SymfoWARE Database

A Relational Database from Fujitsu, the leading IT company in the age of mega-transformation

Technical White paper
September 1999
Contents

Preface................................................................................................................................1

High Reliability, High Fault-resistance and High Availability ...... 4
  Maintaining the integrity of mass data ............................................................................. 4
  Highly reliable system operation ....................................................................................... 5

Improved Capabilities ............................................................................................ 11
  Assurance of multi-user access capability ...................................................................... 11
  Load balancing for smoothing response times .............................................................. 13
  Fast searching in large-scale databases .......................................................................... 14
  Fast creation of a database .............................................................................................. 14
  Use of hardware up to the limits of its capabilities ....................................................... 14

Programming Interface on the Client Side................................................ 16
  Application development environment ............................................................................ 16

Cluster Solution ......................................................................................................... 17
  Assurance of scalability and ensuring of availability .................................................... 17
  Improved system availability .......................................................................................... 21
  Flexible response to operational requirements of the system ........................................ 22

Parallel Data Warehouse Server ........................................................................ 25
  Fast searching in very large scale databases ................................................................. 25
  Fast creation and updating of a very-large scale database ............................................ 27

Conclusion...................................................................................................................... 30
Preface

Business activities nowadays surpass national boundaries. In this environment, corporations are no longer in competition only within their own industry, but are entering a time where they have to struggle for their survival whilst they can hardly see with who they are competing.

These dramatic changes in the business environment require business organizations to be ready at any time to take prompt and effective action. On the one hand, this makes the IT (Information Technology) environment critically important to these organizations; on the other hand, the Internet has revolutionized this IT environment. Enormous numbers of people are able to utilize at will, in the most convenient way, a wealth of diverse information from various locations. This defines a whole new business environment. The IT environment has surpassed intra- and inter-business activities and is now being used as an infrastructure for more general purposes of society. As a result, the number of users, the volume and variety of provided services, and the traffic and quantity of data to be handled are expanding at very significant rates. The volume and content of information is also rapidly changing in this changing environment. Accordingly, the expectations on and evaluation criteria for the database systems that manage information are changing considerably.

Because of more intense competition among business organizations in globally expanded markets, an increasing number of companies demand to be supplied with more and more information to give them the strategic means for better decision making. Because of the wide propagation of the Internet network, it is now easy to collect information from locations all over the world. Hard disks, which store information, are providing more and more storage capacity and are becoming available at ever-lower prices. Under these circumstances, the capacity of databases to manage information is also increasing at a very high rates. Are you, however, aware of potential problems that are the other side of the coin in the expansion in database capacities?
Do your measures allow sufficient time to maintain data integrity (i.e., the time needed to create backups) or to restore data if it has been corrupted? Upcoming database systems should ensure data capacities in the order of terabytes even for normal operation.

Increasingly, UNIX and PC systems are being used in mission-critical areas because of more widespread use of EC and EDI among business organizations, and because companies are employing supply-chain management to ensure the efficiency of business operations. In addition to mainframes, UNIX and PC systems are now, in many instances, directly connected to business activities. Databases in these mission-critical UNIX and PC systems must be as reliable and failure-resistant as possible. They should be able to recover promptly from system failure, as well as providing normal services in case of disk failures.

The number of network users is increasing to an amazing degree as the number of companies and organizations offering various services for consumers through the Internet increases. There are also dramatic increases in data traffic accessing database systems that provide information services. Database systems must, therefore, provide users with sufficient scalability, responsiveness, and throughput to handle increased amounts of data and traffic. In an age of network computing, information is more likely to be widely distributed, to increase, and to become more diversified. For companies, it is becoming more and more important to strategically utilize this diverse information so they maintain their competitive edge. Therefore, companies need database systems that are:

- able to cope with expanded data capacity, and
- capable of processing data on a large scale and at high speed.

The following ten requirements for real-world database systems are based on the conditions described above.

[With respect to data capacity]

1. Sufficient run time must be provided for maintaining the integrity of mass data (i.e., the time needed to create a backup).
2. Sufficient time must be provided for creating and migrating mass data.
3. Sufficient time must be provided for restoring mass data.

[Reliability]

4. A database system should have the capability to restore data regardless of which part of the database (including the database management file) has
been corrupted, without having to reconstruct the database.

(5) A database system should have the capability to determine what data has been corrupted and to isolate all compromised storage areas in such emergencies as power outages.

(6) A database system should be able to resume operation promptly in case of an unexpected system failure.

[With respect to users]

(7) A database system should provide for sophisticated access capabilities from multiple clients.

(8) A database system should assure stable response to multiple clients.

(9) A database system should provide 24-hour services to cater for diverse user requirements.

[Capability for growth]

(10) A database system should provide sufficient scalability to accommodate an increasing number of users.

Any new database systems should satisfy these ten requirements. SymfoWARE is a data warehouse server supporting an Internet OLTP environment, which requires highly capable and reliable servers, as well as being capable of facilitating prompt decisions. It provides powerful support for establishing optimal database systems.

This document outlines the different technologies implemented in SymfoWARE which correspond to the ten requirements. It covers the following categories:

- High reliability, high fault-resistance, and high availability
- Improved Capabilities
- Cluster Solution
- Parallel data warehouse server
**High Reliability, High Fault-resistance and High Availability**

Database systems play an indispensable role in the activities of business organizations and society and are used in principal, mission-critical business areas. Such systems have to be highly reliable, highly resilient and constantly available.

*SymfoWARE* achieves the required reliability, fault-resistance and availability for database systems by meeting the following objectives:

- Maintaining the integrity of mass data
- Highly reliable operation
- System without downtime

**Maintaining the integrity of mass data**

With the increasing diversity of user needs, an enormously large quantity of data is being produced in daily business activities. As data increases, the time taken to maintain data integrity increases proportionally. This may adversely affect daily online business and should not be neglected when selecting a database system. *SymfoWARE* supports the operations required to maintain the integrity of mass data, thus ensuring database operations.

**Partitioning**

Starting with Version 1, *SymfoWARE* is equipped with the necessary technology for dividing tables or indexes (partitioning). Partitioning makes it possible to localize the effect of possible disk failures. All operations for maintaining data integrity, such as creating a backup, recovering, and optimizing data, can be performed in a single partition separately from and in parallel with other partitions.

With the increasing capacity of database systems, other DBMS vendors have also adopted this technology because they recognize the need for partitioning. Partitioning is not essential for conventional databases that contain smaller quantities of data.
**Direct path**

*SymfoWARE* uses a unique access method (direct path) in order to quickly create, reorganize, or recover a database storing mass data. The direct path access method consists of directly reading from/writing to a database based on its physical format. This makes it possible to take full advantage of the I/O capabilities of the system.

**Pipeline control**

*SymfoWARE* executes internal processes separately (i.e., reading from/writing to a database, or conversion), and uses pipeline control to control the flow of data among processes (i.e., creating, reorganizing, making a backup of, or recovering a database). Executing internal processes in parallel speeds up processing for a database that contains mass data.

**Highly reliable system operation**

Database systems to be used in mission-critical business activities have to ensure that data can be restored in case faults occur. To ensure the ability to restore data, database systems have to be capable of:

- Managing database elements separately according to their aims (preventive)
- Isolating possibly corrupted areas (localization)
- Recovering the entire system to its normal state by repairing only the possibly corrupted areas (localized restoration)

*SymfoWARE* uses a sophisticated data-restoration technology to ensure fast and reliable operation of the database system so as to meet the requirements.
described above.

- The following aspects are discussed below:
  - Sophisticated recovery function
  - Step-wise resumption of jobs
  - Measures to counter emergencies, such as momentary power interruptions and disk failures
  - Improvements in operating ratio

**Sophisticated recovery function**

Database systems from other vendors provide no other way to recover from faults in the database management file or log file than reconstruction of the entire database.

*SymfoWARE* allows the database management file, the log file, and the storage area of the database to be placed on separate disks. This makes it less likely that double faults will occur in the database system. If a fault does occur in the database management file or the log file, not to mention the storage area of the database, data can be restored quickly by localizing the fault area. This fast restoration of data is achieved by using the following technologies for high reliability. These technologies are unique to *SymfoWARE*.

**Reverse Creation**

The database management file is the crucial file in controlling the handling of data in the database. In database systems from other vendors, the entire database needs to be reconstructed if a failure occurs in the database management file. Since the time required for entire reconstruction increases in proportion to the size of the database; in actual operation this may make it impractical for a large-scale database system.

*SymfoWARE* provides a function to restore the database management file from the database in case of a fault. This process is called Reverse Creation and enables the database system to resume work after a short time without needing to reconstruct the entire database.

**Reference:**

Time required to resume work after a corruption of the database management file was detected (based on actually measured results):

- Database system from other vendor: 4 days
- *SymfoWARE*: 2 hours
Adjusted recovery

SymfoWARE incorporates a sophisticated transaction management technology that quickly restores data by means of the adjusted recovery function in case the (temporary) log file that records the transaction history fails. Database systems from other vendors that do not provide this function need to restore the entire database from backup if the log file or other resource for assuring transactions is damaged. SymfoWARE is able to assure the transactions of the entire database very quickly, by using the archive log in parallel with the temporary log. Accordingly, if a failure in the temporary log occurs, SymfoWARE is able to quickly restore the data to secure data operations that have been performed up until the time of the failure.

Step-wise resumption of jobs

SymfoWARE is capable of restoring the system on a step-wise basis. It first restores the resources used for the job with the highest priority, so that the job in question can be resumed even before all data in the system has been restored.

Incremental Recovery

In combination with Reverse Creation, which restores the database management file from the database, Incremental Recovery makes it possible to reduce the time required for restoration even more. Incremental Recovery refers to the function of restoring the data in files that are necessary for resuming the job with the highest priority before restoring other data. Other data in the management file, necessary for other jobs, is restored while the job with the highest priority continues execution.

Counter measures for emergencies such as momentary power interruptions and disk failures

As the quantity of data and data traffic increases, the likelihood of system failures increases as well. SymfoWARE supports prevention, detection,
localization of, and restoration from, failures related to disk damage.

**Techniques to exactly determine which data has been corrupted and to localize the corrupted area**

Data can be corrupted due to a momentary power interruption or a disk fault. When **SymfoWARE** detects data corruption, it sends out a message to indicate the location of the corrupted data and prohibits access to the corrupted area so as to prevent secondary damage. This function enables continued access to areas other than the corrupted area so that operation can continue.

**Technique for isolating corrupted data**

When data corruption occurs, the corrupted data should be isolated from other data so that the latter is not affected by the former. **SymfoWARE** does this by prohibiting access to corrupted data in order to prevent secondary damage that may be caused by the corrupted data.

**Data parity check**

Data defects on the disk can occur because of sporadic disk faults. **SymfoWARE** performs a parity check on data when reading from the disk or before writing to the disk. If defective data is detected, **SymfoWARE** prohibits access to this data so as to avoid secondary corruption of other data.

**Improvement in operating ratio**

System availability needs to be as high as possible for principal business activities. **SymfoWARE** meets the requirement of high availability for mission-critical, principal business activities by providing Hot Standby. The Hot Standby function enables prompt resumption of operations in case of a failure.

**Hot Standby (the technique of transferring jobs)**

Exceptional events in hardware or application software often lead to significant adverse effects or damage to jobs. To minimize loss from such damages, a technique for minimizing downtime is necessary. **SymfoWARE** minimizes system-down time by using Hot Standby. Through Hot Standby, the database environment of a node can be duplicated in another node before a job is executed. This means the latter node can take over job execution from the former node, if necessary, and operation resumed promptly by a client executing **SymfoWARE** reconnection processing.
Security

*SymfoWARE* supports privileges in compliance with SQL2 of ISO/SQL. Moreover, *SymfoWARE* allows the user to set the privilege to:

- Create, alter, or drop database definitions, and
- Allocate space for tables or indexes.

The user can set these authorities via GRANT and REVOKE statements to ensure database security.

Non-interruptible Job Policy

*SymfoWARE* makes it possible to perform maintenance on a database without interrupting job execution.

Online Backup of the database

With *SymfoWARE*, a database can be subject to maintenance while it is running online. This feature is called Online Backup and makes it unnecessary to schedule maintenance for late at night or during holidays, because maintenance can be executed during the normal times for job operation.
**Database maintenance at runtime**

*SymfoWARE* does not require operations to stop if disk failure occurs. *SymfoWARE* provides a function for dynamic runtime separation of a database with a disk failure. This function makes it possible to recover the problem database while databases not affected by the failure continue to perform online job operation.

**Online/offline status of a database**

During database operation, tables or indexes may need to be reorganized. *SymfoWARE* provides a function for putting offline one or more partitions of a table or an index that has been split by the partitioning function, thus enabling a partial reorganization of tables and indexes. This allows system maintenance to be performed without interrupting normal operations.

**Expansion of the capacity of a database**

If a database in use runs out of space for tables and indexes, operation normally needs to be stopped to expand the capacity of the database. *SymfoWARE* supports functions for both static and dynamic expansion of database capacity. The static expansion function monitors the used capacity of a database and expands the total capacity when the used capacity reaches a certain value. *SymfoWARE* also makes it possible to automatically expand the capacity of a database as required, so that a database cannot run out of space during operation. This function enables the capacity of a database to be expanded without interrupting online job operation. Normal responses for online operation are maintained while expansion occurs.
**Improved Capabilities**

With increased amounts of data, increased numbers of transactions, and increased numbers of users, it has become much harder to satisfy processing capability requirements. *SymfoWARE* has adopted various improving technologies to meet these requirements.

The following aspects are discussed below:

- Assurance of multi-user access capability
- Load control for equalizing response times
- Fast search in large-scale databases
- Fast database creation
- Use of hardware up to the limits of its capabilities

**Assurance of multi-user access capability**

*SymfoWARE* incorporates advanced technologies for ensuring that response time does not deteriorate when the number of database users increases. Accordingly, it enables information systems for Internet use to accommodate an increased number of users.

Reference:
You can confirm the TPC-C capabilities of *SymfoWARE* by accessing the following URLs:

http://www.fujitsu.co.jp/hypertext/softinfo/product/db/SYMFO

http://www.tpc.org/results/tpc_c.results.page.html

**Staging control**

*SymfoWARE* organizes database processes into units, controls CPUs by means of a unique dispatch function (staging control), and pipelines the processing of each unit.

Staging control takes advantage of the characteristics of multi-threading and minimizes dispatch costs to the OS incurred in switching the run unit. This technology enables *SymfoWARE* to make the most of the SMP configuration. Accordingly, system response time remains at a practical level even if the number of database users increases.
Partitioning

When multiple transactions access a table, there are attempts at simultaneous input/output accesses to the disk and mutually exclusive transactions have to queue. This lowers system responses during processing. As the number of transactions increases, these adverse effects become greater.

SymfoWARE incorporates a partitioning technology that avoids degrading system response during such operations. Dividing tables and indexes into partitions makes the operating unit much smaller, and enables multiple partitions to be processed in parallel.

Partitioned tables and indexes can be processed internally on a partition-by-partition basis. This makes it easier to operate multiple transactions simultaneously as attempted input/output disk accesses are distributed over the partitions. Queuing of mutually exclusive transactions is also less likely to occur because the unit of exclusivity is divided as well.

Buffer control

When processing a large quantity of data in multiple transactions, the efficiency of input/output between the database buffer and the disk has an important effect on system response. SymfoWARE allocates database buffers according to the partition of tables or indexes. This avoids conflicts among database buffers, raises the residency rate for buffers (buffer control), and
SymfoWARE reduces the time required for disk input/output. In disk input/output, multiple sets of data are processed in bulk to raise input/output efficiency. Because SymfoWARE has adopted a tunable technology, there are fewer limits on the performance of the database.

**Concurrent execution of transactions**

SymfoWARE allows the user to use three units of exclusivity: partition, page, and record. Unnecessary exclusive use of queues can be prevented because the user can set the exclusivity range in advance.

It also allows the user to set the level of independence of a transaction, thereby improving the probability of concurrent transaction execution.

**Detection of deadlock**

SymfoWARE automatically detects deadlock and cancels the responsible transactions.

It also provides a command to monitor for deadlock. This makes it possible to check for deadlock even if the applications involved in possible deadlocks are unknown.

**Load balancing for smoothing response times**

With the improved performance of single CPUs and the increased number of CPUs in SMP configuration, the absolute level of hardware performance has risen significantly. If this improved performance is not used efficiently, not only will the cost performance of the system be low, but the required time for processing will also become difficult to estimate. When executing OLTP jobs, it is important to avoid peaks as well as troughs in system response.

SymfoWARE manages tasks so that workload is distributed evenly among the CPUs to ensure that system response during processing increases in proportion to the number of available CPUs.

**Dynamic queue control**

When a new transaction is initiated, SymfoWARE allocates the processing for this transaction to a CPU with low workload so as to distribute the workload evenly among available CPUs. However, with this approach alone, differences in workload would still occur among CPUs over a period, because executed transactions generally end at different points in time. As soon as this occurs, SymfoWARE automatically redistributes the transactions over the CPUs (dynamic queue control) to continually ensure roughly equal workloads, thus providing stable system response times.
Fast searching in large-scale databases

The time required to search large quantities of data increases with the amount of data. SymfoWARE provides high-speed search capabilities by running data analysis on mass data, as well as running principal jobs that use large databases as batch jobs overnight in the form of parallel SQL queries. This function improves responses for batch processing as well as for the analysis of mass data, enabling high system performance.

Parallel queries

SymfoWARE is the first software worldwide that provides partitioning technology for dividing and managing large-scale databases. Parallel queries make it possible to search in and read from database partitions in parallel via a single SQL statement.

Fast creation of a database

The amount of data processed by databases process is increasing steadily. In many cases database creation may take several days. SymfoWARE is capable of storing data in the order of gigabytes into a database in a very short time by means of a utility for fast database creation. The fast creation utility executes all processes necessary to create a database (i.e., reading input data, performing format conversions, and writing to the database) on multiple data sets in parallel. It simultaneously creates indexes defined in tables, as well as those tables themselves. Therefore, the total processing time for creating multiple indexes does not depend on the number of indexes, but is fixed, because the utility creates them in parallel.

Use of hardware up to the limits of its capabilities

To handle large amounts of data in multiple transactions, hardware must provide a high level of capabilities. However, if these capabilities are not well utilized, demands on the system cannot be met and its cost performance will be low. SymfoWARE makes the most of the available hardware capabilities, ensuring that demands on the system are met and its cost efficiency is high.

Handling 64-bit logical address spaces

When handling large amounts of data, the system attempts to load data to memory as far as possible in order to improve processing performance. As the quantity of data increases, the available memory capacity needs to increase as well. SymfoWARE takes full advantage of the 64-bit logical address space to raise the residency rate of memory. It can also control the residency rate according to access frequency in terms of physical units (DSI unit) of tables.
Supporting large RAW devices

SymfoWARE is capable of allocating database areas to RAW devices exceeding 2 gigabytes so as to use large-capacity disks efficiently for database areas. This ensures fast access as well as database safety because it is no longer necessary to create database areas only in an ordinary file system.
Programming Interface on the Client Side

Application development environment
The following PC tools or application programs on the client can remotely access SymfoWARE databases on the server:
- Embedded SQL application programs
- Spreadsheet programs, such as Excel
- DB tools such as Visual Basic and Access
The following link methods are recommended for accessing SymfoWARE databases.

Embedded SQL link [Esql-C, Esql-COBOL]
Fast access to SymfoWARE databases from application programs written in C or COBOL is possible by using embedded SQL statements for data operations. Such application programs can be developed with a SymfoWARE Programmer’s Kit.

ODBC link [ODBC driver]
Access to SymfoWARE databases can be performed via PC tools that are compatible with the ODBC interface, such as Excel, Access, and Visual Basic. SymfoWARE provides the following ODBC drivers:

SymfoWARE ODOS
This is a dedicated SymfoWARE ODBC driver for ensuring the efficient and economic use of resources (as far as server memory is concerned). This driver can use all SymfoWARE functions and performs better than conventional ODBC drivers, as follows:
- It uses only one-third the amount of server memory used by a conventional driver.
- Its response time is 20-40% faster than that of conventional drivers.

Conventional ODBC driver (RDA-SV)
This driver can access databases in global servers and other Fujitsu RDB products (RDB6000 and SymfoWARE7000).

JAVA link [JDBC driver]
This is an interface for accessing databases via Java. SymfoWARE databases can be accessed via the JDBC drivers of Java application programs created based on JDBC API.
Cluster Solution
By combining multiple computers via high-speed interconnection, cluster systems achieve high availability and extend available processing capabilities. Constructing a cluster system enables large quantities of data from multiple transactions to be processed. This fulfills highly complex user requirements. The resulting redundant configuration of hardware and software splits and distributes risks and workload.

Assurance of scalability and ensuring of availability
The Load Share configuration of SymfoWARE can be set up according to the size of operations. If operations expand and the system is in danger of running out of capacity, capacity can be expanded nearly linearly by adding new computers. Hot Standby in combination with Load Share makes it possible to maintain system operation in case of unexpected failures.

SymfoWARE ensures minimum interruption to operations that is caused by exceptional events in hardware or software. It does this by means of various functions implemented with state-of-the-art technology. Consequently, it enables non-stop operation for 24 hours per day for 365 days per year. This achievement is based on a Load Share solution.

Reference:
80% of the causes of unexpected system downs can be attributed to failures in the hardware, disks, applications, and networks.

Load Share
A cluster system configuration enables multiple nodes to run in parallel. SymfoWARE has adopted the Shared Nothing method as parallel transaction technology. In this method, tables and indexes are divided, and managing nodes are determined according to the division unit. Application programs have direct access to data under a managing node, and access to data under other nodes via the automatic routing system. Accordingly, application programs can access data without needing to know where tables or indexes are stored.

Nodes are connected by high-speed interconnection and parallel processing is implemented. This assures a high total throughput.

If a database system is running out of capacity because of increases in the number of users or increases in the workload, nodes can be added without
interrupting system operation. This enables the processing capability of the system to be improved without any adverse effect on the existing database system.

Data maintenance operations can be executed according to the division of tables and indexes that have been distributed over multiple nodes. There are no conflicts of database buffers among nodes because data are handled by managing nodes in a remote access system. Applying this technology, the Load Share function of SymfoWARE achieves nearly linear increases in processing capabilities.

**Scalable partitioning**

SymfoWARE enables tables and indexes to be partitioned, and distributes the partitions over multiple cluster nodes. Partitioning and distributing tables and indexes allows the operation of individual nodes to be kept largely independent.

**Fast interconnect control**

The nodes that form the cluster are connected by a fast interconnect system (AP-Net). AP-Net enables high-speed communication between nodes and ensures high total throughput in combination with parallel processing. The communication interface used complies with VI Architecture (Virtual Interface Architecture), an industry standard for cluster systems.

Reference:

Compaq, Intel, and Microsoft did the original development work for AI Architecture; eventually more than 100 companies joined in formulating the
standard specifications.

**2-Phase Commit control**
In the load share operation of SymfoWARE, tables or indexes are not shared by multiple nodes but are managed by partition units. Accordingly, a transaction that accesses multiple (partitioned) tables may be processed / operated across nodes. Because this type of transaction is controlled automatically by 2-Phase Commit control, the application program need not be aware of the transaction.

**Exclusivity control between nodes**
SymfoWARE has adopted the *Shared Nothing* system to process transactions in parallel. This system executes exclusivity processing per node that manages the data. Accordingly, there is no need to be aware of possible data accesses across nodes related to exclusivity processing. This system enables SymfoWARE to automatically detect deadlock across nodes and cancel the responsible transactions. When the applications involved in deadlock are unknown, SymfoWARE enables the user to identify the applications as well as the nodes on which the applications run by means of the check for deadlock command.

**Responsiveness to system expansion**
Because tables and indexes are not shared by nodes but are managed on each node (*Shared Nothing*), the lock overhead of nodes is minimized and it becomes possible to provide performance scalable with system expansions. The database buffers are also used highly effectively because data is handled by the managing nodes in a remote access system. If the system is running out of capacity because of expanded workload, or increases in the number of transactions or users, the system can be expanded by adding more computers.

**Degeneration**
With SymfoWARE, the system can continue to operate on a scaled-down basis if a node goes down for some reason. The system can continue to use tables and indexes other than those managed by the node that went down because SymfoWARE manages partitioned tables and indexes according to the partition. Accordingly, if a node goes down, the system does not go down.

**Shortening the time for degeneration**
With SymfoWARE, the system can continue to operate on a scaled-down basis if a node goes down. However, the degraded time should be kept short and
normal system performance restored as soon as possible. SymfoWARE introduces Hot Standby and Flash Treatment Recovery to keep the time of degeneration short.

**Hot Standby**

The Hot Standby operation of SymfoWARE ensures fast node switching, so that the operation of a malfunctioning node within a cluster system is swiftly taken over by another node. This assures a quick resumption of system operation after node down, and leads to high system availability. This is a key factor in the age of network computing. Mutual monitoring and switching of nodes is handled by cluster control: SymfoWARE uses cluster control for mutual monitoring of nodes, which ensures consistent switching among multiple system nodes.

**Flash Treatment Recovery**

As the number of nodes increases, the processing capacity of the system increases on a scalable basis. However, the likelihood of node-downs increases as well; therefore, rapid switching from one node to another needs to be performed if a node goes down. SymfoWARE provides the Flash Treatment Recovery function to meet this requirement. The function sends an updating log, at optimized intervals, from the running node to the standby node via the high-speed interconnection in the cluster environment. This means that there is always an updating log ready if nodes need to be switched unexpectedly. Consequently, it is unnecessary to read in the log. Furthermore, SymfoWARE applies Delayed Transaction Recovery to further shorten switching time during recovery. As the tables and indexes of transactions in progress are recovered in the background in parallel with currently running jobs, an almost immediate resumption of operations is possible.

**Parallel down recovery**

SymfoWARE also applies parallel technology in the recovery process. The 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 partition number of tables and indexes. The 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, recovery time can be dramatically shortened.
**Improved system availability**

*SymfoWARE* satisfies the requirement for improved system availability of large-scale database systems, such as data warehouses, by providing Hot Standby.

The Hot Standby system ensures prompt resumption of system operation after a failure.

**Hot Standby**

The Hot Standby function of *SymfoWARE* allows a node with a failure to switch to another node. It also employs various technologies to speed up the switching process.

**Pre-opening (opening the database with a standby node)**

The Hot Standby option of *SymfoWARE* provides the standby node with the same runtime environment as the active node. This pre-opening function makes it unnecessary to set up the runtime environment in the standby node when switching from the active node to the standby node. Therefore, system operations are resumed faster.

**Cache recovery**

*SymfoWARE* makes it possible to define a buffer pool in the standby node for reading in the database in the same way as the running node. When switching from the running node to the standby node, this large capacity buffer pool already exists. The input/output efficiency of the database is much higher when this buffer pool is used than when the default buffer pool is used, because the on-buffer effect is much greater in this case. This user-defined buffer pool also ensures that operations resume promptly.

**Connection of application programs in the standby node**

With *SymfoWARE*, the standby node can also activate application programs prior to switching operations. Application programs on the standby node can be put into a waiting state by issuing a CONNECT statement. As a result, job operation can be promptly resumed without having to activate the application programs by simply changing the destination of data to be transmitted.

**Automatic withdrawal of database access environment**

When executing applications in a client-server model, users may often switch off the power, e.g., in case of a terminal failure. When a user cuts off power to a terminal while an access to the database is still in progress, the database
access environment remains on the server. This may lead to an exhaustion of memory space or to an unnecessary occupation of database resources.

Other Database Management Systems use the KEEPALIVE function as a countermeasure against this problem; in other words, by withdrawing the database access environment after a specified time has passed. Use of this function has two drawbacks: it leads to increased load on communication lines, and database resources are not released until the specified time has passed. SymfoWARE automatically withdraws the database access environment as soon as the terminal is switched off and restarted.

Accordingly, if a terminal is, for some reason, switched off while an access to the database is still in process, memory space and database resources are not used unnecessarily.

This technology is used in various operating modes, such as in Hot Standby operation, the 3-hierarchy model, and Hot Standby operation in the 3-hierarchy model. SymfoWARE helps to achieve stable operation, even in case of abnormal termination of applications or node switching, by using Hot Standby for job operation and therefore not using memory space or database resources unnecessarily.

**Flexible response to operational requirements of the system**

SymfoWARE enables the Load Share operation to create a highly expandable database system as well as supporting the high availability of the Hot Standby operation.

This makes it possible to respond flexibly to the various operational requirements of modern database systems.

**Various forms of operation depending on system/cost requirements**

SymfoWARE makes it possible to choose among N:1 standby operation, mutual standby operation, and 1:1 standby operation depending on system/cost requirements. This is achieved by the Hot Standby operation, or the combination of the Load Share operation and the Hot Standby operation.

These forms of operation are described below.

**[N:1 standby]**

N:1 standby consists of keeping one standby node for several active nodes. This form of operation has the advantage of assuring high processing capabilities in the event of a node going down, while maintaining low standby costs. It is, therefore, suitable for large-scale database systems, such as data warehouses that have to support queries over a large amount of data.
[Mutual standby]

Mutual standby systems consist of multiple nodes that act as standby nodes for each other. This form of operation has the advantage of low costs for standby as well as providing an effective use of nodes. It is, therefore, suitable for database systems that require high reliability and low costs.

[1:1 standby]

1:1 standby operation consists of keeping one standby node for each active node, i.e., the same number of standby nodes as active nodes. This form of operation provides more reliability than N:1 or the mutual standby operation because processing capabilities are maintained to a large degree, even if multiple active nodes go down successively. It is, therefore, suitable for mission-critical database systems in principal operation areas (OLTP systems).
1:1 standby model
**Parallel Data Warehouse Server**

In this age of networked computing, information tends to be distributed over distant locations, to increase, and to be diversified. For companies to keep their competitive edge, they need to use a variety of strategic information. Making SymfoWARE Parallel Server the heart of the central data warehouse is the best way to ensure that a large amount of diversified data can be utilized strategically.

This product employs high-speed-execution technology, such as full-phase parallel execution, which helps to achieve the highest possible search performance. It can provide a performance of up to 100 times higher than that of any of our older database systems in an environment of the same scale.

**Fast searching in very large scale databases**

The most effective way of retrieving information promptly from large amounts of data according to various query conditions, is to process data in parallel on multiple CPUs in such a way that data can be read from multiple disks simultaneously.

SymfoWARE Parallel Server employs excellent and unique parallel technology for fast searching.

**Full-phase Parallel Mechanism**

SymfoWARE Parallel Server uses the full-phase parallel mechanism to make the most of available hardware performance in order to speed up processing. The full-phase parallel mechanism executes the searching phases SCAN, JOIN, SUM, and SORT in parallel (processing phase parallelization technique). At the same time, it allocates data to successive phases according to the start of processing for each phase (phase-to-phase parallelization technique). This combination enables the required processing steps to be completed in a short time.

This mechanism also ensures that a CPU is handed from one process to another after it has processed a certain number data items, so as to ensure that no CPU stays fully occupied by only one particular process. This is to prevent delays because of individual processes. Moreover, this mechanism also ensures that the quantity of data that is to be transferred from one phase to the successive phase is controlled according to the processing load on the target for transfer. It can therefore be assured that the same number of data items are processed on both the original CPU and the target for transfer. In other words, this mechanism speeds up parallel execution by making effective use of all available CPUs.
**Dynamic Load Balancing Mechanism (Technique of Equalizing CPU Load)**

In parallel processing with multiple CPUs, the CPU that takes the longest processing time decides the processing time of the entire system. To address this problem, the quantity of data to be handled by each CPU should be automatically controlled so as to ensure that the workload is equalized. This is implemented by the automatic load balancing mechanism. SymfoWARE Parallel Server uses this mechanism to keep the amounts of data that are processed in all phases of parallel operation roughly equal. Moreover, this mechanism also keeps the amounts of data that are processed in successive phases equal by changing the destination phase for every transfer accordingly. It transfers data to the destination phases evenly. The transfer process is integrated with the process for determining the transfer destination, thereby reducing the processing cost per record. Data allocation imposes a heavy workload on CPUs; it is implemented, therefore, in phases in which only I/O tasks are performed. Accordingly, to help achieve a higher performance, I/O operations overlap with CPU use.

**Automatic Data Balancing Mechanism**

In high-speed retrieval processing, to be suitable for parallel processing data needs to be stored evenly. SymfoWARE Parallel Server has adopted an automatic data balancing mechanism for storing data evenly to disks. When reading data, this mechanism ensures that data evenly stored in multiple disks is read in parallel. This ensures that the required processing time for each of these read operations is roughly equal and makes it possible to read records in a shorter time that in the case of uneven storage of data on multiple disks.
This mechanism determines on what disk data should be stored according to the quantities of data already stored on the disk. This keeps the quantities of data on disks roughly equal, thereby assuring a high performance of the database system without the need for reorganization.

Fast creation and updating of a very-large scale database

*SymfoWARE Parallel Server* employs unique parallel processing mechanisms: retrieval operations are performed in parallel, the amounts of data to be processed are kept roughly equal, and the amounts of data to be stored are roughly equal as well. Moreover, the inserting, updating, and deleting of data is performed in parallel. This enables the use of very-large scale databases that could not be implemented before.

**Parallel Super-loader (Technique of Parallel Creating and Updating Database)**

When a database is created, the Super-loader reads from multiple input files (which store the source data to be input into the database) in parallel. A data format conversion process, executed by multiple CPUs, is applied to the read data (this converts the formats of source data created by applications or ISV products to the format of data stored in the database). The converted data is written in parallel to the database, which is distributed over multiple disks. The transfer of data between these two phases is controlled by the automatic data balancing mechanism in order to ensure that parallel processing is employed as effectively as possible. Data read from input files is immediately made subject to the data format conversion process and then written out to the database. The full-phase parallel mechanism thus makes the most of the potential performance of the system. Data is written to the database via a dedicated high-speed interface to make the most of the performance potential of the hardware and keep I/O time as short as possible.

If multiple indexes are added to a database, the creation processes are scheduled to be executed in parallel to make the time of creation as short as possible.

**Parallel creation of a database**

In addition to the direct path and pipeline control implemented in the old version of the Super-loader, parallel creation has now been implemented. This reduces processing time even more.

Data read from the input files is immediately converted and written to the database.

Data is read from multiple input files in parallel, and the format of this data is...
converted to the internal format of the database by multiple CPUs. The converted data is written in parallel to the database that is distributed across multiple disks. The transfer of data between these two phases is controlled by the automatic data balancing mechanism in order that parallel processing is employed effectively as possible. Data is written to the database via a dedicated high-speed interface to make the most of hardware performance and reduce I/O time as much as possible. The full-phase parallel mechanism therefore makes the most of potential system performance.

**Updating a database**

Conventional RDBMSs generally use SQL to update data. However, when creating a central data warehouse, it may often be necessary to update a database in the order of gigabytes at one time. The Parallel Super-loader makes it possible to update such large amounts of data at a high rate, by effective use of technology developed especially for Super-loader. The Parallel Super-loader reads input data and performs high-speed reading from the database in parallel, via a dedicated access method. It first sorts the input data and the data read from the database in parallel, then creates updating records based on the sorted data, and writes them into the database via a dedicated access method. In other words, the updating process has become much faster by employing such architectures as the automatic data balancing mechanism and the full-phase parallel mechanism.

**Prompt Maintenance and Restoration of Very-large scale Databases**

As databases increase in scale, the time taken to maintain and restore data also increases. To swiftly maintain and restore data in very-large scale databases, SymfoWARE has adopted the following parallel processing technologies for backup and recovery.

**Parallel backup**

SymfoWARE Parallel Server provides a parallel backup function. Backups are saved and distributed among multiple disks. This means that a backup of a database can be made quickly and safely. Parallel backup makes it possible to increase the number of disks on which backup data is saved, thereby improving backup performance in a scalable manner. As a result, a backup of ultra-large scale databases that were previously impossible to backup because of their size is now possible.
Reference:

Time (measured time) required to create a backup of a database with a capacity of 22 GB

In partition units (as performed by conventional systems): 50 min.

- 4-way parallel : 12 min.
- 8-way parallel : 8 min.
- 12-way parallel : 6 min.

**Parallel recovery**

*SymfoWARE Parallel Server* provides a parallel recovery function. This applies parallel processing to all recovery phases, including restoration from a backup, updating in accordance with the log, and restoration of control files. This permits the swift restoration of resources.

If backup data was saved using the parallel backup function, a database can be recovered by restoring only the corrupted areas, thereby making recovery time shorter and reducing adverse impact on job execution.
**Conclusion**

Business organizations consider it critically important to strategically utilize information in order to accommodate to, and strive in, the dramatically changing business environment. With this background, database systems are one of the essential elements in the IT environment. Ten key requirements for modern database systems are summarized below. These requirements are key criteria for selecting a database system in the IT environment.

[With respect to data capacity]

1. Sufficient operation time must be provided for maintaining the integrity of mass data (i.e., the time to create a backup).
2. Sufficient time must be provided for creating and migrating mass data.
3. Sufficient time must be provided for restoring mass data.

[Reliability]

4. A database system should have the capability to restore data regardless of which part of the database, including the database management file, has been corrupted, and without requiring the database to be reconstructed.
5. A database system should have the capability to determine what data has been corrupted and to isolate all compromised storage areas in such emergencies as power outages.
6. A database system should be able to resume operation promptly when an unexpected system failure occurs.

[With respect to users]

7. A database system should provide for sophisticated access capabilities from multiple clients.
8. A database system should assure stable response to multiple clients.
9. A database system should provide 24-hour services to cater for diverse user requirements.

[Capability for growth]

10. A database system should provide sufficient scalability to accommodate an increasing number of users.

This document outlines the technology that SymfoWARE provides to meet these requirements. It also shows that SymfoWARE completely meets important requirements for OLTP systems, in terms of performance, reliability,
availability, scalability, and operability. It also shows how SymfoWARE, as the heart of a central data warehouse, achieves the highest level of processing performance for databases worldwide.

As an industry leader, SymfoWARE provides advanced technology on a timely basis and supports the setup of a strategic IT environment for:

- creating data warehouses,
- business intelligence,
- Internet and intranet usage.
This document offers a basic description of the functions of SymfoWARE Server and SymfoWARE Parallel Server for the Japanese market.

This document is provided for informational purposes only, and the information herein is subject to change without notice.

Please report any errors herein to Fujitsu Limited.

Fujitsu Limited does not provide any warranties covering and specifically disclaims any liability in connection with this document.

- Microsoft, Windows, Windows NT, Visual Basic, and Visual C++ are registered trademarks of Microsoft Corporation in the United States and other countries.
- Access and Excel are products of Microsoft Corporation.
- Java and other service marks containing Java are trademarks of Sun Microsystems, Inc. and refer to products that use the Java brand technology of Sun Microsystems.
- Solaris is a trademark of Sun Microsystems in the USA.
- The other company names and product names indicated in this guide are trademarks or registered trademarks of the respective companies.