Delivering Business Continuity
In a High Reliability Database

Symfoware V10

In the current era, all forms of social and corporate activity are inseparable from IT systems. Continuity within those IT systems has become a critical requirement. When systems fail, not only does work stop, but global distribution of goods ceases, factory productivity is brought to a halt, and communication are impeded. The impact felt by system shutdowns increases with each passing year, leading to an ever-growing demand for business continuity. The role of the database as the cornerstone of IT systems means that the requirement for database systems to deliver business continuity is also increasing. IT systems built on a foundation of strong and stable database systems, can offer better stability and reliability. Corporate activities can then be carried out with the peace of mind that business opportunities will not be missed.

Symfoware Server is Fujitsu's relational database system. Symfoware Server has been pursuing the goal of business continuity for a number of years with a focus on 'creating operational system stability'. Based on technology cultivated since the era of mainframe computers, Symfoware Server has established a powerful track record, having been implemented on a variety of mission-critical systems, including open system environments.

Symfoware Server Version 10 continues to do more than simply aim for business continuity. This version adds a number of new features, including a 'load sharing degeneration' function that combines both stable system operation and a high level of system performance, a 'direct connection' function that delivers data processing up to 10 times the speeds previously available, as well as a 'multi-versioning' function. In addition, as a mission-critical database, Symfoware Server V10 is designed for even stronger functionality and comes with a 'database mirroring' function that achieves both reliable data protection and continuity of work.

This paper explains the features and technologies provided by Symfoware Server V10, the mission-critical database that delivers true business continuity.

- Symfoware is a registered trademark of the FUJITSU LIMITED.
- All other company names and product names used in this paper are the registered trademarks or trademarks of the respective companies.
- Trademark symbols (TM, (R)) may be omitted from the company names, system names, and product names mentioned in this paper.
# Table of Contents

1. **The Requirements of Mission-critical Systems and Symfoware Server Solutions** .............................................. 1
   1.1 The Requirements of Mission-critical Database Systems ................................................................. 1
   1.2 Symfoware Server Solutions for Realizing Mission-critical Systems ......................................................... 2

2. **Technology That Supports Business Continuity** ............................................................................................... 3
   2.1 Failover Handling Multiple Faults and Offering Flexible Operation Modes ........................................... 3
   2.2 Achieving Business Continuity with High-performance, High-scalability Load Sharing ....................... 5
   2.3 Database Mirroring that Realizes Stability and High-speed Switching .................................................. 6
   2.4 Disaster Solutions Capable of Coping with Large-scale Disaster Events ............................................... 9
   2.5 Connection Management Function ......................................................................................................... 11
   2.6 High-speed Backup / Recovery .............................................................................................................. 11
   2.7 Maintenance during Operation ................................................................................................................. 12

3. **Technology that Supports High Performance** .................................................................................................. 14
   3.1 Technology that Maximizes Access Performance .......................................................... 14
   3.2 Technology that Equalizes Response Times .................................................................................. 15
   3.3 Technology that Maintains Response Performance ........................................................................ 16
   3.4 Technology that Distributes Data (Technology that Allows System Scaling) ....................................... 17
   3.5 Technology that Improves Throughput ............................................................................................. 19

4. **Technology Supporting Security** ................................................................................................................... 21
   4.1 Security Functions that Comply with ISO15408 .............................................................................. 21

5. **Operation Tool** .............................................................................................................................................. 23

6. **Conclusion** ..................................................................................................................................................... 26
1. The Requirements of Mission-critical Systems and Symfoware Server Solutions

Mission-critical systems need to meet a number of non-functional requirements, including work continuity, stable performance, and data security. Having successfully addressed these kinds of requirements, Symfoware Server has been implemented in a wide variety of systems, including financial, medical, and goods-distribution systems.

1.1 The Requirements of Mission-critical Database Systems

Mission-critical database systems must fulfill a variety of requirements.

The first of these is business continuity. Within any corporate setting, core systems and information systems are both indispensable to the decision-making process, and therefore need to run on a non-stop basis. IT systems that connect directly to internet-based business and inter-corporate transactions require a similarly high degree of continuity.

Achieving this level of continuous operation would require bringing the number of unforeseen system stoppages down to zero. However, in as much as computer systems are created with physical elements in a physical structure, such a ‘zero stops’ status is not actually possible. What is therefore required is that, when unforeseen system shutdowns do occur, work operations can be restarted in as short a time as possible. In other words, the key is that business continuity is achieved and maintained.

Accordingly, the databases in systems that underpin business continuity need to run in a mirrored configuration with standby servers prepared to operate. Mission-critical databases also require advanced technology capable of high-speed failovers, so that when a system shutdown does occur, the time required to switch over to standby servers is minimized. Not only is it important for databases which experience a failure to recover quickly, but it is also important that stable business operations are re-established in as little time as possible without losing the valuable data stored on the system.

Furthermore, current business practices increasingly demand as short a period as possible for scheduled system stops, such as those required for maintenance or expansion. This demand translates into a need for technology that allows maintenance and system expansion activities to go ahead even as core operations continue, so that there is no halt on the services provided. Whereas the idea of mission-critical systems was previously restricted to a limited range of systems in a limited range of industries, such as financial, public services, and communications, currently a wider range of corporate entities are thinking in terms of non-stop operations. In particular, there has been an overwhelming increase in the number of businesses that now accept the premise that ‘Systems shutdowns may be unavoidable, but service shutdowns are unacceptable. Business operations must continue’. Additionally, the range of services provided by database systems is no longer limited to internal and business-to-business transactions, but has expanded via the Internet to directly involve consumers and the whole of wider society. Already, one third of all systems operating in businesses around the globe today provide services on a 24-hour, 365-days basis.

The second requirement of a mission-critical database is high performance. As time goes on, the volume of data that any enterprise handles increases. Along with expanding networks, data including email and images are being exchanged through business enterprises at an explosive rate of increase. Moreover, the speed at which business activities are conducted is ever increasing, and the expectations of end users grow so that a wait of even a few seconds for a response from a system is unacceptable. More and more, access to systems is occurring without restriction to the limitations of time or location.

In such circumstances, it is unacceptable to have a database simply stop operating. It is now an absolute requirement for any large-scale database system to be capable of delivering high performance mass data processing while guaranteeing a stable response in all areas of work.

The third important factor underpinning any mission-critical system is security. The data stored within a database is a valuable company asset, and it is critical that this asset is not leaked or altered in some way due to an external attack or a mistake from within the organization. The impact and damage caused by this type of accident or crime could be enormous. In addition to the work required to resolve the core problem, business activities could also be hindered by a potential loss of trust for the company.

Only a database capable providing business continuity, high performance, and security can deliver a mission-critical system.
1.2 Symfoware Server Solutions for Realizing Mission-critical Systems

While Symfoware Server is Fujitsu's mission-critical database, it is also the pivotal element in our IT systems solutions. Symfoware has already delivered significant program features, including massive storage capacity capable of accommodating 128 petabytes, and the fastest database loader in the world. Symfoware has also striven to offer business continuity in a variety of ways.

One of the technologies underpinning mission-critical systems is Symfoware's high-speed hot-standby function, which also relates directly to providing business continuity. Moreover, by adopting load sharing technology to distribute processing workloads using the Shared Nothing method, Symfoware Server offers flexible scalability in operating cluster systems. Another of Symfoware's significant features is its automated management of the connection between applications and the database. Since the connection is automatically re-established in the event of a connection error, systems recovery is also carried out in a short amount of time. Finally, Symfoware Server V10 continues to provide a great variety of solutions for meeting business continuity requirements, including Symfoware Active DB Guard for realizing disaster recovery, and database mirroring which is now included as a standard feature of the Symfoware Server suite.

Delivering high performance has also been one of Symfoware Server's strong points from the very start. One such example is the partitioning function. By dividing large-scale tables and indexes and then distributing them, processing time can be reduced. Additionally, in order to maximize access performance, Symfoware Server comes with technologies for increasing the concurrent execution of applications, such as staging control, buffer control, and in-memory storage of the database. Proprietary technologies called Dynamic Queue Control and Stealth Sync have also been developed to help equalize system responsiveness when the number of access attempts is significantly increased.

Symfoware Server V10 also comes with throughput improving technology, including Direct Connection, which improves data accessibility by enhancing the method used to connect application programs and the database, and Multi-Versioning, which improves the concurrent execution of data reference processing and update processing. These new functions have been added to further improve performance.

In the area of security, Symfoware Server also offers a variety of features that comply with International Information Security Standard ISO/IEC 15408, in order to guarantee database security. Examples of this are auditing trails and database access privileges that complies with ANSI/ISO SQL standards. This feature allows database users to select a level of security that matches the requirements of the system being used.

Symfoware Server also offers a variety of operation support tools that support mission-critical systems. Use of these easy-to-understand, easy-to-use tools facilitates the early detection and isolation of problems, leading to earlier system recovery. Moreover, by making use of a preventative monitoring function, potential failures can be identified and prevented before they arise.

Thus, the current version of Symfoware Server consolidates Symfoware's previous know-how and combines it with a structure that addresses the emerging needs of today's society. Symfoware Server offers all these features in a single, unified package to provide total support for achieving true business continuity.

From the following section, this white paper will introduce the range of technologies provided by Symfoware Server to achieve business continuity and superior database performance.
2. Technology That Supports Business Continuity

Symfoware Server comes with a variety of functions designed to accomplish business continuity. Aside from backup and recovery functions that protect data when system failure occurs, Symfoware Server allows the user to choose a variety of configurations, including redundancy, which prevents the system from shutting down when problems occur, thus offering the required level of high reliability.

The mirrored configurations that Symfoware Server provides can be classified broadly into the cluster method configuration and the mirror method configuration.

Within the cluster method, there are still a number of ways to define the configuration. Symfoware Server's cluster method places the data on a number of shared disks, with the database system (RDBMS) copied onto multiple servers. In other words, Symfoware Server uses a method that establishes redundancy using hardware units. This method allows the user to choose a server configuration suitable to their system needs, and to choose the timing used to execute failovers in the event of a systems failure. For example, it is possible to construct a system that is resistant to multiple faults, or to integrate your system with a high-performance, highly expandable load sharing system to create a mirrored configuration.

By contrast, the mirroring method used by Symfoware Server does not establish a hardware-based redundancy, but rather employs duplication (or ‘mirroring’) of the database itself. On one hand, duplicating the entire database means that the user no longer has the flexibility of choosing server configurations or the timing at which a failover is executed. On the other hand, this simple system configuration makes it possible to achieve instantaneous high-speed switching, regardless of where the failure or system problem occurs, and it is also easy to operate.

Aside from these options, Symfoware Server also offers disaster recovery functions capable of responding to a variety of disaster situations.

The following sections detail a variety of technologies that assist in achieving business continuity.

- Failover
- Load sharing
- Database mirroring
- Disaster recovery
- Connection management
- High-speed backup / recovery
- Maintenance during operation

2.1 Failover Handling Multiple Faults and Offering Flexible Operation Modes

Constructing a cluster system enables processing of large quantities of data from multiple transactions and offers business continuity at greater levels. Thus, the significant merit of a cluster system is its ability to split and distribute risks and workload using a mirrored configuration of hardware and software.

Symfoware Server uses cluster software, including PRIMECLUSTER for example, to enable a flexible selection of cluster operating modes that can respond to system requirements and budget (Figure 2-1).

![Failover modes](image)

Figure 2-1 A variety of operating modes

[One to one (1:1) standby mode]

One to one standby mode consists of keeping one standby server for each active server. This form of operation provides a higher level of reliability than other operating modes, because processing capabilities are maintained even if multiple active servers go down successively. This is most applicable for mission-critical OLTP core jobs and applications.
[Cascade mode]

This mode consists of keeping multiple standby servers for each active server. It is used to respond to double faults on servers.

[N:1 standby mode / N:M standby mode]

N:1 standby mode / N:M standby mode consists of keeping one standby server or multiple standby servers for multiple active servers. These configurations protect against the event of a server going down while saving on the cost of standby servers. They are most suitable for large-scale informational database systems such as data warehouses.

- **High-speed switching function (Failover)**

Failover refers to the function of having a standby server take over the data or processing of an active server in the event of a failure. Symfoware Server supports business continuity by providing a number of failover functions which are used to duplicate the system on both the active and standby servers.

There are 3 types of failover functions provided for Symfoware Server, classified according to the level of availability each provides: Standby, Hot-standby, and High-speed hot-standby.

- **[Standby]**

Symfoware Server is started on the standby server after the active server fails. This is also referred to as warm standby. Standby requires a certain amount of time to start the system and configure it.

- **[Hot-standby]**

The hot-standby function has the database environment which is running on the active server identically pre-installed and activated on the standby server before a failure occurs. When the system fails, the client executes reconnection processing for Symfoware Server after the standby server takes over the application, which allows jobs restart smoothly.

In comparison to the standby function, which requires waiting time for the system to be started on the standby server, the hot-standby function significantly improves availability (Figure 2-2).

![Figure 2-2 Hot-standby](image)

- **[High-speed hot-standby]**

By using Flash Treatment Recovery, as described below, Symfoware Server is able to switch to the standby server within seconds. This function makes it possible to have the standby server take over the job far more quickly than in regular hot-standby.

For example, in the case of the systems in Japanese financial institutions, the standard time for a failover in the event of a systems failure has been established at 30 seconds. High-speed hot-standby using Flash Treatment Recovery has achieved switchover of the database itself at actual speeds of just over 4 seconds, on top of which isolation and restore of data currently being processed is carried out at a similarly high speed. Accordingly, this configuration delivers a quick and complete failover of the total system in less than 30 seconds (Figure 2-3).
Figure 2-3 High-speed hot-standby using Flash Treatment Recovery

[Pre-opening on the standby server]
Pre-opening is one function provided in order to achieve smoother execution of hot-standby. On the standby server, this option opens the same database environment as the active server in advance. This removes the need to set up the runtime environment in the standby server when switching from the active server to the standby server, thereby allowing job processing to reopen quickly. This particular function is a direct carry over from the technology of mainframe computers.

[Cache recovery]
When using a hot-standby configuration to switch the operation to the standby server, Symfoware Server offers ‘Cache recovery’, which restores the database from memory. Cache recovery is different from pre-opening the database on the standby server, and is accomplished using advanced technology not available during the mainframe era.

Symfoware Server makes it possible to define a buffer pool in the standby server for reading in the database in the same way as the active server. As it is possible to use this large capacity buffer pool for down recovery during failover, it improves the on-buffer effect, resulting in a higher input/output database efficiency than when the default buffer pool is used. In addition, because this user-defined buffer pool can also be used for operations after recovery, jobs can resume promptly.

2.2 Achieving Business Continuity with High-performance, High-scalability Load Sharing

Load sharing uses a cluster configuration to distribute work load by enabling multiple nodes to process transactions in parallel. Not only is load sharing superior to other modes of operation in terms of scalability and performance, it also delivers superior business continuity. The following sections describe load sharing technology from the perspective of business continuity.

Sections below will provide a description of load sharing’s special features in terms of scalability and high performance.

- Load sharing degeneration
Symfoware Server V10 offers a load sharing degeneration function with enhanced workload distribution and increased availability. The new version of Symfoware makes it possible to use a system configuration that has no standby servers, which were previously required for mirrored configurations. When an active server goes down, its jobs are taken over by other active servers. For example, even in a case where multiple faults occur, workload is distributed up until the last node, with the result that it is possible to achieve stable continuation of the work (Figure 2-4).

Figure 2-4 Load sharing degeneration
Flash Treatment Recovery

Flash Treatment Recovery sends a transaction log to the memory of the standby server via high-speed connection between the servers. When a failover occurs, there is no need to read the log from the disk, because an up-to-date log already exists in memory.

As the number of servers increases, the processing capacity of the system increases on a scalable basis. However, to the same extent, the likelihood of server failures also increases. For this reason, rapid switching from one server to another needs to be performed if a server goes down. Using the Flash Treatment Recovery function, it is possible to reduce the time required for switching between servers by eliminating, to a large extent, the need to read transaction logs.

Flash Treatment Recovery uses the Delayed Transaction Recovery method when recovering for down recovery. Using this technology, the recovery of tables and indexes of transactions in progress is carried out in a shared buffer and reflected in the database in the background. Operations can resume within a matter of seconds, even in the case of a server failing.

When switches between servers can take place in seconds, the effect on work continuity is almost imperceptible. There are a large number of systems that do not permit even a momentary stop in operations due to system failure, including financial, medical, and public service systems. Symfoware Server achieves the high level of 'no down, no stop' operation required by such systems.

Exclusive control during load sharing

Symfoware Server has adopted the Shared Nothing method for processing transactions in parallel. This method executes exclusive processing on each individual server that manages the data. Accordingly, even if there is data access across servers, there is no need for exclusive control to be aware of such access. Furthermore, Symfoware Server automatically detects deadlock across servers and cancels the responsible transactions.

Should the applications involved in the deadlock be unknown, Symfoware can identify the applications and the servers on which they run using a command that monitors the occurrence of deadlocks.

Parallel down recovery

Symfoware Server also applies parallel technology in the down recovery process. This function is known as parallel down recovery.

The dedicated thread for reading the log is activated first, and immediately starts to read the log. After that, the optimal number of recovery threads for applying the log to the database is activated depending on the number of partitioned tables and indexes. The individual recovery threads execute the log application processing in parallel according to the read log.

Since reading the log and database recovery processing are executed in parallel, this method can dramatically reduce the time needed for recovery.

2-phase commit control

In Symfoware Server's load share operation, tables or indexes are managed by partitioned units. In some cases, data may be updated across multiple servers. As this kind of transaction is automatically controlled by 2-phase commit control, consistency of the transactions is guaranteed.

2.3 Database Mirroring that Realizes Stability and High-speed Switching

Database mirroring is a solution for achieving secure data protection and business continuity (Figure 2-5).
Database mirroring provides the following features.

- **Protection of job data even in the case of multiple failures**
  
  Symfoware Server was the first to duplicate databases on open systems. Existing hot-standby or cluster systems sought to increase availability by duplicating the hardware, that is, the servers. Here, Symfoware Server goes further to duplicate middleware, namely the database itself. In database mirroring, in which disks are no longer shared, but kept on two separate servers, loss of important data can be prevented even when a disk failure occurs. TCP/IP is used to carry out data transfers between the two databases to continually maintain the mirroring status.

- **Application of Fujitsu's proprietary log shipping method**
  
  In database mirroring, Symfoware Server adopts a proprietary log shipping method that mirrors only the log required for reproducing the database, that is, only the database update results. Only the table update results are mirrored, and the logical update is executed once again on the secondary server side database. In this way, transfer volumes are reduced and the results can be reflected in the log in real-time.

- **Reliable and secure service continuity**
  
  Previous forms of failover involved the idea of 'switching' to a standby server after the active server goes down. This concept meant that time-consuming production requirements were necessary to achieve this 'switching'. In contrast, Database Mirroring’s approach does not involve sharing the database or the disks on which it runs, so that when the primary server goes down, the database that triggered the failure is detached and the system 'degenerates'. Since degeneration is executed instead of switching, operations resume both quickly and reliably. (Figure 2-6)

- **Automatic high-speed degeneration**
  
  Symfoware Server Mirroring Controller uses a storage copying function to allow even faster automatic degeneration.

The optional Symfoware Server product, Mirroring Controller, offers a variety of functions to effectively utilize Symfoware's database mirroring function.
degeneration. This method has been found to successfully reduce the time required to resume operations down to less than 1 second (actual results achieved in Fujitsu's verification modeling).

Furthermore, by combining this extra function with Symfoware Server's standard database mirroring functions, it is possible to automatically execute degeneration even with database mirroring methods that use TCP/IP.

![Figure 2-7 Database mirroring using remote storage copying function](image)

- **Avoiding unused assets**

  Symfoware Server Mirroring Controller can also be used to more effectively utilize secondary server. Secondary server are real-time backup systems that can be used as is without recovery. Accordingly, because the secondary server is also a database replica, it can be utilized for a variety of useful tasks, including outputting reports and creating data warehouses.
2.4 Disaster Solutions Capable of Coping with Large-scale Disaster Events

As a solution for disaster recovery, Symfoware Server offers Symfoware Active DB Guard. Symfoware Active DB Guard is a solution for rapidly and reliably resuming operations by employing a backup centre construction. In order to accomplish reliable continuity of service, a backup centre is constructed by operating the database in tandem with a storage facility for large-scale database systems that process heavy workloads (Figure 2-8).

Figure 2-8 Backup centre construction

Symfoware Active DB Guard offers 3 special features.

- **No downtime and restoration of database with zero data loss**
  By implementing the first ever completely tandem operation of database software and storage facility in an open system, Symfoware Active DB Guard achieves real-time center linking. Furthermore, using Fujitsu's proprietary logical log shipping method, both centers operate independently while the backup centre is utilized on a daily basis, thus making it possible to avoid the pitfall of operating the backup center as an unutilized, standby-only asset.

- **Reduced communication costs to as little as one-eighth using Fujitsu's proprietary linked centre log system**
  Symfoware Active DB Guard links system centers using optimal data transfer volumes for updating the database by acquiring only the necessary data from the database's differential log. Compared with previous methods, which used the storage center to copy the entire data area, this method makes it possible to operate the backup center with around one-eighth of the line costs (based on comparisons using internal Fujitsu modeling).

- **Effective utilization of the backup center using planned switching and static database**
  Symfoware Active DB Guard demonstrates its true power by bringing an end to scheduled system stops. By having the backup center take over the database service on a scheduled basis, maintenance work that was previously carried out only by stopping the service can be now be carried out while continuing to provide services. This includes scheduled maintenance of the hardware facilities and software, optimization (reorganization) of the database, as well as legal inspections resulting from power outages. Furthermore, work involving the aggregation of static data can be carried out at the backup center at a time chosen by the user without affecting online services.

  [Examples for utilizing the backup center]
  Symfoware Active DB Guard also makes it possible to use the remote site for other reference-related jobs. While maintaining its disaster response functions, the remote site can also be used for aggregation processing of an accounting system or data extraction for use in data warehouses (Figure 2-9, 2-10).

  Likewise, the reproduced database from the remote site could be taken offline and used for an informational resource.
Figure 2-9 Backup center utilization example 1

Figure 2-10 Backup center utilization example 2
2.5 Connection Management Function

Symfoware Server comes with the connection management function that automatically controls the connection between applications and the database server in 3-tier systems comprised of the client, application server, and database server (Figure 2-11).

Figure 2-11 Connection management function

As the connection is re-established internally, even in the event of an error, there is no need to manually re-establish the connection or expressly monitor the status of the database server. Application development productivity can also be enhanced, as there is no need for developers to create applications that carry out special processes like connection collection.

For example, by implementing the connection management function in a cluster system, it is possible to achieve automatic switching of the connection from the application to the database. In such a case, the application only needs to know the business logic behind the job, and does not need to be aware of whether it is connecting to the active server or to the standby server. Using the connection management function, it becomes possible to construct a system that delivers both superior reliability and scalability.

- Establishing connections
  
  When operating load sharing, it is possible to connect directly to the node responsible for the resources required. When a job needs to draw on a number of different database resources, the load can be divided up and the connection established according to how the resources are distributed across the related nodes. Consequently, with improved linear processing power to respond to the number of servers, partition logic does not need to be executed by application.

- Automatic connection switching
  
  The connection management function makes it possible to establish a connection with the standby server in advance. In this way, the time required for failover can be reduced to as much as one-sixth of the time previously required, even in the event of an unforeseen problem when operating in hot-standby mode.

- Automatic connection collection
  
  The connection management function makes it possible to automatically collect connections if the application server goes down or a malfunction occurs in the communication lines. There is no need to wait for the connection to be collected, because the transaction currently being executed on the application server is re-executed on a different application server.

2.6 High-speed Backup / Recovery

- Reverse creation
  
  Symfoware Server utilizes a configuration that is resistant to double failure by allowing each of the database management file, log file, and the database to be allocated to separate disks.

  Furthermore, in the event of a failure with the database management file, Symfoware Server uses a highly reliable technology called reverse creation to restore the file directly from the database. Operations can be resumed quickly and without reconstructing the entire database.
In database systems from other vendors, the entire database needs to be reconstructed if a failure occurs in the log file or the database management file. Since the time required for entire reconstruction of the database increases in proportion to its size, large-scale database systems require a substantial length of time. In an actual case where the RDBMS management file of a certain corporation was corrupted, it was reported that recovery of the system required 4 days. In contrast, when the same scenario was simulated using Symfoware Server, the result was system recovery within approximately 2 hours.

- **Incremental recovery**

  In combination with reverse creation, which restores the database management file from the database, incremental recovery makes it possible to further reduce the time required to restore data. Incremental recovery refers to quickly restoring only that data which is necessary for resuming the job with the highest priority. The database management file with the highest priority is restored at high speed, while the restore of the database management file required for other tasks is carried out while the priority job continues execution. By providing a function which executes the restore processing incrementally, execution of the job with the highest priority is resumed in the shortest time without waiting for all other data to be restored.

- **High-speed recovery using storage functions (ETERNUS link)**

  Making use of the Advanced Copy function of the ETERNUS disk storage system, Symfoware Server allows the fastest possible backup of large volume database systems without halting application operation. Even if a disk failure occurs, recovery can be implemented at high speed.

  As an actual example, a local municipality in Japan that uses this function for their document management system is able to backup a massive 800 gigabytes of data in as little as 10 seconds.

  Medical facilities that use this function for the electronic clinical records system are able to complete backup of one terabyte of data within one minute, allowing them to conduct reliable backup operations every day.

2.7 Maintenance during Operation

Symfoware Server allows the user to alter definitions according to changes in jobs without stopping operations, even if, for example, columns need to be added. Furthermore, using autonomic functions that automatically carry out expansion and reorganization of data areas, data area shortages and the deterioration of response time can be automatically avoided during operation.

Using the high-end, practical functions indicated in the table below, it is possible to establish flexible responses to sudden operational changes.

<table>
<thead>
<tr>
<th>Alterations possible during operation</th>
<th>Automated tasks executed by autonomic functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add partitions</td>
<td>Reutilization of vacant table</td>
</tr>
<tr>
<td>Add indexes</td>
<td>Reorganization of indexes</td>
</tr>
<tr>
<td>Add tables</td>
<td>Extension of table functions</td>
</tr>
<tr>
<td>Add columns</td>
<td>Extension of index areas</td>
</tr>
<tr>
<td>Partition activity maintenance</td>
<td></td>
</tr>
</tbody>
</table>

- **Extension of job applications during operations (adding partitions)**

  Symfoware Server has included partitioning as a standard feature since it was first created. It is possible to add partitions during operation without affecting the applications. This delivers a way to flexibly respond to the extension of job applications, such as those that may be conducted by a retail outlet (Figure 2-12).
Applications can execute processes to add partitions or columns.

Figure 2-12 Adding partitions

- **Autonomic functions (automatic extension of tables, indexes)**
  
  Symfoware Server automatically carries out the extension of tables and index regions during application operations. In this way, operational errors and the deterioration of response times may be avoided before they arise. Furthermore, Symfoware Server notifies the systems administrator with an alarm at the time area extensions are made (Figure 2-13).

Figure 2-13 Autonomic functions

- **Maintenance operations carried out during job processing**
  
  Using the server switching function in cluster systems, maintenance operations such as applying patches, can be performed without stopping jobs. This method is referred to as a rolling update (Figure 2-14).

Figure 2-14 Rolling update
3. Technology that Supports High Performance

Symfoware Server technologies achieve consistently fast and stable system response times, and can be categorized by their ability to:

- Maximize access performance
- Equalize response times
- Maintain response performance
- Distribute data
- Improve throughput

3.1 Technology that Maximizes Access Performance

Symfoware Server includes a variety of functions that maximize the power of access performance. These functions may allow Symfoware Server itself to increase the speed of processing, or have applications that increase the speed of processing.

The function which increases processing speed from the application is an exclusive specification that improves the application’s ability to execute concurrent processes. Increasing simultaneous execution improves the processing power of the entire system and thus increases speed.

The following section describes how Symfoware Server increases processing speed.

- **In-memory database (memory-resident database)**

  Symfoware Server comes with an in-memory database that provides fast data access.

  The in-memory database enhances data access responsiveness by keeping table or index data loaded in memory and removing the need for database I/O processing (Figure 3-1). In comparison with disk access speeds, this method of accessing only the data loaded in memory increases system response times by a magnitude of roughly 10 (actual values returned from Fujitsu’s in-house testing labs).

  Furthermore, in contrast to the in-memory database systems from other vendors, which work on a table unit basis, Symfoware Server loads and keeps all the required data in memory, thereby always achieving fast response times.

  Moreover, in Symfoware Server, this function does not require the user to pre-design the operation, but can be executed ‘with a single touch’ during database operation. The result is that users are able to execute operations flexibly in accordance with system demands.

  The in-memory database is particularly effective when used with indexes or data of control system jobs that handle large volumes of data.

- **Staging control**

  When creating, reorganizing, making a backup of, or recovering a database, Symfoware Server executes each of the internal processes separately (i.e., reading from/writing to a database, or conversion) and controls the flow of data among them. This is called pipeline control. By executing internal processes in parallel, Symfoware Server makes it possible to speed up processing for large-volume databases and enables multiple users to use the database concurrently.

  Moreover, Symfoware Server organizes internal database system processes into units, uses a proprietary
dispatch function called staging control to control CPUs, and pipelines the processing of each unit.

Staging control takes advantage of multi-threading features, and minimizes dispatch costs to the OS incurred in switching the run unit. Using this technology, Symfoware Server draws on the merits of pipeline control and maximizes the capabilities of Symmetric Multiprocessing (SMP) configurations (Figure 3-2).

![Figure 3-2 Staging control](image)

- **Buffer control**
  - The challenge of parallel processing occurs when multiple transactions access the same table and give rise to simultaneous input/output requests to the disk. As a result, mutually exclusive transactions have to queue, and response times are therefore slow. The adverse effects of this situation become greater in proportion to the number of transactions being executed.
  
  In order to avoid this deterioration in response times, Symfoware Server makes use of partitioning technology that divides up tables and indexes during operation. As will be described below, partitioning the tables and indexes makes it easier to execute parallel processing by reducing the size of the operating unit. Partitioned tables and indexes can be processed internally on an independent basis, thereby improving the ability to execute multiple transactions in parallel and distribute input/output disk accesses over the partitions. Furthermore, because the unit of exclusivity is divided as well, queuing of mutually exclusive transactions is also less likely to occur.
  
  Moreover, Symfoware Server is able to allocate database buffers to the individual partitioned table and index units. In this way, competition between database buffers is avoided, the residency rate for buffers is raised, and the number of disk input/output requests is reduced. This is called buffer control. In addition, when executing disk input/output, multiple sets of data are processed in bulk to raise input/output efficiency. Since Symfoware Server adopts a number of such adjustable technologies, it is possible to maintain a high-level of database performance in the face of differing variables.

### 3.2 Technology that Equalizes Response Times

Hardware performance can be markedly enhanced by improving the performance of individual CPU units and increasing the number of CPUs in the SMP configuration. However, if this enhanced performance is not properly utilized, cost performance suffers and processing times become difficult to estimate. Furthermore, it is important to prevent fluctuations in the system's response times if processing efficiency is to be achieved for jobs that process a large number of online transactions.

Symfoware Server carries out task management that evenly distributes the workload over each CPU, thereby guaranteeing better responsiveness in proportion to the number of CPUs in the system.

- **Dynamic queue control**
  - Symfoware Server allocates the processing for newly-initiated transactions to a CPU with a low workload. This is done so that the workload is evenly distributed among the available CPUs. However, because executed transactions generally end at different points in time, as time passes, fluctuations in workload will still occur among the CPUs. As soon as an uneven distribution of workload occurs between the CPUs, Symfoware Server automatically redistributes the transactions over the CPUs. Using this dynamic queue control function continually ensures that the load allocated to each CPU is roughly equal, thus guaranteeing stable system response times.
Stealth Sync

Symfoware Server includes a function known as Stealth Sync. This function writes data in the database cache to the physical disk both quickly and effectively. Using this function, response time fluctuations can be avoided when large volumes of update transactions occur, so that stable response times are maintained even when the system is dealing with a heavy workload.

With normal jobs, most update transactions are processed in batch mode when the OLTP traffic is low. Previously, however, this kind of batch processing still resulted in fluctuations in the OLTP response times. Using Stealth Sync, response times can be maintained with OLTP processing unaffected by the batch processing of update transactions (Figure 3-3).

3.3 Technology that Maintains Response Performance

After a certain period has elapsed since the start of system operations, it is increasingly likely that data updates will lead to fragmentation and deteriorating performance.

By automatically eliminating such fragmentation, Symfoware Server is able to consistently and continually deliver the same level of system responsiveness over time.

Automatic reorganization using Autonomic GC technology

In order to continuously deliver the same level of database response performance, it is necessary to regularly rearrange the data stored in the database. Autonomic GC automatically eliminates fragmentation of space even while job applications are being executed.

To ensure that indexes are always stored in the most optimal way, Symfoware Server includes a function that automatically rearranges stored data without affecting job applications running on the system. This function makes it possible to minimize the effect of performance deterioration caused by the fragmentation of data, which in turn means that stable system response times are maintained over long periods (Figure 3-4).
3.4 Technology that Distributes Data  
(Technology that Allows System Scaling)

Even if system response times are both equalized and maintained, if the volume of data on the system or the number of requests increases, bottlenecks in the hardware may eventuate. To avoid this, it is necessary to distribute the requests before they over-utilize the hardware.

Symfoware Server divides up the data into ‘tables and indexes’, ‘logs’ and ‘servers’ and distributes the requests. In this way, it is possible to construct highly efficient systems because the system can be scaled on demand in response to business expansion (Figure 3-5).

![Figure 3-5 System scaling](image)

- **Partitioning (dispersion of tables and indexes)**

  Symfoware Server includes partitioning functions that partition tables and indexes for operation. Partitioning technology has been an integral part of Symfoware Server since the software was first developed.

  Dividing large-scale tables and indexes and then distributing them results in reduced processing time. Because the range of the process is localized based on the partitioned unit, steady system response times can be maintained even as requests increase.

  Furthermore, if a disk failure should occur, the use of partitioning means that the effect of the failure is also localized. It is possible to isolate the problem partition and initiate recovery processes while continuing systems operation. Moreover, maintenance operations (such as backups of the tables and indexes and optimization of the database) can be performed independently to the partitioned unit, allowing parallel processing.

  Recently, the increasing volumes of data being handled in databases have resulted in other vendors recognizing the necessity of partitioning and adopting this technology. Nevertheless, Symfoware Server, having inherited and continued the high availability technology of mainframe computers, has continuously made use of partitioning as a standard feature (Figure 3-6).

![Figure 3-6 Partitioning](image)

- **Scalable logs (dispersed logs)**

  Even if processing is distributed for partitioned tables and indexes, bottlenecks may still arise as the use of logs increases.
On the other hand, if logs are partitioned and distributed in the same way as tables and indexes, the range of processes used for logs can also be localized. Such an approach enhances the ability of the system to concurrently execute transactions and makes it possible to maintain steady response times across the system.

Symfoware Server partitions and arranges logs using scalable logs. This is also a standard feature. Scalable logs take the two types of log used by Symfoware Server, bring them together as application units, and partition them. Not only do scalable logs provide faster processing in the system, they also have the effect of reducing the time required for maintenance operations like systems recovery and backup operations. This is because backup processing can then also be partitioned and distributed (Figure 3-7).

![Figure 3-7 Scalable logs](image)

**Load sharing (distribution of servers)**

While partitioning and scalable logs are both effective in systems that handle massive volumes of data and a huge number of requests, load sharing is also highly effective. Load sharing refers to distributing the system’s workload. Using the load sharing function, Symfoware Server is able to take a system configuration comprised of multiple active servers and handle them logically as a single database.

Load sharing uses two method types: the Shared Everything method and the Shared Nothing method. Of these, the Shared Nothing method is the one adopted by Symfoware Server. Shared Nothing describes an approach where there is no sharing of memory or disks between servers, and where processing is executed on each individual server.

In the Shared Nothing method, tables and indexes are divided, and managing servers are determined according to the division unit. Application programs have direct access to data under a managing server, and can access data under other servers via the automatic routing system. Accordingly, application programs can access data without needing to know where tables or indexes are stored.

The greatest merit of adopting the Shared Nothing method in a system is that performance increases linearly in relation to the increase in the number of servers in the system. Each server is connected via a high-speed interconnect, and in combination with increased speeds derived from parallel processing, the method achieves a high level of total throughput performance. If a database system is running out of capacity because the number of users increases or there is an increase in the workload, servers can be added without interrupting database system operation. This allows the user to increase the processing capability of the system without any adverse effect on the running database system. In other words, the Shared Nothing method is highly effective for realizing large-scale parallel processing and ensuring system scalability. (Figure 3-8)

![Figure 3-8 Scale-out solutions](image)

By contrast, the Shared Everything method involves significantly greater overheads, so there is a limit to how
much adding servers will benefit the system. In the end, even though the system may be scaled, the effect of the expansion is not clear.

If a business should expand suddenly, then the RDBMS underpinning that business simply cannot accommodate the expansion, and that can be a problem. In particular, system scalability is an important issue for businesses using the Internet. If the business runs Symfoware Server, however, with its ability to utilize real scalability in response to an increased number of servers, this is not a concern.

In Shared Nothing, operations for maintaining and preserving data are executed according to the division of the tables and indexes that have been distributed over multiple servers. Moreover, there are no conflicts of database buffers among the servers because data is handled by managing servers in a remote access system.

When combined with partitioning, Shared Nothing allows the operation of each server to be kept largely independent, because the data is distributed according to partition units. This is known as scalable partitioning.

3.5 Technology that Improves Throughput

In a database, throughput refers to the system’s processing capacity to handle data references or data updates within a specific period of time. As with responsiveness, throughput is an important factor in realizing a high level of database performance. Symfoware Server V10 includes new technologies for achieving increased speeds when executing data references or updates, and delivers ten times the processing performance previously available. Direct Connect and Multi Versioning are technologies that truly display their effectiveness when used for jobs that frequently reference or update the same data content, such as in the case of financial product transactions.

- **Direct Connect**

  Direct Connect improves the performance of the connection between applications and the database by completely separating the processes that monitor connection conditions from those that manage the connection between applications and the database. In earlier versions of Symfoware Server, management of the connection between application and database and monitoring of the connection status itself were both executed in the same process run on the application server. With this approach, the connection monitoring process would unavoidably intervene whenever a request was initiated, thus becoming a factor affecting data access performance. By separating the connection and monitoring processes, Symfoware Server V10 creates a direct connection from the application to the database. The CPU on the application server is only used to connect to the database, thus improving access performance.

- **Multi Versioning**

  Multi Versioning is a technology that improves data processing efficiency using version management of records in cases where there is a conflict between a request to reference a record and a request to update the same record in the database. When a request to reference a record clashes with another request to update the same record, the conventional solution has been transaction delay controls, which ensure the integrity of the database. Multi Versioning works by having the reference request refer to the most recently committed data while the update request moves ahead, so that both requests run in parallel. (Figure 3-9).

![Figure 3-9 Multi Versioning](image-url)
Symfoware Server implements Multi Versioning as the storage structure used for databases. In the case of other vendors incremental reflection in the log is used to create a different record for each version and to keep it in memory, starting from several generations back. In contrast, Symfoware Server takes the most recent data and the data from one generation back and keeps them within the same record. The result is optimized CPU efficiency and memory usage, and Symfoware Server achieves significantly improved I/O efficiency in comparison to the systems from other vendors (Figure 3-10).

Figure 3-10 Multi Versioning implementation differences
4. Technology Supporting Security

Interest in the link between social responsibility and corporate security has grown increasingly with the recent enactment of both the Personal Information Protection Law and the Japanese version of the SOX law. Security functions are essential to protect important information stored in databases. IT infrastructure performing job processing must be robust and equipped to protect data from damage caused by illegal access, theft, illegal removal, and interception of communications. Symfoware Server provides a wide range of essential security measures and compliance with legal requirements (Figure 4-1).

**Figure 4-1 Providing security**

### 4.1 Security Functions that Comply with ISO15408

In order to protect valuable data from the various threats to databases, Symfoware Server provides robust security functions that comply with the international information standard ISO/IEC 15408. As well as protecting against external attacks, it is also equipped with functions that prevent information leaks from within companies.

Symfoware Server Enterprise Extended Edition has obtained certification as a product that conforms to ISO/IEC 15408, as it meets and maintains the high level of outstanding security robustness required.

- **User control**
  
  User control is the basis of all security. Symfoware Server registers, authenticates, and identifies users. It can also preemptively prevent both access to unauthorized databases and the execution of unauthorized functions. Symfoware Server’s user access control complies with ANSI/ISO SQL 2003. It can perform OS-independent authentication and identification of users, and control usable resources.

- **Resource control**
  
  Symfoware Server manages its resources (such as memory, files and raw devices) and controls their direct access through OS functions. It can therefore prevent information leaks and the obstruction of database processing operations.

- **Audit trail**
  
  Symfoware Server can record in an audit log and analyze the processing operations performed by the system administrator and users. This enables detection of abuse of privileges by the system administrator and illegal access by users. The log contains detailed records of operations on databases (including executed SQL statements and parameters.)

  The Audit trail can also retrieve SQL statements and entered variables, and enable identification of the type of information accessed. As it links with the client information managed by Interstage Application Server, it can also identify the user who performed the operations. In other words, by referencing the audit logs on the database server and the application server log, Symfoware Server enables the cross-sectional identification of who has accessed what sort of information when and to what extent (Figure 4-2).
The audit log is managed as a single database, can be queried with SQL, and have backups created and restored.

Figure 4-2 Audit trail

- **Data encryption**
  Data stored in the database is automatically encrypted and decrypted. By encrypting only items for which confidentiality is a high priority, Symfoware Server also allows a quick search without reducing the response levels.

- **Database security**
  The adoption of the database security product, PISO, enables the recording of database access, warnings, and follow-up inspections. This serves as an audit trail. PISO is a tool that allows the collection of access logs via memory access and asynchronous methods, while only imposing a light load level. It can also monitor and provide warnings about real-time illegal access at the SQL statement level using simple operations.
5. Operation Tool

Symfoware Server includes tools that support the full lifecycle of system development, including installation, building, and operation (Figure 5-1).

During installation, Symfoware Server preemptively prevents problems arising from configuration errors simply by requiring that certain items (such as hardware specifications and system scale) be selected for automatic configuration of the appropriate environment. During the build stage, it is possible to check beforehand that the developed program can run without any problems in the operating environment. After operation has started, daily performance and statistics information can be automatically collected and stored at regular intervals. In addition to monitoring for signs of problems, this also reduces the time spent investigating and dealing with performance problems when they do occur.

- **GUI installer**
  
  This is a GUI installer that supports the installation phase. It allows simple installation just by entering an item in 3 screens and by simple selection in other screens.

- **Setup support tool (WebAdmin)**
  
  This is a GUI tool that supports the installation phase and simplifies setup. It aims to reduce configuration errors by minimizing the number of items that need to be input. These items include hardware specifications, system scale (selection of large/medium/small scale) and the number of connections, and they are used to enable the automatic configuration of an appropriate environment. WebAdmin can be used immediately after installation. The parameter values that are automatically configured have been improved in Symfoware Server V10, so the time spent customizing the operating environment will be reduced. (Figure 5-2).

- **Database construction support tool (WebDBtools)**
  
  This is a GUI tool that supports the database construction phase. It enables the simple performance of definition operations (such as database definitions, table definitions or DSI definitions) and enables easy access to definition information. It also simplifies the reference and update of data stored in databases.
The use of WebAdmin and WebDBtools allows the whole setup process to be performed via a GUI (Figure 5-3).

**Application/database performance tuning tool (SQLTOOL)**

This tool checks beforehand that the developed program can run without any problems in the operating environment. To ensure optimal data processing during database operation, this tool interactively checks the appropriateness of the database's storage structure and whether SQL statements are running in the way they were designed. On the basis of these results, it can also implement database performance tuning and preemptively prevent problems after operation has started.

**Performance monitor (Operation)**

This is a tool that supports the operation phase by automatically collecting and storing daily performance and statistics information at regular intervals. It can drill down from weekly or monthly data and analyze what has occurred at any specific time. This enables monitoring for signs of problems, and may reduce the time spent dealing with performance problems if it is used to investigate their cause. It can also quickly identify SQL statements and applications for which processing has been delayed due to performance problems (Figure 5-4).
Specify the SQL statements that cause problems

Continual monitoring automatically detects SQL statements that take longer than a specified timeframe, and the access plan and sampling information are collected and stored.

This information makes it possible to specify SQL statements that require excessive periods of processing time and reduce the impact of the problem areas.

Examples of SQL statement table requiring extended processing time

<table>
<thead>
<tr>
<th>SQL statement</th>
<th>Performance data</th>
<th>Storage file</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdbpmsqllist</td>
<td></td>
<td></td>
</tr>
<tr>
<td>command</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5-4 Performance monitor
6. Conclusion

This white paper has introduced Symfoware Server's functions and technologies from the perspective of business continuity, high performance, and security. Not only does Symfoware Server V10 deliver even stronger mission-critical database functions, it also adds a number of reinforced functions. These include enhancements for working with applications development, including SQL/XML support and support for the most recent application interfaces.

The following table lists enhanced functions not introduced in this white paper:

<table>
<thead>
<tr>
<th>Application development</th>
<th>Supports SQL/XML.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Supports the most recent application interfaces:</td>
</tr>
<tr>
<td></td>
<td>- JDBC 4.0</td>
</tr>
<tr>
<td></td>
<td>- ODBC 3.5</td>
</tr>
<tr>
<td></td>
<td>- .NET Framework 4</td>
</tr>
<tr>
<td></td>
<td>Supports 8-byte binary digits in embedded C. Extended function for batch insertion of multiple lines of embedded SQL.</td>
</tr>
<tr>
<td></td>
<td>Supports value expressions in the VALUES clause of INSERT statements.</td>
</tr>
<tr>
<td></td>
<td>Supports 3-tiered model of the XA interface.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance</th>
<th>Supports specification of access model for individual SQL statements (ASSIST). Supports ROWNUM.</th>
</tr>
</thead>
</table>

| Operation | Allows simplified database construction. Allows output of rdexecsql command in table format. System tables were added. |

In a business environment faced with a rapid rate of change, Symfoware Server continues to strengthen its functionality. By continually improving the inventory of functions needed to realize a mission-critical database, as started in the era of mainframe computers, and by proactively adding new functions in response to new business demands, Symfoware Server will continue to evolve its capacity to support business continuity.