A Migration Methodology for 2-tier to 3-tier Architectured Client/Server Systems
A Migration Methodology for 2-tier to 3-tier Architectured Client/Server Systems

by

Yong-Jin Park

Department of Computer and Communications Engineering

POSTECH Graduate School of Information Technology

A thesis submitted to the faculty of POSTECH Graduate School of Information Technology in partial fulfillment of the requirements for the degree of Master of Engineering in the Department of Computer and Communications Engineering.

Pohang, Korea
December 19, 1997

Approved by

Major Advisor
ABSTRACT

Advances in computing and network technologies along with the changes in the way businesses operate influenced the introduction of client/server systems. These early client/server systems had 2-tier architecture where application logic and presentation logic were tightly coupled with specific database servers. These 2-tiered client/server systems were found to possess several major problems. First, these systems were not easily scalable as the number of users increased and thus were faced with severe performance degradations. Second, the systems could not easily cope with changes to business logic or environment. Third, since the applications were tightly coupled with specific databases, the systems could not be easily upgraded or changed without major efforts.

In order to solve the problems in 2-tier client/server systems, 3-tier client/server systems have been proposed. This thesis presents a simple but efficient methodology for migrating 2-tier to 3-tier client/server systems. Both 2-tier and 3-tier architectured client/server systems are thoroughly examined and compared. The client/server systems at POSCO have been analyzed and were determined to be mostly 2-tier architectured. As a proof of concept, the migration methodology has been applied to migrating 2-tier client/server systems at POSCO to 3-tier client/server systems.
3.1 Æ÷½ºÄÚ ................................. 26
3.2 ½Ã½ºÅÛ ................................. 28
   3.2.1 2°èÃþ ±¸Á¶¿¡¼­ ¹ß»ýµÇ´Â ¹®Á¦Á¡ ..................... 29
   3.2.2 ¼º´É ÀúÇÏ ¿äÀÎ ................................. 30
3.3 ¹Ìµé¿þ¾îÀÇ ............................ 30
3.4 ½Ã½ºÅÛ ¿î¿µÀÇ .......................... 31
   3.4.1 Àå¾Ö¹ß»ý À¯Çü ................................. 31
   3.4.2 ¾ÖÇø®ÄÉÀÌ¼Ç ¹èÆ÷¹æ¹ý ....................... 31
   3.4.3 ¹é¾÷ (Back up) ¹æ¹ý ......................... 32
   3.4.4 °³¹ß½Ã ¿ì¼±¼øÀ§ ......................... 32
4. 3°èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ ±¸Çö ¹æ¾È .......... 34
   4.1 Æ÷½ºÄÚ ½Ã½ºÅÛ ¾ÆÅ°ÅØÃ³ ±¸¼º .................. 34
   4.2 ¼­¹ö ½Ã½ºÅÛº° Àû¿ë¾÷¹« ......................... 36
   4.3 ¾ÖÇø®ÄÉÀ̼Ç ±¸¼º¹æ¾È ......................... 38
   4.4 3°èÃþ ±¸Á¶ ±¸Ãà½Ã °í·Á»çÇ× ..................... 39
   4.5 Åø ¼±Á¤ ±âÁØ ................................. 41
      4.5.1 Ŭ¶óÀÌ¾ðÆ® °³¹ßÅø ......................... 41
      4.5.2 ¹Ìµé¿þ¾î ¼±Á¤ ±âÁØ ......................... 42
5. 3°èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ Àû¿ë ............ 44
   5.1 Æ÷½ºÄÚ ½Ã½ºÅÛ ¾ÆÅ°ÅØÃ³ ±¸¼º .................. 44
   5.2 ¾ÖÇø®ÄÉÀ̼Ç ¾ÆÅ°ÅØÃ³ ±¸¼º ..................... 47
6. °á·Ð ¹× ÇâÈÄ °èȹ ................................ 49

© © © ................................. 51
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ŭ¶óÀ̾ðÆ®/¼­¹ö</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2°èÃþ</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>3°èÃþ</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>RPC</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>±¸Á¶ÀÇ</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>±¸Á¶ÀÇ</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>Object Management Architecture (OMA)</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>ÆÐ½Ì</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>±¸µé¿þ¾î</td>
<td>23</td>
</tr>
<tr>
<td>10</td>
<td>±×¸²</td>
<td>25</td>
</tr>
<tr>
<td>11</td>
<td>±×¸²</td>
<td>27</td>
</tr>
<tr>
<td>12</td>
<td>3°èÃþ</td>
<td>34</td>
</tr>
<tr>
<td>13</td>
<td>3°èÃþ</td>
<td>38</td>
</tr>
<tr>
<td>14</td>
<td>3°èÃþ</td>
<td>44</td>
</tr>
<tr>
<td>15</td>
<td>3°èÃþ</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>23</td>
<td>34</td>
</tr>
<tr>
<td>2.</td>
<td>23</td>
<td>45</td>
</tr>
<tr>
<td>3.</td>
<td>23</td>
<td>56</td>
</tr>
</tbody>
</table>
1

1.1

(Downsizing), (Rightsizing), (Upsizing) / (business processes)

(2-tier Architecture) / (User interface) / (Business Logic) / (database)

2 °èÃþ (transaction) / 

(transaction) / (business environment)

°º°í, °º°í (flexibility)

(legacy data sources) / Legacy

(2-tier Architecture)/ (User interface) / (Business Logic) / (database)

(2-tier Architecture) / (User interface) / (business environment) / (transaction) / (business processes) / (database) / (business environment) /

(legacy data sources) / (flexibility) / (flexibility) / (flexibility) / (flexibility) /
1.2 引入和概述

1.3.1 重要注释

1.3.2 需求

1.3.3 基本概念

1.3.4 技术实现

1.3.5 使用示例

1.3.6 总结

1.3.7 附录
2  

°ü·Ã ¿¬±¸ 3 °èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ ±¸ÃàÀ» ÀÇÇØ ÇÊ¿ä·Î ÇÏ´Â ÁÖ¿ä °ü·Ã ¿¬±¸·Î¼­ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ, ¹Ìµé¿þ¾î, ´ëÇ¥ÀûÀÎ °´Ã¼ÁöÇ⠹̵é¿þ¾îÀÎ ÄÚ¹Ù (CORBA), 3 °èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛÀÇ ±¸Ãà »ç·Ê¿¡ ´ëÇØ »ìÆìº»´Ù.

2.1  

°ü·Ã ¿¬±¸·Î¼­ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ 

°ü·Ã ¿¬±¸·Î¼­ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ¿¡ 2 ³ºìè® ¿å Àå´ÜÁ¡À» ³¬±³Çϰí, ÁÖ¿ä Àû¿ë¾÷¹«¿¡ ´ëÇØ¼­¼úÇϰڴÙ.

2.1.1  

°ü·Ã ¿¬±¸·Î¼­ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛÀÇ Á¤ÀÇ ¿¬°áµÈ ÁýÇÕü·Î¼­, ½Ã¸®¾ó ¶óÀÎ À¸·Î ¿¬°áµÈ Å͹̳¯ (terminal)·Î µ¿±â/ºñµ¿±â·Î Åë½ÅÇÏ´ø ±âÁ¸ÀÇ ¸ÞÀÎÇÁ·¹ÀÓ (mainframe)°ú´Â ´ëÁ¶µÇ´Â °è³äÀÌ´Ù [1]. ¿¹¸¦ µé¸é Local Area Network (LAN) À¹³ë Wide Area Network (WAN)¿¡¼­ ³×Æ®¿÷ ½Ã½ºÅÛ, ÀüÀÚ¸ÞÀÏ (Electronic Mail), Ŭ¶óÀ̾ðÆ®¹ö ½Ã½ºÅÛ µîÀÌ ¸ðµÎ ÀÌ¿¡ ¼ÓÇÑ´Ù°í ÇÒ ¼ö ÀÖ´Ù. ÀÌ·¯ÇÑ °ü·Ã ¿¬±¸·Î¼­ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛÀÇ ÃâÇö¹è°æ Àº ´ÙÀ½°ú °°´Ù [2, 3].

?  

 Hải (Hardware)¿¡¼­ ·®»ý»êÀÇ Ä¨À» ÀåÂøÇÑ ¼­¹ö°¡ ±âÁ¸ÀÇ ¸ÞÀÎÇÁ·¹ÀÓÀ̳ª ÁßÇüÄÄÇ»Åͺ¸´Ù °ª½Î°í °­·ÂÇÑ ¼º´ÉÀ» ¹ßÈÖÇϰí, ¸Þ¸ð¸® ĨÀÇ ¿ë·®Àº 3 ³â¸¶´Ù 4 ¹è·Î Áõ°¡ÇÏ´Â µî

SOFTWARE (Software)°ÅÌ¿


NETWORK (Network)°ÅÌ¿

Bridge, Router, Gateway, LAN, 10Mbps Ethernet, 100Mbps Fast Ethernet, 155Mbps Asynchronous Transfer Mode (ATM), 1000Mbps Gigabit Ethernet, LAN, WAN T1 1.54Mbps, T3 44.73Mbps, E1 2.04Mbps, E3 34.36Mbps, WAN Backbone.
2.1.2 Client/Server Computing

Client/Server computing involves a client request being processed by a server and the resulting data being returned to the client. [1]. This allows for a separation of user interface and data processing, which can be beneficial for scalability and security. [5].

![Client/Server Diagram]

**Client:**
- Manage display
- Interact with user
- Perform data validation
- Operate on retrieved data
- Generate data requests

**Server:**
- Process user data requests
- Send retrieved data
- Provide cocurrency control
- Maintain data integrity
- Manage transactions
- Provide storage

---

1. Client/Server Computing
2.1.3 ＵＰdater /ＵＩ UNTACT

 afterEach /each row of the user interface (User Interface) 更新 each row of presentation logic (Presentation logic), each row of business process (Business Process) 更新 each row of application logic (Application logic),
2.1.3.1 2-tier Architecture

2 °èÃþ Ŭ¶óÀ̾ðÆ®/¼­¹ö ¸ðµ¨Àº Ŭ¶óÀÌ¾ðÆ®¿¡´Â »ç¿ëÀÚ ÀÎÅÍÆäÀ̽ºÀÇ ÇÁ¸®Á¨Å×ÀÌ¼Ç ·ÎÁ÷ÀÌ, ¼­¹ö¿¡´Â µ¥ÀÌŸº£À̽º°¡ À§Ä¡ÇÏ´Â °ÍÀ¸·Î ¾ÖÇø®ÄÉÀÌ¼Ç ·ÎÁ÷ÀÇ À§Ä¡¿¡ µû¶ó ±×¸² 2 °°ÀÌ ¿ø°Ý µ¥ÀÌŸº£À̽º Á¢±Ù ¸ðµ¨ (Remote Database Access Model)°ú µ¥ÀÌŸº£À̽º ¼­¹ö ¸ðµ¨ (Database Server Model)·Î ³ª´­ ¼ö ÀÖ °Ú.

(a) Remote SQL
(b) Stored Procedure

2. 2°¡Áö 2°¡Áö 2°¡Áö
 Stored Procedure (SP) に よる 多機能性 が 期待 される。この ストアード プロセッサ（stored procedure）に関しては、Message passing と いわれ る（thin client）もの が ある。

2 つの サーバー 間 の スレッド は 1 つ の プロセッサ 上 で実行 される。どちら の サーバー 間 でも 2 つの サーバー 間 の インタフェース が 期待 される。DBMS は 1 つ の サーバー 上 で実行 される。この サーバー の インタフェース は 期待 される。

ビジネスプロセス（business processes）は ビジネス環境（business environment）である。PC は デスクトップ コンピュータ である。

RDBMS は SQL、ストアード プロセッサ、APIs、データソース（data sources）を 持っている。legacy data sources を 持っている。flexibility が 期待 される。
2.1.3.2 3-tier Architecture

LAN °ÍÀÌ, ¾÷¹«·®ÀÌ ÀûÀº ºÎ¼­´ÜÀ§ÀÇ Batch¾÷¹« (Åë°èºÐ¼®, ¹ßÇà µî) ¿Í ñÁÀÔÇÁ·¹ÀÓ¿¡¼­ Ŭ¶óÀ̾ðÆ®/¼­¹ö ÄÄÇ»ÆÃȯ°æÀ¸·ÎÀÇ ÀüȯÃʱ⿡ À§ÇèºÎ´ãÀ» ÃÖ ¼ÒÈ­Çϰí Àú·ÅÇÑ ºñ¿ëÀ¸·Î ½Ã½ºÅÛÀ» ±¸ÃàÇϰíÀÚ ÇÒ ¶§ Àû´çÇÏ´Ù.

2 °èÃþ ±¸Á¶ÀÇ ¹®Á¦Á¡À» ÇØ°áÇÏ´Â ¹æ¾ÈÀ¸·Î¼­ ÇöÀç ¸¹Àº °ü½É»ç·Î ´ëµÎµÇ°í ÀÖ´Â °ÍÀÌ 3 °èÃþ ±¸Á¶ÀÌ´Ù. ÀÌ ¹æ¹ýÀº µ¥ÀÌŸº£À̽º°¡ À§Ä¡ÇÑ ¼­¹ö ºÎºÐ°ú »ç¿ë ÀÚ°¡ ÁÖ·Î »ç¿ëÇϴ Ť¶óÀÌ¾ðÆ®¸¦ ¿ÏÀüÈ÷ ºÐ¸®Çɠ, ¾ÖÇø®ÄÉÀÌ¼Ç ·ÎÁ÷À» ¸ðµâÈ­ ÇÏ¿© Áß°£°èÃþ¿¡ º°µµ·Î À§Ä¡½ÃÅ´À¸·Î¼­ Á»´õ À¯¿¬Çϰí È®Àå °¡´ÉÇÑ ½Ã½ºÅÛ ±¸Çö ÀÌ °¡´ÉÇÏ´Ù. ±×·¡¼­ Áß°£°èÃþ¿¡ ¹Ìµé¿þ¾îÀÇ À¯¹«¿¡ µû¶ó 2 °¡Áö ¸ðµ¨·Î ±¸ºÐÇÒ ¼ö ÀÖ´Ù [±×¸² 3].

(a) 3°èÃþ ¾ðÆ®ºÎÅÍ ¾ÖÇø®ÄÉÀ̼Ǹ¸À» ºÐ¸®Çؼ­ ÀÛ¼ºÇÏ´Â °ÍÀ¸·Î ½Ç Ú¶óÀÌ¾ðÆ® ¾ÖÇø®ÄÉÀ̼Ç (a)

(b) 3°èÃþ ¾ðÆ®ºÎÅÍ ¾ÖÇø®ÄÉÀ̼Ç (b)

3. 3°èÃþ ¾ðÆ®ºÎÅÍ 2°¡Áö 2°¡Áö
ÁøÁ¤ÇÑ ÀǹÌÀÇ 3 °èÃþ ±¸Á¶ÀÇ Å¬¶óÀÌ¾ðÆ®¿þ¾î/¼­¹ö ±¸Á¶ÀÇ °³º° ¾ÖÇø®ÄÉÀÌ¼Ç ¼­¹ö°¡ºëÁö°í °³¹ßÀÚµéÀÌ ÇÁ·Î±×·¡¹Ö ¾ð¾î¿¡ Å©°Ô ±¸¾Ö ¹ÞÁö ¾Ê°í °³¹ßÀÌ °¡´ÉÇϰí, ¾ÖÇø®ÄÉÀÌ¼Ç ·ÎÁ÷À» ¸ðµâÈ­ÇÏ¿© ¹æ¹ýÀ¸·Î ÀϹݰü¸®¿Í °£¼ÒÈ­ÇÏ¿© À¯Áö°ü¸®ÀÇ ºñ¿ëÀ» ÁÙÀÏ ¼ö
<table>
<thead>
<tr>
<th>11</th>
<th>20</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Online Transaction</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>
2.2 ¹Ìµé¿þ¾î

¹Ìµé¿þ¾î´Â °³¹ß¿¡ µû¸¥ ºñ¿ëÀ» ÁÙÀÏ ¼ö ÀÖÀ» »Ó¸¸ ¾Æ´Ï¶ó °³¹ß±â°£À» ´ÜÃà Çϰí ȯ°æº¯È­¿¡ ½Å¼ÓÇϰí À¯¿¬ÇÏ°Ô ´ëó ÇÒ ¼ö ÀÖÀ¸¸ç, ´Ù¾çÇÑ Çϵå¿þ¾î ȯ°æ¿¡ ¼­µµ ¿ÀǽýºÅÛÀ» ½±°Ô ±¸Ãà °¡´ÉÇÏ´Ù´Â ÀÌÁ¡À» °¡Áö°í ÀÖ´Ù [2,9].

±×·¡¼­ º» Àý¿¡¼­´Â ¹Ìµé¿þ¾î ±â¼úÀÇ Æ¯Â¡°ú µ¿ÀÛ¿ø¸®¸¦ ¾Ë¾Æº¸°í ±â´Éº° ¹Ì µé¿þ¾î Á¾·ù¿¡ ´ëÇØ¼­ »ìÆì º»´Ù.

2.2.1 ¹Ìµé¿þ¾î Ư¡

ºÐ»êó¸® ȯ°æ¿¡¼­ ÇÊ¿¬ÀûÀÎ multi-vendor, multi-protocol ¿¡ ´ëÇØ µ¶¸³ ¼ºÀ» Á¦°øÇϰí, Çϵå¿þ¾î ¹× ³×Æ®¿÷ º¯È­³ª È®Àå½Ã¿¡ À¯¿¬ÇÏ°Ô ´ëóÇÒ ¼ö ÀÖ´Ù.

°¢°¢ÀÇ Çϵå¿þ¾î Ç÷§Æû¿¡¼­ ¿î¿µµÇ´Â ´Ù¾çÇÑ µ¥ÀÌŸº£À̽ºÀÇ ¸ðµç ±â´É À» Áö¿ø ÇÑ´Ù. »ç¿ëÀÚ´Â ÀÚ½ÅÀÌ °¢°¢ÀÇ Çϵå¿þ¾î Ç÷§ÆûÀÇ µ¥ÀÌŸº£À̽º ¸¦ »ç¿ëÇÏ´Â °Í°ú °°Àº Åõ¸í¼ºÀ» Á¦°øÇÑ´Ù.

»ç¿ëÀÚ Áß½ÉÀÇ ½Ã½ºÅÛ °³¹ßÀÇ º¯È­·Î ´Ù¾çÇÑ »ç¿ëÀÚ °³¹ßÅøÀ» Áö¿øÇÑ´Ù.

¸ðµç ±â¼ú¿¡ ´ëÇÑ Ç¥ÁØÀ» Áö¿øÇÏ¿© ´Ù¿î»çÀÌ¡À» °¡¼ÓÈ­ ÇÑ´Ù.

ÇöÀç ÄÄÇ»ÆÃȯ°æ¿¡ Áï½Ã Àû¿ë °¡´ÉÇÑ °³¹ß ȯ°æÀ» Á¦°øÇÏ¿© ÀüüÀûÀÎ ½Ã ½ºÅÛ ±¸Ãà ½Ã°£À» ÃÖ¼ÒÈ­ÇÏ°í µµÀÔ¿¡ µû¸¥ ºñ¿ëÀ» Àý°¨ÇÑ´Ù.

¹Ìµé¿þ¾î´Â À̱âÁ¾ÀÇ º¹ÇÕ¼­¹ö°¡ Æ÷ÇÔµÈ ¸ÖƼ¹ê´õÀÇ Å¬¶óÀÌ¾ðÆ® ¾ÖÇø®ÄÉÀ̼ÇÀ¸·ÎºÎÅÍ µ¥ÀÌŸº£À̽º¿¡ ´ëÇÑ ºü¸£°í Åõ¸íÇÑ ¾×¼¼ ½º°¡ ¿ä±¸µÇ´Â ȯ°æ ¿¡¼­ ÇÊ¿äÇÏ´Ù.
2.2.2 Remote Procedure Call (RPC)

Remote Procedure Call (RPC) [2] enables communication between a client and a server, where the client invokes a method on the server through a network connection. The client sends a request to the server, and the server processes the request and sends a reply back to the client.

RPC (Remote Procedure Call) is a protocol where a client application can call a procedure on a server without knowing the location of the server. This is achieved through a network connection, allowing for distributed computing.

The diagram illustrates the communication flow between the client and the server. The client initiates the request, and the server responds with a reply. The middleware acts as a bridge between the client and the server, handling the network communication and request processing.

For more details on the RPC mechanism, please refer to Section 4.
RPC

(메시지 전송) 

[2] 메시지 전송 (Stored and Forward) 

RPC, 메시지 전송은 원격 서버로 메시지를 보낼 때, 메시지를 중계 서버에 저장한 후 원격 서버로 보내는 방식입니다. 이는 RPC의 응답 시간을 줄일 수 있는 방식입니다.

RPC 요청은 원격 서버의 메시지 관리자를 통해 중계 서버로 보내집니다. 중계 서버는 메시지를 대기한 후 원격 서버의 메시지 관리자로 보냅니다.

 decency 5

RPC 3
1. ハードウェアの Manager は、ネットワークの分岐の管理を担当します。このようなネットワークの分岐は、ネットワークのパフォーマンスを向上させ、リソースを最適に利用するのに役立ちます。2. タイムリレーション関係は、RPC サービスの要求と応答のタイミングを制御します。3. シーケンシングは、RPC サービスの要求が、固定の順序で処理されることを示しています。

<table>
<thead>
<tr>
<th>標準</th>
<th>詳細</th>
<th>RPC状態</th>
</tr>
</thead>
<tbody>
<tr>
<td>ハードウェア</td>
<td>Manager</td>
<td>ハードウェアの Manager</td>
</tr>
<tr>
<td>タイムリレーション関係</td>
<td></td>
<td></td>
</tr>
<tr>
<td>シーケンシング</td>
<td>No fixed sequence</td>
<td></td>
</tr>
<tr>
<td>Style</td>
<td>Queued</td>
<td>Call-Return</td>
</tr>
<tr>
<td>Load balancing</td>
<td>FIFO</td>
<td>TP Monitor</td>
</tr>
<tr>
<td>Message filtering</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

2. ハードウェアの Manager の RPC サービス状態 [2]。
2.2.3 ¹Ìµé¿þ¾î

2.2.3 ¹Ìµé¿þ¾î

Distributed Computing Environment (DCE) [11]°°Àº ÅëÇÕ Åë½Å ¹Ìµé¿þ¾î¿Í µ¥ÀÌŸº£À̽º Àü¿ë ¹Ìµé¿þ¾î, ±×¸®°í °´Ã¼ ÁöÇâ °³³äÀ» Àû±Ø ¼ö¿ëÇÑ ºÐ»ê°´Ã¼ ¹Ìµé¿þ¾î µîÀ¸·Î ¹ßÀüÀ» °ÅµìÇÏ°Ô µÈ´Ù. ´ëÇ¥ ÀûÀÎ ºÐ»ê°´Ã¼ ½Ã½ºÅÛÀ¸·Î Object Management Group (OMG) ÀÇ Common Object request broker Architecture (CORBA) [12]Ã© ° ³® ° Component Object Model / Distributed COM (COM/DCOM) [13]À» µé ¼ö ÀÖ´Ù. ÀÌ·± °üÁ¡¿¡¼­ °¢ ºÐ¾ßº° ¹Ìµé¿þ¾î¿¡ ´ëÇÑ ±â´É°ú Á¾·ù¸¦ ¾Ë¾Æ º»´Ù [16].

? ±â´É Áß½ÉÀÇ ¹Ìµé¿þ¾î

1) Remote Procedure Call (RPC)

RPC ´Â Åë½Å ¹Ìµé¿þ¾î Áß °¡Àå ±¤¹üÀ§ÇÏ°Ô »ç¿ëµÇ¾ú°í Áö±Ýµµ ±âŸ ´Ù¸¥ ¹Ì µé¿þ¾î ½Ã½ºÅÛÀÇ ÇϺΠ±¸Á¶·Î ÀÌ¿ëµÇ´Â ½Ã½ºÅÛÀ¸·Î, »ç¿ëÀÚ¿¡°Ô ¿ø°ÝÁö¿¡ ÀÖ ´Â ÇÔ¼ö¸¦ È£ÃâÇÒ ¼ö ÀÖ´Â ¹æ¹ýÀ» Á¦°øÇÔÀ¸·Î¼­ ºÐ»êµÈ ȯ°æ¿¡¼­ÀÇ ÇÁ·Î±×·¡ ¹ÖÀ» ½±°Ô Á¦°øÇÏ´Â °ÍÀ» ¸ñÇ¥·Î ¸¸µé¾î Áø °ÍÀÌ´Ù. ÀÌ·¯ÇÑ RPC ¸¦ Á¦°øÇÏ´Â °ÍÀ¸·Î´Â Sun Open Network Computing (ONC)ÀÇ RPC ºÎ OSF DCE ÀÇ RPC OSF DCE °ê RPC ÀÇ.

2) Distributed Computing Environment (DCE)

DCE °Î OSF °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê OSF °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê °ê
Open Database Connectivity (ODBC) [14], Integrated Database Application Program Interface (IDAPI) [18], SQL Access Group (SAG) Call Level Interface (CLI) [16], "Glue" [17].

- Open Data Base Connectivity (ODBC)
- On Line Transaction Processing (OLTP)
- On Line Transaction Processing (OLTP)
Remote Procedure Call (RPC) は、Object Linking and Embedding (OLE) と並んで、Microsoft のオブジェクト指向プログラミングの技術として発展しました。RPC は、プログラム間の通信を提供し、OLE はオブジェクトのリンクと埋め込みを可能にしました。これらは、特に CORBA と DCOM に影響を及ぼしました。

以下に、ネットワーク技術の発展を示した図を挙げます。1982年から1998年までの間、ネットワーク技術は急速に進化しました。

1982年: Ethernet Era
1986年: Client/Server Era
1990年: DCE
1994年: CORBA
1998年: COM/DCOM

これらの技術は、ネットワークの開発の基盤を築き、より高度な通信を可能にしました。
2.3 Common Object Request Broker Architecture (CORBA)


2.3.1 Object Management Architecture (OMA)

OMG 对 OMA 的描述包括命名、持久性、生命循环、属性、并发性、集合、安全性、交易、关系、时间、变更管理、安全交易等 [图 7]。
2.3.2 CORBA

CORBA and OMA define a service interface that includes the following features: query, ORB (Object Request Broker), CORBA, persistence, concurrency, naming, life-cycle, security, printing, and mail. This service interface is known as the Common Object Service Specification (COSS), which is implemented in CORBA. OMA is an extension of CORBA that provides additional features.

CORBA is the core of CORBA and OMA, which is implemented in ORB Core. The ORB Core is responsible for providing the basic functionality of CORBA, including object identity, object location, and object creation. This functionality is provided through the CORBA interface, which defines the methods that clients can use to interact with objects.

In summary, CORBA is a standard for distributed object computing that provides a uniform interface for accessing remote objects. It is implemented in ORB Core, which provides the basic functionality of CORBA. OMA is an extension of CORBA that provides additional features, such as persistence, concurrency, and security. These features are implemented in the Common Facilities and Application Objects of CORBA.
Object Implementation と Object Adaptor は、それぞれ実装側とオブジェクト側を表す。特に、実装側は実装側スケルトンSkeleton（Implementation Skeleton）や、オブジェクトアダプターObject Adaptorを指す。実装側スケルトンは、インターフェースRepository（Interface Repository）を介して、実装側とオブジェクト側を連絡する。また、実装側は実装側スケルトンを介して、リモート側の実装を呼び出す。
CORBA 2.0 は、実装リポジトリ（Implementation Repository）を提供する。

CORBA は、動的（Dynamic Invocation）と静的（Static Invocation）の 2 つの形式を提供します。ORB は、API を提供し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出すことができます。ORB は、実装リポジトリからオブジェクトの実装を取得し、アプリケーションは実行時にコードを呼び出し
2.4.3 (vendor) / ñë¬ï 3 °èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ ÀÌ ±¸Ãà »ç·Ê°¡ ÈçÇÏÁö ¾Ê´Ù.

2.4.1 µ¿¿ø»ê¾÷

µ¿¿ø»ê¾÷ Àº Á¾ÇÕ½Äǰȸ»ç·Î¼­¸ÞÀÎÇÁ·¹ÀÓ±Þ IBM 4381 ¹ìµé¿þ¾î ¼­¹ö /±µº½º ¡¡ ¡¡ ¡¡ ¡¡ ¡¡ ¡¡ ¡¡ [38].

µ¿¿ø»ê¾÷ Àº µ¥º£À̽º´Â À¶óŬÀ» °³¹ßÅøÀº À¶óŬ ÊëÁî µîÀ» »ç¿ëÇßÀ¸¸ç. Ú¶óÀÌ¾ðÆ®¿Í ñë¬ï HP 9000/70 À¥¼­¹ö. À¶ó¿ìÅÍ ÀüÀÚ¿ìÆí°ü¸® ÇÁ·¹ÀÓ, °ë¿ìÃ÷ ÀüÀÚ ¿µ¾÷¼Ò.
2.4.2 2.4.2

SUN E5000 6  ES6000  10  ES3000  4  PC  1700

M760
10. The diagram shows the network configuration.

- 100M exchange - P (Central Exchange) network
- 3 sub-exchanges: SMS, NMS, MIS
- TP connects to the sub-exchanges
- TP connects to the network
- Three sub-exchanges / servers
- TP connects to the sub-exchanges
- TP connects to the network
- The central database (DBMS) connects to the network
- TP connects to the sub-exchanges
- 3 sub-exchanges
- TP connects to the network
- The network diagram represents the network topology.
3. Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ Çö»óºÐ¼®

ÀÌ Àå¿¡¼­´Â Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛÀÇ ¿¹·Î¼­ Æ÷½ºÄÚÀÇ ÀüüÀûÀÎ ½Ã½ºÅÛ ¾Æ Å°ÅØÃ³¿¡ ´ëÇÑ ³»¿ëÀ» »ìÆì º¸°í, ÇöÀç Æ÷½ºÄÚ¿¡¼­ ±¸Ãà ¿î¿µÁßÀÎ ¾à 70 ¿©°³ÀÇ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ¿¡ ´ëÇØ¼­

3.1 Æ÷½ºÄÚ ½Ã½ºÅÛ ¾ÆÅ°ÅØÃ³

°ü¸®ÇÏ´Â ¸ÞÀÎÇÁ·¹ÀÓ ñºÀ̽ºÀÇ HOST ½Ã½ºÅÛÀÌ Áö¿ªº° ¼­¿ï, Æ÷Ç×, ±¤¾çÀ¸·Î ºÐ»êµÇ¾î ÀÖ°í, ÃÖ±Ù¿¡ ±¸ÃàÇÏ¿© ¿î¿µ Áß¿¡ Àִ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ, °øÀå¿¡¼­ÀÇ °øÁ¤ Á¦¾î¸¦ ¸Ã°í ÀÖ´Â ÇÁ·Î¼¼½º ÄÄÇ»ÅÍ (Process Computer) ÀÌ 70 áì ¿î ¿¬°á¾î¿ë ¼­¿ïÀÇ ATM, Æ÷Ç×, ±¤¾çÀÇ FDDI LAN À¸·Î ±¸¼ºµÇ¾î ÀÖ°í ±× ÇϺο¡ °¢ °øÀå ¹× »ç¹«½Ç ´ÜÀ§·Î EthernetÀÌ ±¸¼ºµÇ¾î ÀÖ´Ù. ¶ÇÇÑ °®¾óŸÀÓ ÇÁ·Î¼¼½º·Î Æ®·¡ÇÈÀÌ °¹ÀÌ ¹ß»ýµÇ´Â ¾÷¹«°¡ ´ëºÎºÐÀÇ ½Ã½ºÅÛÀÌ ¼­·Î ÀÌÁúÀûÀΠȯ°æÀ¸·Î ±¸¼ºµÇ¾î ÀÖ°í, ƯÈ÷ Á¦Ã¶¼Ò Á¶¾÷°ü¸® ½Ã½ºÅÛÀº ¿¬Áß ¹«ÈÞ·Î ¿î¿µµÇ¸ç ±×·¡¼­ ½Ã½ºÅÛ Áß´ÜÀÌ ¹Ù·Î Á¶¾÷ Áß´Ü À¸·Î ¿¬°áµÇ¾î ¸·´ëÇÑ ¼Õ½ÇÀ» °¡Á® ¿Ã ¼ö ÀÖ¾î ½Ã½ºÅÛ ¾ÈÁ¤¼ºÀÌ ¹«¾ùº¸´Ù Áß¿äÇÏ Àº.

³×Æ®¿÷ ±¸¼ºÀº ¼­¿ï, Æ÷Ç×, ±¤¾çÀ» ¿¬°áÇÏ´Â T1 ±Þ WAN ÀÌ ¼³Ä¡µÇ¾î ÀÖ°í °¢ Áö¿ªº° ¸ÞÀÎÇÁ·¹ÀÓÀ» Áß½ÉÀ¸·Î ÇÑ Á¶¾÷¿ë ±¤ LAN °ú Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛÀ» Áß½ÉÀ¸·Î ÃÖ±Ù¿¡ ¼³Ä¡µÈ ¼­¿ïÀÇ ATM, Æ÷Ç×, ±¤¾çÀÇ FDDI LAN À¸·Î ±¸¼ºµÇ¾î ÀÖ°í ±× ÇϺο¡ °¢ °øÀå ¹× »ç¹«½Ç ´ÜÀ§·Î EthernetÀÌ ±¸¼ºµÇ¾î ÀÖ´Ù [±×¸² 11].

¶ÇÇÑ °®¾óŸÀÓ ÇÁ·Î¼¼½º·Î Æ®·¡ÇÈÀÌ °¹ÀÌ ¹ß»ýµÇ´Â ¾÷¹«°¡ ´ëºÎºÐÀÇ ½Ã½ºÅÛÀÌ ¼­·Î ÀÌÁúÀûÀΠȯ°æÀ¸·Î ±¸¼ºµÇ¾î ÀÖ°í, ¼­¹ö ½Ã½ºÅÛÀº UNIX, NT, OS/400, VMS µî ´Ù¾çÇÑ Á¾·ù °¡ ÀÖ°í °øÀå ¹× »ç¹«½Ç¿¡´Â ¾à 8000 ´ëÀÇ 486 ´ë 486 PC° ñ°.
27
Graphic User Interface (GUI) can be used to enhance user experience. RDBMS is a central component of such systems, allowing data manipulation through various interfaces including GUI. With a high usage of Data Gateway, the performance of RDBMS can be significantly improved.

3.2 Data Transfer Efficiency

Batch transfer is commonly used due to its high efficiency, often achieving 90% data transfer efficiency. With RDBMS oriented interfaces, the transfer process is further streamlined, especially when using Embedded SQL within Stored Procedure functions. This approach ensures efficient data manipulation and transfer.


29

3.2.1 2°èÃþ ±¸Á¶¿¡¼­ ¹ß»ýµÇ´Â ¹®Á¦Á¡

°¡Àå Å« ¹®Á¦Á¡À¸·Î ¼º´ÉÀÌ ÀúÇÏÇÏ´Â °ÍÀÌ´Ù. µ¿½Ã »ç¿ëÀÚ°¡ ÀûÀº ½Ã½ºÅÛÀÇ ¾à 80% ´Â 3~5ÃÊ À̳»ÀÇ ÀÀ´ä½Ã°£À» ³ªÅ¸³»Áö¸¸ µ¿½Ã »ç¿ëÀÚ¼ö°¡ 100 ÀÎ ÀÌ»óÀÎ °æ¿ì´Â ¼º´ÉÀÌ ÇöÀúÇÏ°Ô ¶³¾îÁ® 10 ÃÊ ÀÌ»óÀÇ ÀÀ´ä½Ã°£´ë¸¦ ³ªÅ¸³Â´Ù. ¹°·Ð »ç¿ë ÀÚ¼ö¿Í »ó°ü¾øÀÌ ÀÀ´ä¼Óµµ°¡ ´ÊÀº °ÍÀº ½Ã½ºÅÛ ±¸Á¶»óÀÇ ¹®Á¦À̱⠺¸´Ù´Â µ¥ÀÌŸº£À̽º ¼³°è ¹× ¾÷¹«¼³°è À߸øÀ¸·Î, µ¥ÀÌŸº£À̽º ¹× SQL Æ©´×À¸·Î ¼º´ÉÇâ»óÀÌ °¡´ÉÇÏ´Ù°í »ý°¢ µÈ´Ù.

µÑ°, ½Ã½ºÅÛ ¾ÈÁ¤¼ºÀÌ ºÎÁ·ÇÏ´Ù´Â °ÍÀÌ´Ù. ´ëºÎºÐÀÇ ½Ã½ºÅÛÀÌ ºÎ¼­°íÀ¯ÀÇ Åë°èºÐ¼® ¾÷¹«À̱⿡ Å« ÁöÀåÀº ¾ø´Ù°í´Â ÇÏÁö¸¸ ¼­¹ö ½Ã½ºÅÛ°ú µ¥ÀÌŸº£À̽º ½Ã ½ºÅÛÀÌ ÀÚÁÖ ´Ù¿îµÈ´Ù. Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ °³¹ßÃ hakk°ü¸® °æÇèºÎÁ·¿¡¼­ ¿øÀÎ ºÐ¼® ¹× Àå¾Öº¹±¸¿¡µµ ¸¹Àº ½Ã°£ÀÌ ¼Ò¿äµÇ°í ÀÖ´Ù.

¼Â°, ½Ã½ºÅÛ È°¿ë¼º ÀúÇÏ´Ù. ´ëºÎºÐÀÇ ½Ã½ºÅÛÀÌ ºÎ¼­ ´ÜÀ§ ´Üµ¶½Ã½ºÅÛ, Áï ÇϳªÀÇ ¼­¹ö¿¡ ÇϳªÀÇ µ¥ÀÌŸº£À̽º ¼­¹ö·Î¼­ ¿î¿µµÇ´Â °ÍÀ¸·Î ¸ÞÀÎ ÇÁ·¹ÀÓ°úÀÇ ¿¬°è Ȱ¿ëÀÌ ¾î·Æ´Ù.
3.2.2 量子論てん甲

量子論てん甲・理論・量子・投射・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計算・量子・計
3.4 ¿î¿µÀÇ Çö»ó  

Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛÀÇ °¡Àå Å« ÀåÁ¡ÀÌ »ç¿ëÀÚ¿¡°Ô Æí¸®ÇÑ ÀÎÅÍÆäÀ̽º¸¦ Á¦°øÇÏ´Â ¹Ý¸é¿¡, Ŭ¶óÀ̾ðÆ®ÀÇ ¾ÖÇø®ÄÉÀ̼ÇÀÌ ¸¹¾ÆÁö°í º¹ÀâÇØ Áü¿¡ µû¶ó ½Ã½º ÅÛ º¸¾È, Àå¾Ö¿¡ ´ëÇÑ º¹±¸, µ¥ÀÌŸ ¹é¾÷ µîÀÇ À¯Áöº¸¼ö¿¡ »ó´çÇÑ ºñ¿ë°ú ³ë·ÂÀÌ ÇÊ¿äÇÏ´Ù.

3.4.1 Àå¾Ö¹ß»ý À¯Çü  

¿î¿µ¼º Ãø¸é¿¡¼­µµ °¡Àå Å« ¹®Á¦Á¡ÀÌ ½Ã½ºÅÛ ¾ÈÁ¤¼º¿¡ ´ëÇÑ ´ëÃ¥ÀÌ ºÎÁ·ÇÏ´Ù °ÍÀÌ´Ù. Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛÀÌ Ãʱ⠱â¼ú·ÂÀÌ ¿ÏÀüÈ÷ Á¤ÂøµÇÁö ¾ÊÀº ½ÃÁ¡ ¿¡ µ¥ÀÌŸº£À̽º ¹× ¼­¹ö ½Ã½ºÅÛ¿¡ ´ëÇÑ ¿øÀÎÀ» ¾Ë ¼ö ¾ø´Â Àå¾Ö°¡ ´Ù·® ¹ß»ýÇÑ ´Ù. ¶ÇÇÑ Å¬¶óÀÌ¾ðÆ® ¼ö°¡ Áõ°¡Çϸ鼭 ¾ÖÇø®ÄÉÀÌ¼Ç ºÎ¹®ÀÇ Àå¾Ö°¡ ¸¹°í, Ŭ¶óÀÌ ¾ðÆ® ȯ°æ¼³Á¤ ÆÄÀÏÀÇ »ç¿ëÀÚ ÀÓÀÇ º¯°æÀ¸·Î ÀÎÇÑ Àå¾Öµµ »ó´çºÎºÐ ¹ß»ýÇϰí ÀÖ´Ù.

3.4.2 ¾ÖÇø®ÄÉÀÌ¼Ç ¹èÆ÷¹æ¹ý  

°³¹ß Ãʱ⿡´Â °¢°¢ÀÇ Å¬¶óÀÌ¾ðÆ®¿¡ ÀÏÀÏÀÌ ¾ÖÇø®ÄÉÀ̼ÇÀ» µî·ÏÇÏ´Â ¹æ¹ýÀ» »ç¿ëÇØ ¿ÔÀ¸³ª, ÇöÀç´Â ´ëºÎºÐ ¼­¹ö¿¡¼­ ÇØ´ç Ŭ¶óÀÌ¾ðÆ®¿¡ Àϰý ´Ù¿î·ÎµåÇÏ´Â.  

3.4 ½Ã½ºÅÛ ¿î¿µÀÇ Çö»ó, Naming ¾Ê, Data Dictionary ¾Ê ÀÎÇÑ ÀÎÅÍÆäÀ̽º¸¦
3.4.3 (Back up)

3.4.4 °³¹ß½Ã ¿ì¼±¼øÀ§
¾ø¾î ½Ã½ºÅÛ È¿À²ÀÌ ÀúÇÏµÈ Ãø¸éÀÌ ÀÖ´Ù.

Áö±Ý±îÁö Æ÷½ºÄÚ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ ¾ÖÇø®ÄÉÀ̼ǿ¡ ´ëÇÑ Çö»ó°ú ¹®Á¦
Á¡À» ºÎ¹®º°·Î »ìÆì º¸¾Ò´Ù. ÀÌ·¯ÇÑ ½ÇÁ¦ ¿î¿µ»óÀÇ ¹®Á¦Á¡À» Åä´ë·Î È¿À²ÀûÀΠŬ
¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ ±¸Çö ¹æ¾È¿¡ ´ëÇØ »ìÆì º¸°Ú´Ù.
4. 3°èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ ±¸Çö ¹æ¾È ÀÌ Àå¿¡¼­´Â ±â¾÷ Àüü¿¡ Àû¿ë °¡´ÉÇÑ 3 °èÃþ ±¸Á¶ÀÇ ¿£ÅÍÇÁ¶óÀÌÁî Ŭ¶óÀÌ ¾ðÆ®/¼­¹ö ½Ã½ºÅÛÀÇ Àû¿ë¹æ¾È, Àû¿ë½ÃÀÇ °í·Á»çÇ×, ¿ä¼Ò±â¼ú µî¿¡ ´ëÇØ ¼³¸íÇÑ´Ù. 4.1 ¾ÆÅ°ÅØÃ³ ±¸¼º ¹æ¾È ±â¾÷¿¡¼­ ¿£ÅÍÇÁ¶óÀÌÁî Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ ÇÁ·ÎÁ§Æ®¸¦ °èȹÇÒ ¶§ Ư¼º À» ¹«½ÃÇÏ°í ´Ü¼øÈ÷ Çϵå¿þ¾î Áõ¼³¸¸À¸·Î ÇØ°á °¡´ÉÇÑ °ÍÀ¸·Î »ý°¢ÇÑ´ÙµçÁö, Çö Àç ³ª¿Í ÀÖ´Â »ó¿ëÅøÀÌ À̵éÀ» ÀÚµ¿À¸·Î Áö¿øÇØ ÁØ´Ù°í ¹Ï´Â °æÇâÀÌ ÀÖ´Ù. ÀÌ´Â °³¹ßÀÚµéÀÌ ¿£ÅÍÇÁ¶óÀÌÁî Ŭ¶óÀ̾ðÆ®/¼­¹ö¿¡ ´ëÇÑ Á¤È®ÇÑ ±â¼úÀ» °¡Áö°í ÀÖÁö ¾Ê±â ¶§¹®ÀÌ´Ù. ¿£ÅÍÇÁ¶óÀÌÁî Ŭ¶óÀ̾ðÆ®/¼­¹ö¿¡ ´ëÇÑ Æ¯¼ºÀ» ÃæºÐÈ÷ ¹Ý¿µ ÇÏ¿© È¿À²ÀûÀÎ ½Ã½ºÅÛ ±¸Ãà [±×¸² 12]À» À§ÇØ °í·Á µÇ¾î¾ß ÇÒ ³»¿ëÀº ´ÙÀ½°ú °°´Ù [9, 26].
idle

50

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle

idle
ÄÉÀÌ¼Ç °³¹ß½Ã¿¡ Àç»ç¿ëÇÒ ¼ö ÀÖµµ·Ï ÇÑ´Ù.
¿©¼¸Â°, Àüü½Ã½ºÅÛ ¾ÆÅ°ÅØÃ³¸¦ °í·ÁÇÏ¿© ½Ã½ºÅÛÀ» ±¸ÃàÇØ¾ß ÇÑ´Ù. ÁýÁßµÈ ÄÄÇ»ÅÍ È¯°æÀ¸·ÎºÎÅÍ °¡´ÉÇÑ ¸¹Àº ÇÁ·Î¼¼½ÌÀ» ´ú±â À§Çؼ­´Â ¸®¸ðÆ®ÀÇ ¼­¹ö¸¦ ÀÌ ¿ëÇÑ´Ù.
Àϰö°, Áß¾Ó¿¡ ÁýÁßµÈ ÄÄÇ»ÆÃ ȯ°æ¿¡¼­ Àüü ½Ã½ºÅÛÀ» °ü¸® °¨µ¶ÇÒ ¼ö ÀÖµµ ·Ï ÇÑ´Ù. ´ëºÎºÐÀÇ ¼öÁ¤ ¹× º¯È­°¡ °ü¸®ÀÚ¿¡ ÀÇÇØ Áß¾Ó¿¡¼­ ÀÌ·ç¾î Áö¹Ç·Î ¹öÀü °ü¸®, À¯Áö º¸¼ö, ¹é¾÷, ½Ã½ºÅÛ Æ©´× µîÀÇ °ü¸®°¡ ÇÊ¿äÇÏ´Ù.
¿©±â¼­ °¡´ÉÇÑ ±âÁ¸ÀÇ ½Ã½ºÅÛ È¯°æÀ» ÃÖ´ëÇÑ È°¿ëÇϸ鼭 Çö»ó ¹®Á¦Á¡À» ÇØ °áÇÒ ¼ö ÀÖ´Â ¹æ¾ÈÀ» ¸¶·ÃÇϰíÀÚ Çß´Ù.
4.2 ¼­¹ö ½Ã½ºÅÛº° Àû¿ë¾÷¹«
¸ðµç Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛÀÌ ÀÏ·üÀûÀ¸·Î3 °èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö·Î °¡´Â °ÍÀº ¹Ù¶÷Á÷ÇÏÁö ¾Ê´Ù. ±âÁ¸ÀÇ ºÎ¼­´ÜÀ§ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ À» ±¸ÃàÇÏ ¿© ºñ±³Àû È¿À²ÀûÀ¸·Î Ȱ¿ëÇϰí ÀÖÀ¸¸ç, ´Ù¼ÒÀÇ ÀÀ´ä½Ã°£ ÀúÇϴ Ŭ¶óÀ̾ðÆ®/¼­ ¹ö ½Ã½ºÅÛ ±¸Ãà °æÇèºÎÁ·¿¡¼­ ¿À´Â °ÍÀ¸·Î ÆÛÆ÷¸Õ½º Æ©´×ÀÛ¾÷, ¾÷¹« ¹× µ¥ÀÌŸº£ À̽º Àç¼³°è µîÀ¸·Î ÃæºÐÈ÷ ÇØ°á °¡´ÉÇÑ ¹®Á¦´Ù. µû¶ó¼­ ¾÷¹« À¯Çü, »ç¿ëÀÚ¼ö, ½Ã½ºÅÛ ±Ô¸ð¿¡ µû¶ó ¾ÆÅ°ÅØÃ³¸¦ ±¸¼ºÇØ¾ß ÇÑ´Ù. ±×·¡¼­ ¾ÆÅ°ÅØÃ³ ±¸¼º ±âÁØÀº ´ÙÀ½°ú °°´Ù.
ù°, ·ÎÄà ¼­¹ö ½Ã½ºÅÛÀº ±âÁ¸¿¡ ±¸ÃàµÇ¾î ÀÖ´Â 2 °èÃþ ÇüÅÂÀÇ ½Ã½ºÅÛÀ¸·Î ¼­¹ö¿¡´Â µ¥ÀÌÅͺ£À̽º¿Í Stored procedure ·Î ±¸¼ºµÇ¾î ÀÖ°í Ŭ¶óÀÌ¾ðÆ®¿¡¼­´Â ¾ÖÇø®ÄÉÀÌ¼Ç ·ÎÁ÷À» Æ÷ÇÔÇÑ »ç¿ëÀÚ ÀÎÅÍÆäÀ̽º°¡ ³õÀÌ´Â ÇüÅ·μ­ ´ÙÀ½ÀÇ °æ¿ì ¿¡ ÁÖ·Î Àû¿ëÇÑ´Ù.
¶ÜªÀ Ó÷¹«
?
¿¬ÀÎ ¶ÌÇℹÔ­ ºÎ¼­ ´ÜÀ§ ¼Ò±Ô¸ð ¾÷¹«¿¡ Àû¿ë.
?
´ëºÎºÐÀÇ µ¥ÀÌŸ°¡ ´ÜÀÏ µ¥ÀÌŸº£À̽º·Î ±¸¼ºµÇ¾î ÀÖ´Â ¾÷¹«·Î¼­ Ÿ ºÎ¹®°ú ÀÎÅÍÆäÀ̽º°¡ ÀûÀº µ¶¸³µÈ ¾÷¹«
?
»ç¿ëÀÚ°¡ º¸´Ù ½±°í Æí¸®Çϵµ·Ï ÇÏ´Â ½Ã½ºÅÛÀÇ À¯¿¬¼ºÀ» ÇÊ¿ä·Î ÇÏ´Â ¾÷¹«ÁÖ¿ä Àû¿ë ¾÷¹«´Â ´ëºÎºÐÀÌ Batch¼º ¾÷¹«·Î¼­ Åë°èó¸®, ºÐ¼®¾÷¹«, ¸®Æ÷Æ®¹ßÇà µîÀÇ ¾÷¹«¿¡ Àû´çÇÏ´Ù.
?
µÑ°, ºÎ¹® ¼­¹ö ½Ã½ºÅÛÀº ¼­¹ö¿¡ µ¥ÀÌÅͺ£À̽º¿Í ¾ÖÇø®ÄÉÀÌ¼Ç ·ÎÁ÷À» µÎ°í Ú¶óÀÌ¾ðÆ®¿¡¼­´Â »ç¿ëÀÚ ÀÎÅÍÆäÀÌ¸¸À» µÎ´Â ÇüÅ´Ù. ºÎ¼­°£¿¡ ¿©·¯ °³ÀÇ ¼­¹ö µ¥ÀÌŸ¸¦ °áÇÕÇØ¼­ ¾×¼¼½º°¡ ÇÊ¿äÇÒ ¶§ Àû¿ëµÇ´Â ¾÷¹«·Î¼­ ÁÖ¿ä ±âÁØÀº ´ÙÀ½°ú°°´Ù.
?
µ¥ÀÌŸ ¹ß»ýÀÌ ¿©·¯ ºÎ¼­¿¡¼­ ¹ß»ýÇÏ°í µ¥ÀÌŸ °»½Åº¸´Ù´Â °Ë»öÀ» ÁÖ·Î ÇÏ´Â ¾÷¹«
?
µ¥ÀÌŸº£À̽º ±Ô¸ð°¡ 50~200GB ±Ô¸ðÀÇ Áß´ëÇü ¾÷¹«
?
response timeÀÌ 3~10ÃÊ Á¤µµ·Î¼­ ÆÛÆ÷¸Õ½º¿¡ Á÷Á¢ÀûÀÎ ¿µÇâÀ» ÁÖÁö ¾ÊÀº ¾÷¹«
?
»ç¿ëÀÚ¼ö°¡ 100~500¸í Á¤µµÀÇ ½Ã½ºÅÛ¿¡ Àû´çÇÏ´Ù.
?
¿Â¶óÀÎ ¾÷¹«°¡ ´ëºÎºÐÀÌÁö¸¸ ¸®¾ó¼ºÀ» ¿äÇÏÁö ¾ÊÀº ¾÷¹«ÁÖ¿ä Àû¿ë ¾÷¹«´Â ´ëºÎºÐÀÇ ºÎ¼­¿¡¼­ °øÅëÀ¸·Î »ç¿ëÇÏ´Â ¸ÞÀÏ ¼­¹ö, À¥¼­¹ö, ±×·ì¿þ¾î µîÀÇ ¾÷¹«¿¡ Àû´çÇÑ °ÍÀ¸·Î ±¸Ã¼Àû ±âÁØÀº ´ÙÀ½°ú°°´Ù.
4.3 13. ¾ÖÇø®ÄÉÀÌ¼Ç ±¸¼º ¹æ¾È

4.3 13. ¾ÖÇø®ÄÉÀÌ¼Ç ±¸¼º ¹æ¾È
4.4 3°èÃþ ±¸Á¶ ±¸Ãà½Ã °í·Á»çÇ×

3°èÃþ ±¸Á¶´Â ¸¹Àº ÀåÁ¡À» °¡Áö°í ÀÖÁö¸¸ ±¸Çö »ç·Ê°¡ ¸¹Áö ¾Ê°í ¶ÇÇÑ ¸ÞÀÌ Ä¿ (maker)¿¡¼­ ³ª¿Â ±â¼ú ¶ÇÇÑ ¿ÏÀüÇÏ°Ô °ÍÀÌ ¾Æ´Ï´Ù. µû¶ó¼­ Ŭ¶óÀ̾ð Æ®/¼­¹ö ½Ã½ºÅÛ ±¸Ãà¿¡ À־ Á¦Ç°¿¡ ´ëÇÑ ´ÜÆíÀûÀÎ ±â¼úÀÌ °ðµç °ÍÀ» ÇØ°áÇØ ÁÖ´Â °ÍÀ¸·Î ÀνÄÇÏ¸é ¾ÈµÈ´Ù. µû¶ó¼­ º¸´Ù Æ÷°ýÀûÀ̰í Àüü ½Ã½ºÅÛÀ» °¨¾ÈÇÑ ½Ã ½ºÅÛ ±¸ÃàÀÌ ÇÊ¿äÇϸç ÀÌ¿¡ ´ëÇÑ °í·Á»çÇ×°ú ¿¹»óµÇ´Â ¹®Á¦Á¡¿¡ ´ëÇØ ¾Ë¾Æ º¸¸é ÀñÀ½°ú °°´Ù.

½Ã½ºÅÛ °³¹ß¿¡ ¾Õ¼­ Ç¥ÁØÀ» ¸ÕÀú ¸¸µé¾î¾ß ÇÑ´Ù. ¹Ù·Î À̰ÍÀÌ Solution
Architecture ... Solution Architecture ... Framework ... [27].

- GU, Naming Convention, Network Protocol, Component Standard & project control standard.

- Packaged Software.
- [30]

- 2°ø Á¶°¡ ´ëºÎºÐ µ¥ÀÌŸº£À̽º¸¦ ¾×¼¼½ºÇÏ´Â ¹Ý¸é 3°ø Á¶°¡ ´Â ¾ÖÇø®ÄÉÀÌ¼Ç ·ÎÁ÷À» ¾×¼¼½º Çϱ⠶§¹®¿¡ ¹Ýµå½Ã ¾÷¹«Çõ½ÅÀ̳ª Ç¥ÁØÈ­¸¦ ÅëÇØ ¾÷¹« ·ÎÁ÷À» ¸íÈ®ÇÏ°Ô Á¤ÀÇÇØ¾ß ÇÑ´Ù.

- 2°èÃþ ±¸Á¶°¡ ´ëºÎºÐ ´Ü¼ø µ¥ÀÌŸº£À̽º¸¦ ¾×¼¼½ºÇÏ´Â ¹Ý¸é 3°èÃþ ±¸Á¶°¡ ´Â ¾ÖÇø®ÄÉÀÌ¼Ç ·ÎÁ÷À» ¾×¼¼½º Çϱ⠶§¹®¿¡ ¹Ýµå½Ã ¾÷¹«Çõ½ÅÀ̳ª Ç¥ÁØÈ­¸¦ ÅëÇØ ¾÷¹« ·ÎÁ÷À» ¸íÈ®ÇÏ°Ô Á¤ÀÇÇØ¾ß ÇÑ´Ù.
4.5 Åø ¼±Á¤ ±âÁØ

3 °è À ÎÇÑ À§Çè ½Å±Ô °³¹ß À§Çè Àû¿ëÇÑ ÀÇ ÀûÀº ¾÷¹«¿¡ ¿ì¼± ½Ã¹ü Àû¿ëÇÑ´Ù.

ÀÎÅͳݰúÀÇ ¿¬°è¸¦ °í·ÁÇÑ ½Ã½ºÅÛ °³¹ßÀÌ ÇÊ¿äÇÏ´Ù.

4.5.1 Åø °±ÀÇ °³¹ßÅø

3 °è À ÎÇÑ À§Çè ½Å¼ÓÈ÷

16ºñÆ® ÄÚµå (À©µµ¿ì 3.1)¿¡¼­ ÀÛ¼ºµÈ ¾ÖÇø®ÄÉÀ̼ÇÀ» À©µµ¿ì 95 ¿¡¼­ º° ´Ù¸¥ ÄÚµù¾øÀÌ ´Ù½Ã ÄÄÆÄÀÏÇÏ·Á¸é 32 ºñÆ® ¾ÖÇø®ÄÉÀ̼ÇÀ» »ý¼ºÇÒ ¼ö ÀÖ´Â À̽ļºÀÌ ÀÖ¾î¾ß ÇÑ´Ù.

½ºÅ©¸³Æ® ¾ð¾îÀ̾î¾ß ÇÑ´Ù.
(class)  (attribute)  

CASE Åø°úÀÇ ¿¬°áÀÌ °¡´ÉÇÏ¿© °³¹ß Åø È­¸éÀ¸·Î ÀÚµ¿ »ý¼ºÀÌ °¡´ÉÇØ¾ß ÇÑ´Ù.

4.5.2 ¹Ìµé¿þ¾î ¼±Á¤ ±âÁØ

°³¹ß Åø Àº °´Ã¼ÁöÇâ ¿øÄ¢¿¡ »ó´çÈ÷ ±â¹ÝÀ» µÎ°í ÀÖ´Ù.

ÇöÀçÀÇ ½Ã½ºÅÛ È¯°æ»Ó¸¸ ¾Æ´Ï¶ó ¹Ì·¡ÁöÇâÀû ±â¼ú Á¢±ÙÀÌ ÇÊ¿äÇÏ´Ù. ÇöÀç´Â µ¥ÀÌŸº£À̽º »ó¿¡¼­ ´Ù¾çÇÑ DBȣȯ¼ºÀ» °¡Áö°í »ç¿ëÇØ ¿ÔÁö¸¸, ÇâÈÄ ±â¼ú Ãß¼¼´Â °´Ã¼ÁöÇâÀÇ °³³äÀ» °¡Áö°í RDB ¿¡¸¸ ±¹ÇÑµÇ¾î »ç¿ëµÇÁö ¾Ê °í Object Oriented Database (OODB) µî ´Ù¾çÇÑ µ¥ÀÌŸº£À̽º¸¦ Áö¿øÇÒ µÎ°í ÀÖ´Â ¹Ìµé¿þ¾î ¼±ÅÃÀÌ ÇÊ¿äÇÏ´Ù.

°³¹ß ÅøÀº ¾ÖÇø®ÄÉÀÌ¼Ç Å¸ÀÔÀ» Áö¿øÇØ¾ß ÇÑ´Ù. ½Ã½ºÅÛ°£¿¡ ȣȯ¼ºÀº¸Þ½ÃÁö Çڵ鸵 (Message handling), Á¤º¸°øÀ¯, ´Ù¾çÇÑ API ÀÇ Áö¿ø µî ¸¹ ¿ä¼Ò°¡ ÀÖ´Ù.
RPC, GSF, DCE, Nantong, HP/UX, CORBA, Trade-Off, ±â¾÷ÀÇ ±¸ÄµÀÇ ±â´É ÀÚü¸¦ »ó½Ç ÇÔ²² Ç¥ÁØÈ­ÀÇ ¹®Á¦µµ °ËÅäÇÏ¿©¾ß ÇÑ´Ù. À߸øµÈ Á¦Ç°ÀÇ ¼±ÅÃÀº ¹Ì µé¿þ¾îÀÇ ±â´É ÀÚü¸¦ »ó½ÇÇϱ⠶§¹®ÀÌ´Ù.
5. 3° łê ÀÇ ÀÌ Àå¿¡¼­ ª 4 Àå¿¡¼­ Á¦½ÃÇÑ 3 °èÃþ ±¸ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ Àû¿ë ÇØ ±âÁ¾ÀÇ ±¸ ÇÑ´Ù. 4° À°¶å 3° ÁÇ ÀÌ Àå¿¡¼­ Á¬À²Àº ±ºÀº "±âÁ¾ÀÇ ±¸ ÇÑ´Ù. "

5.1 Æ÷½ºÄÚ ½Ã½ºÅÛ Åû¿ëÇØ ±¸°íÀÚ ÇÑ´Ù. ±¸¼ºµÇ¾î ÀÖ°í, ±Ù·¡¿¡ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ µµÀÔÀ¸·Î ±¸ÀÇ Åë°èºÐ¼®À» ¸ñ ÀûÀ¸·Î ±¸ÀÇ ¼ºµÇ¾î Áö°í ÀÖ´Ù. ÀÌ Àå¿¡¼­´Â 4 Àå¿¡¼­ ¼ºµÇ¾î ÀÖ°í, ±Ù·¡¿¡ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ µµÀÔÀ¸·Î ±¸ÀÇ Åë°èºÐ¼®À» ±¸Çö¹æ¾ÈÀ» Æ÷½ºÄÚ ½Ã½ºÅÛ¿¡ Àû¿ëÇØ ±¸°íÀÚ ÇÑ´Ù. ±×¸² 14. ±×¸² 14.
Wan%, Enterprise Server, IBM, MVS, 250GB, 0
Local Server, SUN, HP, UNIX, 100~250GB
Workgroup Server, Compaq, Windows NT, 100GB
Client, PC 486, Windows 95, 2GB

3 さまざまなオプションが用意されています。FTP, ODBC, DB-Library, SQL, など、さまざまなシステムが対応しています。
46
5.2 Overview of the System

[Diagram showing the system architecture with layers and components, including Enterprise Server, Local Server, and Workgroup Server, with relationships to Client Application, Server Application, and Database.]

System/ws: [Description of System/ws components]
System/T: [Description of System/T components]
System/Host: [Description of System/Host components]

15. [Additional notes or references]
6. °á·Ð ¹× ÇâÈÄ °èȹ

 À§ÇØ ¹æ¾È ¼ö¸³À» À§ÇØ °ü·Ã ¿¬±¸ºÎ¹®¿¡¼­ 2 °èÃþ±¸Á¶¿Í 3 °èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ¿¡ ´ëÇÑ ¹®Á¦Á¡À» ÇØ°áÇϱâ

 À§ÇÑ ¹æ¾È ¼ö¸³À» À§ÇØ °ü·Ã ¿¬±¸ºÎ¹®¿¡¼­ 2 °èÃþ±¸Á¶¿Í 3 °èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ¿¡ ´ëÇØ¼­ ¾Ë¾Æ º¸¾Ò´Ù. ±×¸®°í 3 °èÃþ Ŭ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ ±¸Ãà¿¡ À־ Ç'ö ±â¼úÀÎ ¹Ìµé¿þ¾î¿¡ µ¿ÀÛ¿ø¸®¿Í °¢ ±â´Éº° Á¾·ù¿¡ ´ëÇØ¼­ ¾Ë¾Æ º¸¾ÒÀ¸¸ç, °´Ã¼ Áö Ç⠹̵é¿þ¾îÀÇ °¡Àå ´ëÇ¥ÀûÀÎ ÄÚ¹Ù¿¡ ´ëÇØ¼­ ¾Ë¾Æ º¸¾ÒÀ¸¸ç, ¶ÇÇÑ ÇöÀç ±â¾÷¿¡¼­ 3 °èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ ±¸Çö »ç·Ê¸¦ »ìÆì º½À¸·Î¼­ º» ¿¬±¸ÀÇ ÁÖÁ¦ ÀÎ 3°èÃþ ±¸Á¶·ÎÀÇ ÀÌÇà ¹æ¾È ¼ö¸³¿¡ Âü°í·Î Ȱ¿ë Çß´Ù. ½ÇÁ¦ ±¸ÃàÇÏ¿© ¿î¿µÁßÀÎ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ¿¡ ´ëÇØ¼­ »ìÆì º½À¸ ·Î¼­ Çö»ó ¹®Á¦Á¡À» º¸´Ù ±¸Ã¼ÀûÀ¸·Î ¾Ë ¼ö ÀÖ¾úÀ¸¸ç ÀÌ·¯ÇÑ ºÐ¼® °á°ú ÇöÀç Ŭ ¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛÀÇ ´ëºÎºÐÀÌ 2 °èÃþ ±¸Á¶ÀÇ ¾ÆÅ°ÅØÃ³·Î ±¸¼ºµÇ¾î ÀÖÀ¸¸ç Á¡ À÷ È®´ëµÇ¾î °¡´Â °úÁ¤¿¡¼­ ¸¹Àº ¹®Á¦Á¡µéÀÌ µµÃâµÇ¾ú´Ù. ¸ÕÀú, »ç¿ëÀÚ¼ö¿Í ¾ÖÇÃ¸®ÄÉÀÌ¼Ç ¾ÆÅ°ÅØÃ³ ±¸¼º¹æ¾È, 3 °èÃþ ¾ÆÅ°ÅØÃ³ ±¸¼º½Ã °í·Á »çÇ× µîÀ» Á¦½Ã ÇÏ¿´´Ù. °¡´ÉÇÑ ±âÁ¸ÀÇ ½Ã½ºÅÛÀ» Ȱ¿ëÇϸé ¼­ 3 °èÃþÀ¸·Î ÀÌÇà¹æ¾ÈÀ¸·Î¼­ ´ÙÀ½À» ¿ì¼±ÀûÀ¸·Î °í·Á ÇÏ¿´´Ù. ù°, ¼º´É ÀúÇÏ ¿¡ ´ëÇÑ ¹æ¾ÈÀ¸·Î´Â ¾ÖÇø®ÄÉÀÌ¼Ç ¼­ºñ½º¸¦ ¿©·¯´ëÀÇ ¼­¹ö¿¡ ¹èÄ¡ÇÏ¿© ºÎÇϸ¦ ºÐ »êÇϰíÀÚ Çß°í, ³×Æ®¿÷ Æ®·¡ÇÈÀ» ÁÙÀ̱â À§ÇØ Áö¿ªº° ¼­¹ö¸¦ º°µµ·Î ±¸ÃàÇÏ¿´´Ù. µÑ°, ½Ã½ºÅÛ ¾ÈÁ¤¼ºÀ» À§Çؼ­´Â ±âÁ¸ÀÇ ¸ÞÀÎ ÇÁ·¹ÀÓ¿¡ °³¹æÇü ½Ã½ºÅÛÀ» µµÀÔÇÏ
¿© ¿£ÅÍÇÁ¶óÀÌÁî ¼­¹ö¸¦ ±¸ÃàÇÏ¿© Áß¿ä ¾÷¹«ÀÇ ¸ÞÀÎ µ¥ÀÌŸº£À̽º·Î Ȱ¿ëÇÏ¿´´Ù．

¶ÇÇÑ Áß¾Ó¿¡¼­ ÁýÁß °ü¸® °¨µ¶ÀÌ °¡´ÉÇÑ ½Ã½ºÅÛÀ» µµÀÔ Àû¿ëÇÏ¿´´Ù．¸¶Áö¸·À¸·Î，
Àü»ç ½Ã½ºÅÛÀÌ °³¹æÇüÀ¸·Î °¡¸é¼­ µ¥ÀÌŸ °øÀ¯ ¹üÀ§°¡ ³Ð¾îÁö°í »ç¿ëÀÚ´Â µ¥ÀÌŸ
À§Ä¡¿¡ ¹«°üÇÏ°Ô Á¢±Ù °¡´ÉÇÏ°Ô ÇÔÀ¸·Î¼­ Ȱ¿ë¼ºÀ» Áõ´ëÇÏ¿´´Ù．ÀÌ·¯ÇÑ ÀÌÇà ¹æ
¾ÈÀ» Æ÷½ºÄÚ ½Ã½ºÅÛ¿¡ Àû¿ëÇÏ¿© 3 °èÃþ ±¸Á¶ÀÇ Å¬¶óÀ̾ðÆ®/¼­¹ö ½Ã½ºÅÛ ¸ðµ¨À»
Á¦½Ã ÇÏ¿´´Ù．

ÇâÈÄ¿¡µµ
¹ßÀüÇØ
°¡´Â
Á¤º¸±â¼ú¿¡
´Éµ¿ÀûÀ¸·Î
´ëóÇϰí，
񃬣°æÀï·Â
¿ìÀ§¸¦
À§Çؼ­´Â
Â÷¼¼´ë
±â¼úÀ̶ó
ÇÒ
¼ö
ÀÖ´Â
ºÐ»ê°´Ã¼¸¦
Ȱ¿ëÇÑ
ÄÚµå
Àç»ç¿ë
±â¼ú°ú
¹Ì
µé¿þ¾îÀÇ
Àû¿ëÀ»
ÅëÇÑ
3 
°èÃþ
¹æ½ÄÀÇ
¾ÖÇø®ÄÉÀ̼Ç
°³¹ß·Î
°³¹ß
»ý»ê¼ºÀÇ
Çâ»ó
¹×
¿î¿µ
±â¼ú·ÂÀ»
È®º¸ÇÏ¿©¾ß
ÇÑ´Ù

À»
¿°µÎÇØ
µÎ°í
°³¹ßÇØ¾ß
Çϸç
Web
CORBA
３
°èÃþ
¼ö
ÀÖ´Â
ºÐ»ê°´Ã¼¸¦

ÀûÀÎ
¿¬µ¿
¹æ¾È¿¡
´ëÇØ¼­µµ
Áö¼ÓÀûÀÎ
¿¬±¸
³ë·ÂÀÌ
ÇÊ¿äÇÏ´Ù
．
[23] Peter J. Houston, “Introduction to DCE and Encina”,

51

[24] Encina and DCE for CICS,


[27] Transarc, “Building Large Scale Client-Server Systems Using the Encina Monitor”,


[35] This page defines 'NetBIOS' (Network Basic Input/Output System),
http://whatis.com/netbios.htm

[36] Named Pipes,
http://www.ece.sc.edu/class/eece890n/shared/projects/dan/namedpipes.htm


[38] Open Computing, Jan, 1997.