Postprint of Applied Research on Zabbix-based Monitoring of Intranet Editing Systems

Authors: Shao Jianhui

Date: 2023-10-08T00:00:00+00:00

Abstract

Servers represent the core component of contemporary computer network architectures, with monitoring systems playing a vital role in the operation and maintenance management of server clusters. This study investigates the open-source Zabbix monitoring system and constructs, based on Zabbix, a monitoring solution for an internal network editing system. The system enables surveillance of critical application processes, manuscript directories, and inbound/outbound statuses on servers. Through user customization of Zabbix, including the addition of specific monitoring items and other tailored functions, a distributed and efficient monitoring framework is established, achieving automated discovery-to-alarm workflows for various monitoring targets such as servers, thereby enhancing the automation level of monitoring operations and maintenance activities.

Full Text

Research on the Application of Zabbix-Based Monitoring for Internal Network Editing Systems

Abstract: Servers constitute the core of modern computer network architectures, and monitoring systems play a vital role in the operation and maintenance management of entire server clusters. This paper investigates the open-source Zabbix monitoring system and builds an internal network editing system monitoring solution based on Zabbix. The system enables monitoring of critical application processes, manuscript directories, and import/export status on servers. Through user customization of Zabbix, including the addition of specific monitoring items and functionalities, we implement a distributed, efficient monitoring system that automates the entire process from discovery to fault alerting for various monitored objects such as servers, thereby improving the automation level of monitoring and maintenance operations.

Keywords: Zabbix; server monitoring; monitoring system; distributed monitoring
Classification Code: TP393.3
Document Code: A
Article ID: 1671-0134(2018)07-058-03
DOI: 10.19483/j.cnki.11-4653/n.2018.07.015

1. Introduction to Zabbix

Zabbix is a distributed, open-source enterprise-level automated operations and maintenance solution based on a web interface. It can monitor various network devices, storage systems, and server application performance parameters, providing flexible alerting mechanisms that enable system administrators to quickly locate faulty equipment. Zabbix features active monitoring capabilities, multi-dimensional alerting functions (including WeChat, phone, and email), support for heterogeneous platforms across multiple systems, monitoring of any IP-protocol-enabled device, open-source software that allows for customized development as needed, and support for script execution to achieve automated operations. Consequently, it has gained widespread recognition and adoption among major internet companies.

1.1 Zabbix Architecture and Operation

Depending on network environment and monitoring scale, Zabbix can be deployed in three primary architectures: server-client, master-node-client, and server-proxy-client. This paper adopts the server-client architecture, which is the simplest Zabbix configuration where monitoring servers and monitored machines interact directly without any intermediary agents. This architecture is suitable for relatively simple network environments with fewer devices. In this setup, the Zabbix server is installed on the monitoring server, collects data in a C/S mode, stores historical data in a database (such as MySQL or Oracle), and displays data and system configuration through a web interface. It provides monitoring and data collection for remote servers and network status via SNMP, Zabbix agent, ping, port monitoring, and other methods, and can run on Linux, Solaris, HP-UX, AIX, Free BSD, Open BSD, OS X, and other platforms.

The Zabbix agent is installed on target monitored servers and is primarily responsible for collecting information about hardware, system performance, file systems, applications, and networks. Data collection by the Zabbix agent operates in two modes: active and passive.

1.1.1 Active Mode In active mode, the agent requests the server for a list of active monitoring items and proactively submits the required detection data to the server.

1.1.2 Passive Mode In passive mode, the server requests data for monitoring items from the agent, and the agent returns the requested data.

1.2 Zabbix Monitoring and Alerting Process

A complete Zabbix monitoring and alerting process consists of the following components: Item (monitoring item), Trigger (alerting trigger), Action (response action), Media (alerting medium), User (user account), and Event (event). An Item represents a monitored metric, which can be either a built-in Zabbix item or a custom user-defined item added through UserParameter configuration in .conf files. A Trigger is an alerting mechanism that evaluates collected item values against defined conditions to generate events with specific severity levels. An Action defines the response measures that the server must take when an Event occurs, with the response method determined by the Media type, which includes five options, the most common being mail, SMS, and script.

The complete monitoring and alerting workflow is as follows: The server periodically collects Item data through agents. Triggers evaluate this data against their defined conditions; if conditions are met, events are generated with corresponding severity levels (classified as Disaster, High, Average, Warning, or Information) and notifications are sent to designated users through defined media, awaiting user acknowledgment and handling.

---

2. Application in Internal Network Editing System Monitoring

2.1 Zabbix Installation and Deployment

Based on the scale of servers in the internal network editing system, the monitoring system adopts the server-client architecture, with direct data interaction between Zabbix server and Zabbix agentd, using passive mode for data collection where the server requests monitoring data from agents.

Server-side installation requires: (1) Installing the PHP runtime environment; (2) Installing database support such as MySQL; (3) Creating a Zabbix system account on the server; (4) Downloading Zabbix source code and compiling and installing the server-side components on Linux. For deployment in an internal network environment, Zabbix server version 3.0TLS (with the latest version being 3.0.19) was selected for source compilation installation on CentOS 7.5.

Client installation involves selecting and installing the appropriate Zabbix agent client based on the operating system of the monitored server. Windows servers use a direct 64-bit executable installation, while Linux servers employ source compilation installation.

2.2 Creating Monitoring Items

Based on business requirements analysis of the internal network editing system, monitoring items are categorized into several main types: file systems, system performance, application processes, manuscript directories, network status, and web monitoring. The specific monitoring items are detailed in Table 1 .

Table 1. Monitoring Items for Internal Network Editing System

Monitoring Item	Zabbix Key and Parameters	Return Data Type
System disk or partition usage	vfs.fs.size[disk,pused]	Percentage used
CPU usage	system.cpu.load[,avg1]	Average percentage
Memory availability	vm.memory.size[available]	Available bytes
Application processes	proc.num[„,process_{name}]	Process count
Manuscript directory empty check	user.filecount[directory_{path}]	File count
Manuscript directory timeout check	vfs.file.time[directory_{path}]	Unix timestamp
Network connectivity	icmpping	Status
Web service status code	web monitoring	Status code

File system monitoring primarily tracks disk space usage on servers, with the monitoring key set as `vfs.fs.size[disk,pused]`, returning the percentage of disk usage.

System performance monitoring focuses on CPU usage and memory availability, with keys set as `system.cpu.load[,avg1]` and `vm.memory.size[available]`, returning average CPU usage percentage and available memory bytes, respectively.

Application process monitoring tracks the running status of various application processes on servers, with the key set as `proc.num[,,process_{name}]`, returning the number of running instances of the specified application process.

Manuscript directory monitoring tracks the processing status of incoming and outgoing manuscripts through two checks: directory empty verification and directory timeout verification. The timeout check uses the key `vfs.file.time[directory_{path}]`, returning a Unix timestamp in seconds. The empty check cannot be implemented with built-in Zabbix keys and requires customization. The customization method involves modifying the `zabbix_{agentd}.conf` configuration file on the agent server by setting

UnsafeUserParameters=1 and adding UserParameter=user.filecount[*],ls -l "$1" |grep "^-"|wc -l. On the server side, the custom key is defined as user.filecount[directory_{path}], returning the number of files in the monitored directory.

2.3 Creating Triggers

The general trigger format is: {<server>:<key>.<function>(<parameter>)}<operator><constant>. Trigger settings for the internal network editing system are configured as follows:

1. **File system monitoring trigger**: {server_{name}:vfs.fs.size[disk,pused].last(0)}>80, which generates a High severity alert when disk usage exceeds 80%.
2. **System performance monitoring triggers**: {server_{name}:system.cpu.load[/,avg1].last(0)}> generates a High severity alert when average CPU usage exceeds 80%; {server_{name}:vm.memory.size[available].last(0)}<1 generates a High severity alert when available memory falls below 1GB.
3. **Application process monitoring trigger**: {server_{name}:proc.num[,,,process_{name}].last(0 generates a High severity alert when the process count is not equal to 1.
4. **Manuscript directory monitoring triggers**: {server_{name}:user.filecount[directory_{path}] generates an Average severity alert when the directory remains empty for 10 minutes; {server_{name}:vfs.file.time[directory_{path}].fuzzytime(1h)}=0 generates a High severity alert when the directory has not been updated for 1 hour.
5. **Network status and web monitoring** use system default values, triggering alerts when servers become unreachable via PING or when web pages return error status codes.

2.4 Setting Up Graphical Monitoring Interface

Based on the configured monitoring items, graphical monitoring interfaces can be created and aggregated to facilitate intuitive viewing of historical monitoring status and future trends. Combined with properly configured triggers, this enables early problem detection. An example of the graphical monitoring interface is shown in Figure 1 [Figure 1: see original paper].

Conclusion

This paper demonstrates the deployment of a distributed, efficient, and visualized monitoring system for an internal network editing environment using the open-source Zabbix tool. Through user customization and the addition of specific monitoring items, the system enables monitoring of critical application processes, manuscript directories, and import/export status on servers. The

web-based interface provides comprehensive visibility into current and historical server performance across the entire system, offering system administrators real-time diagnostic capabilities and the ability to predict potential issues proactively. This implementation significantly reduces the burden of server management tasks and plays a crucial role in ensuring the continuous operation of internal network editing systems.

References

[1] Distributed System Monitoring Zabbix. Open Source Community Network. https://www.oschina.net/p/zabbix.
[2] Huang Jian. Application and Research of ZABBIX in Server Monitoring[J]. Science and Technology Information, 2010(20).
[3] Yang Hao, Lü Haiming. Application of Zabbix System in Networked Video Surveillance Platforms[J]. Information & Communications, 2014(10): 86-87.
[4] Wang Feng. Design and Implementation of an Internal Network Monitoring System Based on Zabbix[D]. Shanghai Jiao Tong University, 2012.

Author Affiliation: Technical Bureau, Xinhua News Agency

Note: Figure translations are in progress. See original paper for figures.