Featured Articles

Measures for Ensuring High Reliability in Public-sector Systems

Fumihiko Nii
Noriaki Nishimura
Takao Narita
Yasufumi Aburada

OVERVIEW: Large public-sector systems often incorporate a number of separate systems, with system considerations that include ensuring processing performance and maintaining consistency when systems interoperate, and business considerations that include the ability to control processing and permissions in ways that take account of legal restrictions and other factors. As large public-sector systems are currently undergoing an ongoing shift to open systems, this article uses a number of actual examples to describe measures for ensuring high reliability in open systems.

INTRODUCTION

PLANS for the optimization of processes and systems at government agencies[1), 2)] that were prompted by the e-Japan Strategy and e-Japan Strategy II published by the Japanese government in 2000 and 2003 respectively are still ongoing, and most of these involve a shift to open systems. Because the 2015 revision of the "Declaration to be the World's Most Advanced IT Nation" specifies policies for reducing operating costs[1), 2)] among the reforms to government information systems, it is anticipated that the trend toward the adoption of open systems, including cloud-based systems, will continue in the future.

Accordingly, because of the way the public sector thinks about reliability and the unique regulatory requirements and other operational characteristics that apply to the information systems of government agencies responsible for important public services, it is necessary for open systems to comply with these requirements as well. This article describes ways of thinking about reliability that are specific to the public sector, and techniques for achieving this reliability.

CHARACTERISTICS OF PUBLIC-SECTOR INFORMATION SYSTEMS

System Reliability

(1) Ensuring consistent processing performance

Because the public-sector information systems that administer activities such as taxation or pensions handle large amounts of important data, there are cases where not only system outages or loss of data but even delays in processing have the potential to impact the public.

For example, control of the relative priority of online and batch processing is needed to prevent over-the-counter services from being interrupted by the execution of high-volume batch processing during working hours. There are also cases when there is a need to control the relative priorities of different online processes or different batch jobs. In other words, in the context of public-sector systems, reliability also includes ensuring consistent processing performance.

(2) Ensuring consistency when business systems interoperate

Because large systems are made up of multiple individual business systems, there is a need to consider consistency across these systems.

(a) Synchronized message exchange

In cases that involve the exchange of messages between different business systems, the maintenance of data consistency requires that synchronization between sender and receiver be restored when there is a fault in the receiving system.

As this synchronization is not supported by standard message queue (MQ) products, providing such a function on open systems involves implementing the same message exchange function independently on each system. Unfortunately, difficulties arise that are specific to open systems if each system is from a different vendor, such as the need to take account of factors such as differences between the middleware in each business system or the use of a cluster configuration.

(b) Retry and cancelation of file transfers

If a transfer of files between business systems cannot be completed due to a fault in the target system, it is best to halt the transfer and then perform automatic retry once the system has recovered. To allow for cases when the status of the target system makes retrying undesirable, it is also necessary to have a function that cancels the transfer on the source system.

As this mechanism is not available in standard file transfer protocol (FTP) software, it presents the same problems as in synchronized message exchange.

(3) Fault analysis and operational statistics covering multiple servers

If a fault occurs on information systems that play a central role in providing services to the public, it is important to identify the cause and to restore system (and service) operation quickly. In the case of processing that involves the interoperation of a number of servers, however, because the information needed to analyze a fault is spread across these different servers, analyzing the cause is made difficult by having to use different applications and middleware that provide different levels of output and use different formats.

Accordingly, it is essential to be able to perform timely fault analysis using this information. Also required are statistical functions that can monitor the variation in processing load under normal conditions.

Application Characteristics

(1) Ensuring the ability to maintain application programs

Public-sector systems sometimes require program changes at short notice to coincide with the introduction of amendments to law. This makes it desirable to have individual programs that are highly independent, with minimal impact on each other.

To achieve this, it is necessary to adopt practices such as a service-oriented architecture (SOA) and to provide a framework under which each application program can run independently.

(2) Batch processing in accordance with legal deadlines

It is common for public-sector systems to handle procedures such as applications or payments that have deadlines, with batch processing often being matched with business procedures. Depending on the task, batch processing must be available either as batch jobs run by a data center or batch jobs run online.

Batch jobs that are run by a data center are used to process large volumes of data at high speed or for overnight processing of data that was updated online during the day. Because of the large volume of data involved, mechanisms are provided for halting batch jobs and restarting them on the following day so that the work can be split up and run over the course of several days.

Batch jobs that are run online are used in cases such as performing searches of large volumes of data based on criteria that a user enters from a screen, the output of large reports based on user-specified conditions, or when a batch job is invoked from data that is input online.

(3) Output report control

Public-sector systems include reports with information that, if leaked, could have a major impact on the public, such as administrative documents or address cards relating to city residents or the general public. Accordingly, the confidentiality of these reports means they need to be handled in ways that take account of security, including preventing them from being output unnecessarily or from being accessed by users other than those who are authorized to see them.

(4) Exclusive control of application programs

Because of factors such as legal restrictions on the processing order of application programs and limitations on the execution permissions of the staff responsible, public-sector systems include application programs that cannot be run concurrently and that require control of the order of their execution. Similarly, to prevent the impact of problems such as program errors or data corruption in the processing for a particular locality from spreading throughout the country, there is also a need to specify rules governing when processing can be executed for different regions, or to otherwise control execution.

This exclusive control of application programs and their execution normally cannot be implemented using the functions available on an application server, or even if it can be implemented to a certain extent, it is significantly difficult to maintain. Accordingly, it is very important to be able to provide systems with standardized middleware (common platform) so that these functions can be made available on vendor-specific middleware as application interfaces.

MEASURES FOR ENSURING RELIABILITY

Practices for Ensuring Reliability

Hitachi has adopted the following practices to provide the functions that these characteristics make necessary.

(1) Enhance the ability to maintain application programs while also providing functions that need to interoperate with application programs in the form of

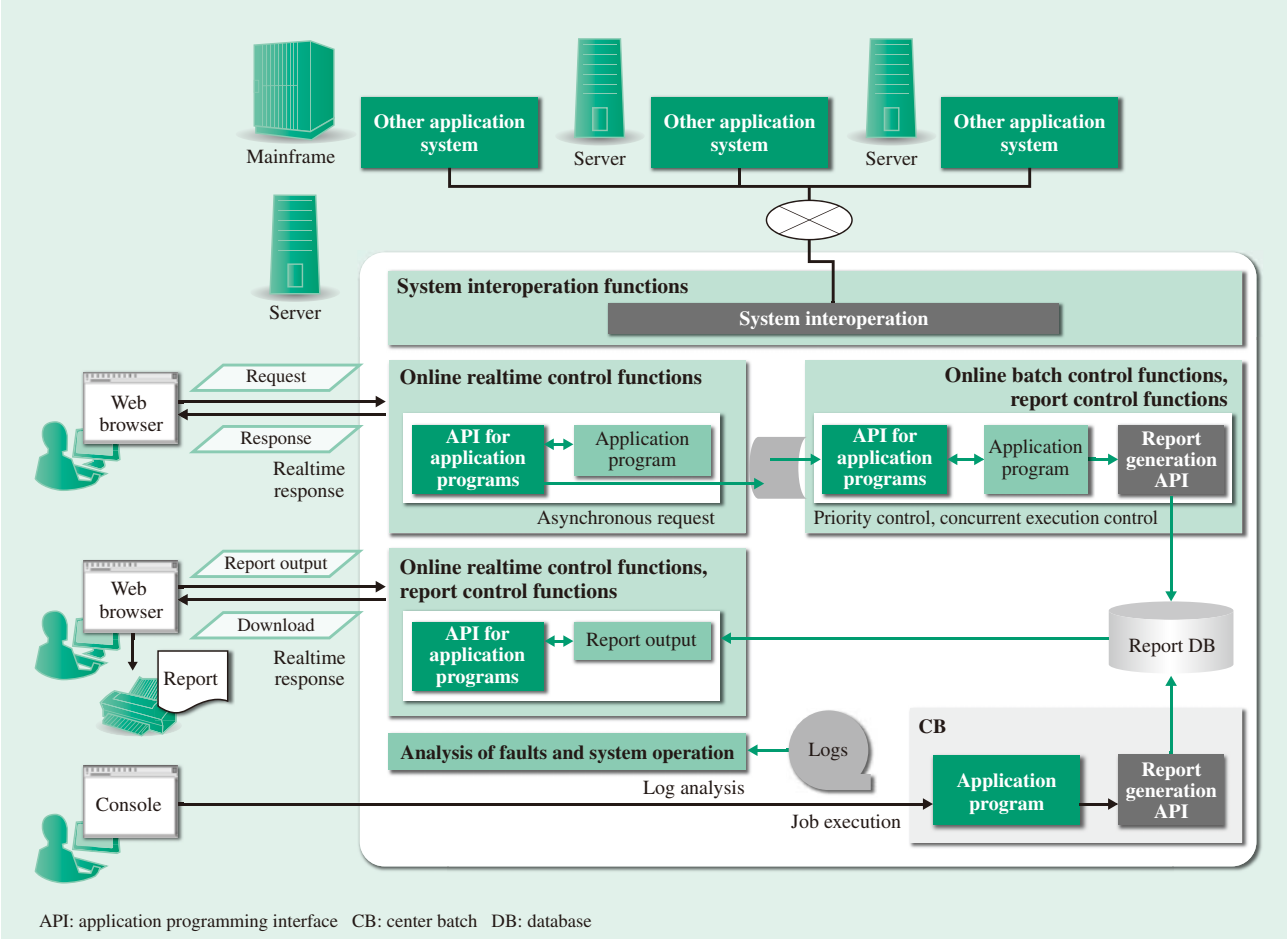

Fig. 1—Processing Functions of Common Platforms.
The figure shows a flowchart of the business processes performed using functions provided by the common platform.

an application framework. These functions include control functions for ensuring reliable performance; functions for interoperation between systems, such as message exchange and file transfer; functions that support the analysis of faults or system operation; functions for running batch jobs online; output report functions; and functions for exclusive control of application programs.

(2) Extend functions as needed for those aspects that are handled using existing utilities or other products, especially in the case of functions for running batch jobs at a data center.

Measures Adopted

(1) Application framework

Hitachi supplies an application framework with the following functions that it developed itself.

Developers can use the application programming interfaces (APIs) provided by the application framework to perform development in accordance with the business logic (see Fig. 1).

(a) Online realtime control functions

This API provides application processing results in realtime in response to requests from a web browser, and also handles tasks such as disabling application programs and checking whether they are permitted to run.

(b) Online batch control functions

This API runs application programs independently of realtime processing and includes multitasking control for priority-based scheduling and ensuring efficient use of server resources.

(c) Report control functions

This API includes functions for producing output reports such as address cards or administrative documents for the public. It also includes functions for managing the scope of access granted to users based on their access rights, and setting how long information is to be kept, and for enhancing the confidentiality of report data by working with End-User Reporting (EUR), a report management product, to prevent report data from being saved on client devices.

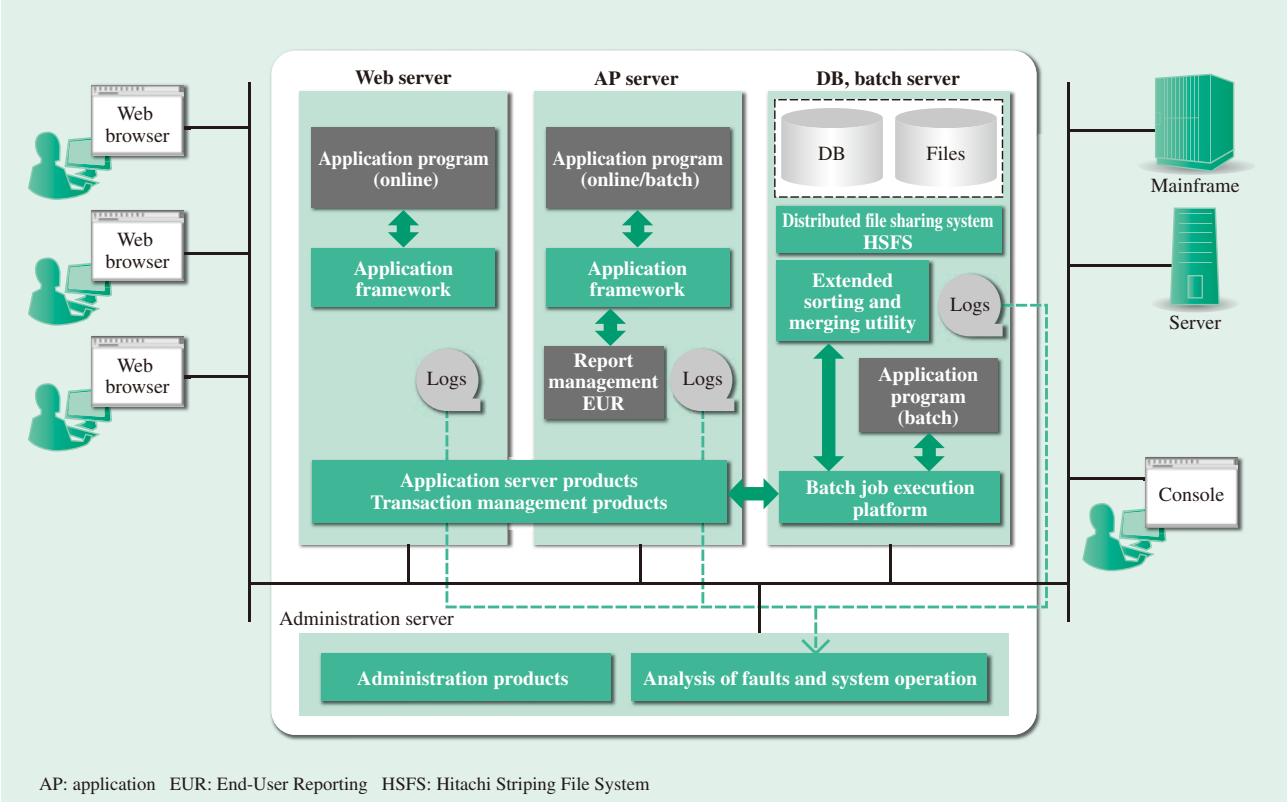

Fig. 2—Example Open System Configuration.
The open system is made up of a variety of servers with different roles, including web servers, AP servers, DBs, batch servers, and administration servers.

(d) System interoperation functions

To ensure the seamless coordination of data between business systems, this API provides functions for establishing communications with other systems as well as rollback and other mechanisms for keeping systems synchronized, while also keeping the influence on business applications to a minimum. To adapt flexibly to operational requirements, the functions work with both open systems and mainframes and can use either message passing for handling small amounts of data or file transfer for large amounts of data.

(e) Analysis of faults and system operation

To reduce the workload associated with consolidating and analyzing logs in the event of a fault that involves a number of servers, this API provides executable commands that can combine application logs and other journal data that is output by multiple servers and collate it in a time-series format.

Also, because of the need to identify in advance any over- or under-resourcing of servers or other bottlenecks when fluctuations in processing load are anticipated for reasons such as amendments to law, the API also provides executable commands that can collate information for analyzing things like the number and time-per-transaction of the transactions handled by each application.

(2) Batch jobs run by a data center

As public-sector systems require changes to their processing capacity, methods, etc. used to extract and edit data in order to process a large volume of data, Hitachi has made changes to products to achieve this (see Fig. 2).

(a) Extended sorting and merging utility

Extended sorting and merging utility can sort and merge different types of data quickly and efficiently using the same functions as a mainframe, including selecting, collating, and editing records and splitting files.

(b) Distributed file sharing system

Because open systems typically achieve their performance by operating a number of application and other servers in parallel, it is necessary to consider the effect that transferring data between servers will have on the network and how to use disk resources efficiently. Hitachi Striping File System is a distributed file sharing system that can combine a number of logical units into a single file system that can be accessed by multiple servers at the same time.

(c) Batch job execution platform

Batch job execution platform provides job definition, spooling and other functions on open systems. It enables existing job control language (JCL) to be reused when migrating from a mainframe to an open system.

CONCLUSIONS

This article has described how to ensure the reliability of public-sector systems that handle particularly large quantities of important data.

Along with ongoing work on optimization plans for the activities and systems of government agencies and the introduction from 2015 of the identification number system, it is anticipated that the scope of systems that demand high reliability will expand further in the future. Based on the technologies it has built up over time, Hitachi intends to respond to these requirements in ways that include enhancing its application framework and extending the functions it provides and the format in which it provides them according to the size of the user base.

REFERENCES

(1) Prime Minister of Japan and His Cabinet Website, http://japan.kantei.go.jp/index.html
(2) Ministry of Internal Affairs and Communications Website, http://www.soumu.go.jp/english/index.html

ABOUT THE AUTHORS

Fumihiko Nii

Government & Public Systems Department 1, Government Solution Operation 1, Government & Public Corporation Information Systems Division, Information & Telecommunication Systems Company, Hitachi, Ltd. He is currently engaged in the planning and development of application frameworks, and the marketing and implementation of public-sector systems.

Noriaki Nishimura

Government & Public Application Department 1, Government Solution Operation 1, Government & Public Corporation Information Systems Division, Information & Telecommunication Systems Company, Hitachi, Ltd. He is currently engaged in the planning and development of application frameworks.

Takao Narita

Government & Public Systems Department 1, Government Solution Operation 1, Government & Public Corporation Information Systems Division, Information & Telecommunication Systems Company, Hitachi, Ltd. He is currently engaged in the marketing and implementation of public-sector systems.

Yasufumi Aburada

System Service Department 5, Solution Division 4, Hitachi Government & Public Sector Systems, Ltd. He is currently engaged in the marketing and implementation of public-sector systems.