Chapter 4 : Monitoring IBM Spectrum Protect performance with operating system tools

Operating systems like Windows, Linux, AIX & z/Linux are the primary interface for both the hardware and the applications of your datacenter. Consolidating performance statistics, system availability, and actionable insights from multiple platforms is the fastest way to analyze your operating environment and understand root cause.

Familiarize yourself with platform-specific issues so that you know what performance options the operating system provides. The goal should be to run most of the time in application mode, also called user mode, rather than system mode. The ratio of time spent in each mode is only a symptom of the underlying problem, which might involve the following: Paging or swapping Executing too many operating system calls Running too many processes If such conditions exist, then there is less time available for the application to run. The more time you can release from the operating system side, the more transactions an application can perform. You can easily monitor many factors with the Windows administrative performance tool: On a busy system, free memory likely contains a page belonging to one or more currently active process. When that access occurs, a soft page fault takes place, and the page is included in the working set for the process. If the process cannot expand its working set, then one of the pages currently mapped by the process must be moved to the free set. Any number of processes might have pages of shared memory within their working sets. The sum of the sizes of the working sets can thus markedly exceed the available memory. Then, determine whether sufficient CPU resources are available and recognize when your system is consuming too many resources. Begin by determining the amount of CPU resources the Oracle instance utilizes with your system in the following three cases: However, if your system shows high utilization at normal workload, then there is no room for a peak workload. For example, Figure illustrates workload over time for an application having peak periods at Each user entering one transaction every 5 minutes translates into 9, transactions daily. Over an 8-hour period, the system must support 1, transactions an hour, which is an average of 20 transactions a minute. If the demand rate were constant, then you could build a system to meet this average workload. However, usage patterns are not constant and in this context, 20 transactions a minute can be understood as merely a minimum requirement. If the peak rate you need to achieve is transactions a minute, then you must configure a system that can support this peak workload. As users are added to an application, the workload can rise to what had previously been peak levels. No further CPU capacity is then available for the new peak rate, which is actually higher than the previous. CPU capacity issues can be addressed with the following: Tuning, or the process of detecting and solving CPU problems from excessive consumption. Increasing hardware capacity, including changing the system architecture See Also: Oracle Database Resource Manager does this by allocating and managing CPU resources among database users and applications in the following ways: Limit number of active sessions for each Consumer Group This feature is particularly important when a Consumer Group has a lot of parallel queries and you want to limit the total number of parallel queries. This feature can lower the CPU consumption of low-priority sessions. Runaway queries Oracle Database Resource Manager can limit the damage from runaway queries by limiting the maximum execution time for a call, or by moving the long-running query to a lower priority Consumer Group. Therefore, tuning non-Oracle factors can also improve Oracle performance.

Chapter 5 : Monitoring Virtual Machine Performance

Monitoring the performance of operating systems and processes is essential to debug processes and systems, effectively manage system resources, making system decisions, and evaluating and examining systems. These tools are primarily divided into two main categories: real time and log-based. Real.

The Performance console is available only on Windows hosts. You cannot monitor performance for virtual machines on Linux hosts. However, you can monitor the performance of any virtual machines running on the Windows host, including those running Linux guest operating systems. The VMware Workstation performance counters can monitor the following data from a running virtual machine: Reading and writing to virtual disks Memory used by the virtual machine Virtual network traffic You can track virtual machine performance only when a virtual machine is running. The performance counters reflect the state of the virtual machine, not the guest operating system. For example, the counters can record how often a virtual machine reads from a virtual disk, but they cannot track how many processes are running inside the guest operating system. An explanation of each counter appears in the Performance console. To add counters to track virtual machine performance, use the Windows Performance console. Take the following steps. Open the Administrative Tools control panel and double-click Performance. The Performance console opens. The Add Counters dialog box appears. In the Performance object list, select VMware. Decide whether you want to add all counters or select specific counters from the list. To use these counters for all running virtual machines, select All instances. To use the counters for specific virtual machines, select Select instances from list, then choose the virtual machines you want. The names shown in the list correspond to the display names of running virtual machines. For a brief description of each counter, click Explain. When you select a counter from the list, a description appears below the Add Counters dialog box. Click Add to add the counters to the Performance console.

Chapter 6 : SAP on IBM i: Monitoring Operating System Performance with CA APM IntroscopeÂ® | SAP B

As part of the overall performance monitoring, CA APM Introscope is offering operating system performance monitoring, such as CPU utilization or disk I/O performance data. The data is provided on the monitored system through the Diagnostics Agent, which receives its data from the SAP Host Agent.

There can be many processes, and each process has its own 2 GB of private virtual address space. This frees that RAM frame for other uses. There can be one such file in each disk partition. Users frequently ask "how big should I make the pagefile? If there is no other information available, the typical recommendation of 1. On server systems, you typically want to have sufficient RAM so that there is never a shortage and so that the pagefile is basically not used. On these systems, it may serve no useful purpose to maintain a really large pagefile. On the other hand, if disk space is plentiful, maintaining a large pagefile for example, 1. Performance, architectural limits, and RAM On any computer system, as the load increases the number of users, the volume of work , performance decreases, but in a nonlinear manner. Any increase in load or demand, beyond a certain point, causes a significant decrease in performance. This means that some resource is in critically short supply and has become a bottleneck. At some point, the resource that is in short supply cannot be increased. This means that an architectural limit has been reached. Some frequently reported architectural limits in Windows include the following: However, Windows Vista, Windows Server , and Windows 7 do not all share these architectural limits. The limits on user and kernel memory numbers 1 and 2 here are the same, but kernel resources such as PTEs and various memory pools are dynamic. This new functionality enables both paged and nonpaged memory. This also enables PTEs and session pool to grow beyond the limits that were discussed earlier, up to the point where the whole kernel is exhausted. Frequently found and quoted statements such as the following: However, you have to monitor your system to know whether they apply to your particular system or not. In some cases, these statements are conclusions from specific Windows NT 4. Significant changes were made to Windows Server to reduce the probability that these architectural limits will in fact be reached in practice. For example, some processes that were in the kernel were moved to non-kernel processes to reduce the memory used in the shared virtual address space. Monitoring RAM and virtual memory usage Performance Monitor is the principle tool for monitoring system performance and for identifying the location of the bottleneck. Here is a summary of some important counters and what they tell you: This counter is a measure of the demand for virtual memory. This shows how many bytes were allocated by processes and to which the operating system has committed a RAM page frame or a page slot in the pagefile or perhaps both. As Committed Bytes grows greater than the available RAM, paging will increase, and the pagefile size that is being used will also increase. At some point, paging activity starts to significantly affect performance. This counter is a measure of the virtual memory in "active" use. This counter is a measure of how much of the pagefile is actually being used. Use this counter to determine whether the pagefile is an appropriate size. If this counter reaches , the pagefile is full, and things will stop working. Depending on the volatility of your workload, you probably want the pagefile large enough so that it is generally no more than percent used. This counter is one of the most misunderstood measures. A high value for this counter does not necessarily imply that your performance bottleneck stems from a shortage of RAM. The operating system uses the paging system for purposes other than swapping pages because of memory over-commitment. This counter shows how many virtual memory pages were written to the pagefile to free RAM page frames for other purposes each second. This is the best counter to monitor if you suspect that paging is your performance bottleneck.

Chapter 7 : Operating System and Process Monitoring Tools

Take the complexity out of monitoring operating systems and simplify the process with a unified solution that allows you to visualize core physical and virtual metrics, and gain deep insight into memory, CPU and I/O processes.

April 27, Last Updated: After being a Linux Administrator for 5 years in IT industry, I came to know that how hard is to monitor and keep systems up and running. These commands are available under all flavors of Linux and can be useful to monitor and find the actual causes of performance problem. This list of commands shown here are very enough for you to pick the one that is suitable for your monitoring scenario. Linux Command Line Monitoring 1. The top command used to dipslay all the running and active real-time processes in ordered list and updates it regularly. It also shows high memory and cpu utilization of a running processess. The top command is much userful for system administrator to monitor and take correct action when required. By default vmstat command is not available under Linux systems you need to install a package called sysstat that includes a vmstat program. The common usage of command format is. The open files included are disk files, network sockets, pipes, devices and processes. One of the main reason for using this command is when a disk cannot be unmounted and displays the error that files are being used or opened. With this commmand you can easily identify which files are in use. The most common format for this command is. It also provides a option to save captured packages in a file for later analysis. Netstat â€" Network Statistics Netstat is a command line tool for monitoring incoming and outgoing network packets statistics as well as interface statistics. It is very useful tool for every system administrator to monitor network performance and troubleshoot network related problems. Htop â€" Linux Process Monitoring Htop is a much advanced interactive and real time Linux process monitoring tool. This is much similar to Linux top command but it has some rich features like user friendly interface to manage process, shortcut keys, vertical and horizontal view of the processes and much more. For more information on installation read our article below. Install Iotop in Linux 8. This tool is often used to trace storage device performance issues including devices, local disks, remote disks such as NFS. Psacct or Acct â€" Monitor User Activity psacct or acct tools are very useful for monitoring each users activity on the system. Both daemons runs in the background and keeps a close watch on the overall activity of each user on the system and also what resources are being consumed by them. These tools are very useful for system administrators to track each users activity like what they are doing, what commands they issued, how much resources are used by them, how long they are active on the system etc. For installation and example usage of commands read the article on Monitor User Activity with psacct or acct Monit â€" Linux Process and Services Monitoring Monit is a free open source and web based process supervision utility that automatically monitors and managers system processes, programs, files, directories, permissions, checksums and filesystems. The system status can be viewed from the command line or using it own web interface. Linux Process Monitoring with Monit NetHogs â€" Monitor Per Process Network Bandwidth NetHogs is an open source nice small program similar to Linux top command that keeps a tab on each process network activity on your system. It also keeps a track of real time network traffic bandwidth used by each program or application. It has a built in HTTP web server that regularly collects system and network information and display them in graphs. It Monitors system load average and usage, memory allocation, disk driver health, system services, network ports, mail statistics Sendmail, Postfix, Dovecot, etc , MySQL statistics and many more. It designed to monitor overall system performance and helps in detecting failures, bottlenecks, abnormal activities etc. Monitorix Monitoring Read More: It continuously keeps watch on Ethernet traffic and produces a log of IP and MAC address pair changes along with a timestamps on a network. It also has a feature to send an email alerts to administrator, when a pairing added or changes. It is very useful in detecting ARP spoofing on a network. Arpwatch to Monitor Ethernet Activity VnStat PHP monitors a network traffic usage in nicely graphical mode. It displays a total IN and OUT network traffic usage in hourly, daily, monthly and full summary report. With the Nagios system, administrators can able to monitor remote Linux, Windows, Switches, Routers and Printers on a single window. This tool comes in two modes: Online Mode and Capture Mode. Nmon Monitoring Read More: All-in-One Performance Monitoring Tool Collectl is a yet another powerful and feature rich command line based utility, that can be used to gather information about Linux system resources such as CPU usage, memory, network, inodes, processes, nfs, tcp, sockets and much more. Collectl Monitoring Read More: Install Collectl All-in-One Performance Monitoring Tool in Linux We would like to know what kind of monitoring programs you use to monitor performance of your Linux servers?

Chapter 8 : Monitoring Guest Operating System Performance

SysGauge is a system and performance monitoring utility allowing one to monitor the CPU usage, memory usage, network transfer rate, operating system performance, the status and resource usage of running processes, file system performance, USB performance, disk space usage, disk read activity, disk write activity, disk read transfer rate, disk write transfer rate, disk read IOPS and disk write.

Try It Free This is a living, community-editable document. Minimize production issues; Do all this with as little effort as possible. Challenges Three historical challenges to effective monitoring are a false sense of security, a lack of cohesive tools, and the wrong mindset. These are related to one another. The false sense of security looks like this: A ping test will let me know if my site goes down, so all I need is a ping test! Similarly, most hosting providers will give you some default graphs and alerts on your dashboard: Basic traffic and CPU metrics, an email alert if the server fails, and maybe some raw log access. If your web server instance runs out of memory, starts swapping to disk, and site performance drops to nil, you might not notice for a long time. What happens when a problem hits and all three systems start peppering you with alerts? This leads to the third major challenge - and perhaps the most important: Our Tenets We address these challenges with a cohesive approach to monitoring. This approach will help ensure that all your bases are covered, and put you in a mindset of monitoring confidence and security. Identify as many problems as possible. Identify problems as early as possible. Longer lead times are your friend. Generate as few false alarms as possible. False alarms aka false positives can lead to "alert fatigue" - where you get so used to seeing a noisy alert that it starts to carry less psychological weight. This alert fatigue can, paradoxically, lead to increased downtime. Do not, in other words, allow your monitoring system to cry wolf. Do it all with as little work as possible. We are engineers, after all. The Tools Our core components are alerts, graphs, and logs. These work together to help you identify and troubleshoot issues easily and quickly. Alerts notify you when things change beyond specified thresholds. Setting the proper threshold is key to successful alerting we address this in depth in our companion guide, appropriately named How To Set Alerts. Used properly, alerts can warn you well in advance of a disaster and enable you to fix problems on a convenient schedule and without affecting users. Graphs are the visual representations of your data. Graphs help you see trends that might not be obvious in raw form. We humans have remarkably good visual-pattern-recognition systems, and graphs enable us to identify anomalous data quickly. They also look super cool on a huge flatscreen TV dashboard mounted in the middle of your office. Logs hold your raw operational data, recording events in the operating system or application. Whereas graphs summarize an overall trend potentially hiding important details , logs will always have the finest level of detail and be the "base source of truth. In a production environment, there will likely be multiple systems we want to monitor e. You can consolidate as you move down the stack. The Application The first layer of metrics we care about are those generated from, and specific to, the application itself. Those come lower down in the stack. The Process While the specifics of the process model are slightly different between UNIX and Windows, the concept is the same - a process is the running instance of an application within the operating system. An application may have multiple processes running at any given time, and there can be dozens to hundreds of processes running on a server. The Server One way of looking at a server is simply as a container for your processes. At this layer, we care about the health and resource usage of the overall system. Server metrics include system-wide resource usage data CPU, memory, disk and network usage , summary metrics total of processes, load average, socket state and availability and hardware state and health disk health, memory health, physical-port access and use, CPU temperature, and fan speed. The Hosting Provider Your servers live within your hosting provider. Also included in this layer are the associated hosted services that support your application. These may include databases, load balancers, and other "cloud" services your application implicitly or explicitly depends on. As more and more critical services move away from local servers and into cloud-based services, monitoring the status of those services becomes increasingly important, particularly when the health of those services is out of your immediate control. When lower-level services fail, many higher-level alerts will trigger and loudly. The extra alert telling you that your hosting provider is down will save you steps in the analysis stage. External Dependencies Very often overlooked but absolutely critical. Several are common to nearly all production web services, and failures here can lead to unexpected and painfully long downtime. Domain Names - DNS renewal dates can creep up and cause severe headaches if forgotten just ask Microsoft. SSL certificates - these also expire, with consequences almost as severe as DNS, and certificate providers are not as proactive with expiration warnings as DNS providers are. The User Finally, we get to the last but certainly not least layer in our stack - the user. In practice, this can be as simple as an external HTTP monitor that looks for a OK response from an API endpoint, or a series of automated browser tests that check a sequence of pages for specific responses. This can also include wider-ranged metrics that happen to roll up a number of lower-level behaviors and implicitly test them. Monitoring Strategies For each layer of the stack, you can apply four core monitoring strategies: Monitor Potential Bad Things Most problems are detectable in advance if you look carefully enough. Since one of our stated goals is to do this all with "as little work as possible" - and firefighting is a lot more work than casual maintenance - averting problems is a good place to start. CPU overload, network saturation, queue overflows, disk exhaustion, worker processes falling behind, or API rate limits being exceeded. These failures rarely come out of nowhere. The effect may appear very suddenly, but the cause creeps up gradually. This gives you time to spot the problem before it reaches a tipping point. Any form of storage can fill up. Monitor unused memory on each server, free space in each garbage-collected heap, available space on each disk volume, and slack in each fixed-size buffer, queue, or storage pool. Any throughput-based resource can become oversubscribed. A background thread or fixed-size thread pool can choke even if the server has CPU to spare. Lock contention can also cause a tipping point on an otherwise lightly loaded server. Internally at Scalyr, we generate a log message each time we acquire or release certain critical locks, allowing our log analyzer to compute the aggregate time the lock is held. Monitor Actual Bad Things Despite our best efforts, things will break. When they do, your best defense is to be prepared and respond quickly. To respond quickly, you need to know about the problem as soon as it occurs. Action Items for Monitoring Actual Bad Things Identify the resources in each layer of your stack that can stop functioning correctly; Understand the effects that each potential failure can have on the rest of the system; Make sure alerts are set to trigger when those resources fail, and that those triggers reach you with an appropriate level of urgency; and Have tools in place to move quickly from alert to graph to log as necessary in order to identify the root cause of a failure. In a famous incident in , for over an hour, Oracle. Exception Monitoring Exception monitoring is worth special mention here. Exceptions come in several flavors -- most commonly as application exceptions that wind up as a stack trace in your application log. Application exceptions provide some of the best monitoring bang for your buck, since they generally correlate with an application bug, and can surface edge-case issues that are hard to spot in the high level statistics used by the rest of your monitoring. A bonus is that, by closely watching exceptions, you get to impress users by proactively reaching out to them before they even complain about an error. Even in the best-intentioned setups, there will be gaps in the "potential bad things" and "actual bad things" strategies outlined above. The specifics will depend on your application, but you might look for a dropoff in page loads or invocations of important actions. For instance, if you were in charge of operations for Twitter, you might start by monitoring the rate of new tweets, replies, clickthroughs on search results, accounts created, and successful logins. Action Items for Monitoring Good Things Identify the key user actions and system behaviors that indicate a "green" system status; and Set alerts to notify you when these values fall out of line. This avoids false positives or negatives due to normal variations in usage. After each incident, ask yourself how you could have seen it coming. Sometimes, the hints are already there in your log and you just need to add an alert. Other times, you might need to make a code change to log additional information. Start with tight thresholds and broad, aggressive alerts and iteratively narrow them to eliminate false alarms. Pay particular attention to noisy alerts false positives and work on quieting them. Noisy alerts play on a feature or bug, depending on your perspective in the human brain that allows us to grow tolerant of repetition. Further Reading If you found this guide useful, be sure to look at some of the other material on best practices in the Scalyr Community. Summary Properly monitoring your systems and setting up thorough alert coverage requires effort. How do you think about the monitoring of your own production systems? Let us know in the comments! About Scalyr

Scalyr offers a fast, powerful server monitoring, alerting, and log management service. So we decided to fix that.

Chapter 9 : Monitoring operating system activity

Linux Top command is a performance monitoring program which is used frequently by many system administrators to monitor Linux performance and it is available under many Linux/Unix like operating systems. The top command used to dipslay all the running and active real-time processes in ordered list and updates it regularly.