Postprint: High-Availability and Disaster Recovery Design for CMS Based on Open-Source Relational Databases

Authors: Zhao Qiang, Wang Wenzheng

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

Abstract

This paper describes a method for implementing automated disaster recovery backup for high-load news publishing systems using a complementary approach based on MySQL full backups and binlog logs, meeting the business requirement of activating the backup system in the disaster recovery data center for subsequent publishing within the shortest possible time in the event of a total data center failure.

Full Text

High-Availability Disaster Recovery Design for CMS Based on Open-Source Relational Databases

Abstract: This paper describes an automated disaster recovery and backup method for high-load news publishing systems, implemented through a complementary approach combining MySQL full backups and binlog replication. The solution meets business requirements for enabling backup systems in disaster recovery data centers within the shortest possible time when a global failure occurs in the primary facility.

Keywords: Content Management System; Disaster Recovery; MySQL; binlog; ElasticSearch
CLC Number: G210.7
Document Code: A
Article ID: 1671-0134(2018)01-079-03
DOI: 10.19483/j.cnki.11-4653/n.2018.01.029

Authors: Zhao Qiang, Wang Wenzheng

1. CMS System Overview

1.1 Application Architecture

The CMS is a Browser/Server (B/S) structured web application system that uses Apache as the web server and MySQL open-source database as the core content storage. Editors upload various content materials—including text, images, and videos—through the publishing system. After layout editing, column configuration, and multi-level review, the system organizes these materials into formatted news pages. These pages are then generated as static files by distribution servers and deployed to multiple browsing servers for user access.

1.2 Core Technology Components

1.2.1 MyISAM Storage Engine MySQL has become the most widely used open-source relational database on internet platforms due to its increasing functional maturity and reliability. Among MySQL' s storage engines, MyISAM offers several advantages: its file structure is relatively independent, data files can be compressed to save storage space, it provides good cross-platform portability, and individual tables can be backed up and restored separately—facilitating file-based backup strategies. Additionally, MyISAM delivers fast query performance. While it has limitations, including lack of transaction support and table-level locking during writes (rather than row-level locking), the CMS system has implemented extensive optimization through table partitioning and query/write optimization to address these constraints. Given the system' s requirement for high query speed and these optimization measures, MyISAM was selected as the storage engine to enhance overall operational efficiency and maintainability.

1.2.2 rsync rsync (remote sync) is an efficient remote data synchronization tool that employs the "rsync algorithm" to synchronize local and remote files by transmitting only the differing portions rather than entire files, resulting in high transfer efficiency. It also supports SSH-based file transfer, ensuring good security. The China News Service CMS currently supports approximately 280 concurrent users daily, publishing tens of thousands of articles. The system generates over 100,000 new database records per day, with peak database query loads reaching 500 QPS. Operating on a 7$\times$24 continuous publishing schedule, the system must maintain disaster recovery capabilities to switch to backup nodes when the primary fails, while carefully managing backup bandwidth consumption to avoid impacting normal business operations during peak hours.

1.2.3 ElasticSearch ElasticSearch (ES) is a search server based on Lucene that functions not only as a full-text search engine but also as an excellent distributed real-time document storage system. ES cluster deployments can scale to hundreds of servers, processing petabytes of structured or unstructured data.

2. Limitations of Traditional Backup Solutions

Due to business data characteristics and structural requirements, the China News Service CMS cannot simply use MySQL' s native master-slave replication for database synchronization. Instead, database synchronization relies on transferring database files directly. However, this approach presents significant challenges: in large database tables, only a small portion of records may be inserted or modified, but from the database file perspective, the entire table file has changed. File-level synchronization therefore requires analyzing and synchronizing these large database files in their entirety. Although rsync technology can transfer only file differences, its analysis of MySQL' s proprietary binary file structure is not precise—minor table modifications often result in substantial data transfers. This synchronization process not only consumes significant network bandwidth resources but also requires considerable transmission time. Furthermore, since MySQL instances do not flush real-time content to disk files, substantial discrepancies remain between the backup system and the primary database. This backup method leads to prolonged business interruption, which is unacceptable for a high-density, 24-hour continuous publishing environment.

3. CMS High-Availability System Design and Implementation

3.1 High-Availability Metrics

Information system high-availability capability is typically measured by two metrics: RTO (Recovery Time Objective) and RPO (Recovery Point Objective). RTO refers to the time period between service interruption and system restoration to operational status following a failure. RPO refers to a past point in time to which data can be recovered when a disaster or emergency occurs.

3.2 Data Disaster Recovery Through Redundancy and Backup Technologies

Redundancy technology is a highly effective method for improving system availability using parallel system models. The CMS high-availability architecture is designed based on this redundancy concept, aiming to achieve two levels of redundancy: first, host system redundancy, and second, data center redundancy. The critical challenge in implementing redundant architecture in web systems lies in data synchronization, as synchronization capability directly determines the RPO. An effective synchronization strategy should minimize RPO—ideally approaching or equaling the failure occurrence time to achieve minimal or zero data loss—while avoiding excessive network transmission that consumes server and bandwidth resources and impacts production operations. CMS business

data includes application programs, business data files (HTML pages, images, video files), and databases.

3.2.1 Redundant Deployment Architecture As shown in [Figure 1: see original paper], the CMS system is deployed across two data centers, achieving data center-level redundancy. At the network level, redundant connections are implemented from hosts to switches and from switches to upstream egress devices. The primary data center deploys two CMS systems to achieve host-level redundancy. These two web systems run on separate physical hosts, each connecting to its own database. Both systems can operate independently without any business dependencies or associations, ensuring rapid failover to the backup system when one system fails.

3.2.2 Binlog Backup Synchronization Strategy Data backup and synchronization are implemented through the operating system' s task scheduler combined with rsync commands. The detailed synchronization strategy is as follows:

Application Synchronization: Since application development iterations occur without fixed schedules and require immediate activation of new versions, we employ Linux' s signal mechanism instead of the inefficient crontab timing mechanism, which causes high-frequency resource consumption. When new versions are uploaded to designated directories, the system automatically detects file changes and immediately initiates synchronization scripts to replicate program files to backup hosts within the same network segment and across different network segments, achieving fast, efficient synchronization with minimal system resource consumption.

Full Database Backup: Full database backups are essential for system reconstruction but significantly impact production systems. Due to the enormous data volume, the backup process can cause system "pseudo-paralysis" for over 30 minutes, making it unsuitable for high-load operations. Therefore, we strictly control backup frequency, performing full backup operations only once daily at 2:00 AM.

Binlog File Synchronization: Binlog records all DDL and DML statements, capturing both data structure and content changes. Backing up binlog ensures eventual consistency of database content and, more importantly, records the data change process. In cases of data loss or misoperation, data can be fully restored using cold backups combined with binlog. Unlike full backups, binlog collection and transmission have minimal impact on system performance and business operations. Based on business requirements, we set the binlog incremental file synchronization cycle to 5 minutes.

3.2.3 Emergency Response Procedures A well-designed emergency response plan is critical to backup recovery effectiveness. Without a reasonable

emergency plan, the overall disaster recovery solution will fail to meet business continuity requirements. The CMS system has developed corresponding emergency response plans for different failure scenarios:

1. When the primary system node crashes or becomes unavailable, Zabbix monitoring and application monitoring can promptly detect the failure and notify operations personnel via SMS, WeChat, or email.
2. Operations personnel assess failure severity. If the fault can be repaired locally within a short time (under 15 minutes), they proceed with repairs. If determined to be a severe system or hardware failure that cannot be quickly repaired locally, the emergency switching process is initiated.
3. Following the emergency switching process, the system is switched to the backup system.
4. Business validity is verified in the backup system.
5. The backup system goes live to provide services.
6. After business system restoration, the primary node system is repaired and validated.

3.2.4 System Switching Methods System switching is the most critical link in the overall emergency response process. All backup strategies are designed to enable effective system switching during failures.

Web System Switching: When a failure occurs on the CMS system host but other devices in the same network segment remain healthy, the backup system within the same segment is activated. As previously mentioned, the backup system's applications and data files are synchronized in real time, making the backup web service immediately available. The switching method involves swapping IP addresses between primary and backup servers. This approach modifies only the IP layer (Layer 3) of the OSI model, leaving upper-layer applications unchanged. DNS resolution and calls from other associated systems remain unaffected, allowing the backup web system to go live without requiring any adjustments to other systems.

For data center-level failures (such as egress network equipment failure, carrier line failure, or power interruption), the web system must switch to the backup system in the remote data center. This requires modifying the system' s domain name resolution and activating the backup system. As this is an internal system using an internal DNS server, DNS propagation time can be minimized, with the current TTL setting at 600 seconds.

Data Switching and Verification: As mentioned in the backup strategy, backup database files are synchronized periodically using cold backup with a 1-day cycle for both same-site and remote backup data centers. Relying solely on cold backup data would result in excessive data lag from an RPO perspective—theoretically up to 24 hours for same-site backups—which is unacceptable for the CMS system. Therefore, additional differential data recovery is required. Recovery and verification are performed from both database and business perspec-

tives. First, the database recovery program performs fine-grained restoration to ensure database entry completeness. Then, the data verification program automatically checks recovery effectiveness and completeness based on specific business data (articles, special topics) stored in ElasticSearch.

3.3 Data Recovery Process

Data recovery consists of two steps: full database restoration and incremental addition. Full restoration occurs daily at 2:00 AM after the database backup, when a script synchronizes operational files to backup hosts and performs database loading and repair to achieve usable status.

The incremental addition workflow proceeds as follows: Binlog logs generated by the primary system are synchronized to the backup system in real time every 5 minutes. When the primary system fails, the backup system imports multiple binlog files generated after 2:00 AM that day to restore differential data to the most recent state. Based on binlog restoration, the system identifies the last write timestamp in the backup MySQL database, then uses MySQL' s built-in mysqlbinlog tool to export SQL files, which are imported into the MySQL database to complete data recovery.

3.4 Data Automatic Verification and Backfilling

The ElasticSearch database cluster serves multiple functions in the website publishing system architecture:

As a Full-Text Search Database: Built on the renowned open-source full-text database Lucene, ES naturally fulfills full-text search requirements.

As a Dynamic Content Display Platform: While large information websites typically present news display pages and homepages as static files, pages like list pages and "more content" pages are better suited for dynamic presentation. ES enables dynamic display when users browse column pages or content-rich pages.

As an Automated Data Verification Tool: Since the ES repository retains every article and its major versions published by editors, this characteristic can be leveraged to automatically verify database integrity after system failover migration to prevent article loss. This comparison process can be completed through pre-written scripts. The following details the business workflow when using ES as a data verification tool.

When the publishing system generates content data, it writes content to the MySQL database while simultaneously writing key content data to the ES cluster. The ES cluster employs a redundant architecture with high availability, where single-node failures do not affect cluster health. After the publishing system database recovery is completed, the data automatic checking program is activated. The checking program verifies business data from the failure time point to the present, examining each entry to confirm whether the MySQL

database contains the record. Unlike traditional MySQL verification, this process validates based on actual business data occurrence rather than MySQL' s data structure. Specifically, it retrieves each article' s information from ES in chronological order and searches for it in relevant MySQL tables. If the MySQL database lacks certain records, the missing content is supplemented into the MySQL database.

3.5 Backup Scheme Drills and Evaluation

Backup scheme feasibility requires validation through actual drills. For this system, we simulated both CMS primary node single-host failures and data center network failures.

Drill Methodology: For single-host failures, we used host shutdown; for data center network failures, we directly modified web system DNS resolution records.

Drill Results: As shown in , the CMS disaster recovery solution achieves RPO=5 minutes and RTO=5 minutes for web system failures, RPO=10 minutes and RTO=5 minutes for database single-host failures, and RPO=15 minutes and RTO=5 minutes for data center-level failures. In contrast, traditional disaster recovery solutions achieve RPO=5 minutes and RTO=5 minutes for web system failures, but require RPO=10 minutes and RTO<4 hours for database failures, and RPO=15 minutes and RTO<1 day for data center-level failures. The results demonstrate that for web system failures, both approaches achieve similar recovery times due to real-time file synchronization. However, for database or data center-level failures, the CMS disaster recovery solution significantly reduces both RPO and RTO compared to traditional approaches, representing a fundamental improvement in data availability. The current CMS disaster recovery solution can restore system operation within 10 minutes of a severe system failure and recover data to within 5 minutes of the failure point within 20 minutes.

(Author Affiliation: China News Service)

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