



Sun Java™ System

Sun Java Enterprise System 2005Q1

部署規劃指南

---

Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054  
U.S.A.

文件號碼：819-1922

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 版權所有。

Sun Microsystems, Inc. 對本文件中所描述產品中使用的技術擁有相關智慧產權。特別是 (但不僅限於)，這些智慧產權可能包括一項或多項在 <http://www.sun.com/patents> 上列出的美國專利，以及一項或多項美國和其他國家/地區的其他專利或待批專利。

**本產品包含 Sun Microsystems, Inc. 的機密資訊和商業秘密。未經 Sun Microsystems, Inc. 事先明確的書面許可，禁止使用、公開或複製本產品。**

美國政府權利 - 商業軟體。政府使用者應遵守 Sun Microsystems, Inc. 標準授權合約以及 FAR 及其增補文件中的適用條款。

本發行物可能包含由協力廠商開發的材料。

產品的某些部分可能源自 Berkeley BSD 系統，並經加州大學授權。UNIX 是在美國和其他國家/地區的註冊商標，由 X/Open Company, Ltd. 獨家授權。

Sun、Sun Microsystems、Sun 標誌、Java、Solaris、JDK、Java Naming and Directory Interface、JavaMail、JavaHelp、J2SE、iPlanet、Duke 標誌、Java 咖啡杯標誌、Solaris 標誌、SunTone Certified 標誌和 Sun ONE 標誌是 Sun Microsystems, Inc. 在美國和其他國家/地區的商標或註冊商標。

所有 SPARC 商標均在授權下使用，它們是 SPARC International, Inc. 在美國和其他國家/地區的商標或註冊商標。帶有 SPARC 商標的產品均基於 Sun Microsystems, Inc. 開發的架構。

Legato 和 Legato 標誌是註冊商標，它們和 Legato NetWorker 都是 Legato Systems, Inc. 的商標或註冊商標。Netscape Communications Corp 標誌是 Netscape Communications Corporation 的商標或註冊商標。

OPEN LOOK 和 Sun (TM) 圖形化使用者介面由 Sun Microsystems, Inc. 為其使用者和授權者開發。Sun 感謝 Xerox 在研究和設計電腦業中視覺化或圖形化使用者介面這個觀念上所作的領先努力。Sun 保有 Xerox 對 Xerox 圖形使用者介面非獨佔性的授權，這項授權也涵蓋獲得 Sun 授權使用 OPEN LOOK GUI 並符合 Sun 的書面授權合約的廠商。

本服務手冊所涵蓋的產品和包含的資訊受到美國出口控制法規的控制，並可能受到其他國家/地區進出口法規的管轄。嚴禁核子武器、飛彈、生化武器或海上核動力裝備等最終用途或最終使用者直接或間接使用本產品。嚴禁向受到美國禁運的國家/地區或美國出口除外清單 (包括但不僅限於被拒人清單和特別指定的國家/地區清單) 上標識的實體出口或再出口本產品。

本文件以「現狀」提供，所有明示或暗示的條件、陳述與保證，包括對於適銷性、特定用途的適用性或非侵權行爲的任何暗示性保證在內，均恕不負責，除非此免負責聲明在法律上被認為無效。

# 目錄

圖清單 .....	7
表清單 .....	9
<b>前言 .....</b>	<b>11</b>
本書適用對象 .....	12
本書的編排方式 .....	12
相關文件 .....	12
此文件集所包含的書籍 .....	12
線上存取 Sun 資源 .....	14
連絡 Sun 技術支援 .....	14
相關協力廠商網站參照 .....	14
Sun 歡迎您提出意見 .....	15
<b>第 1 章 部署規劃簡介 .....</b>	<b>17</b>
關於 Java Enterprise System .....	17
系統服務 .....	17
內建服務和自訂開發服務 .....	19
遷移到 Java Enterprise System .....	20
關於部署規劃 .....	21
解決方案生命週期 .....	21
業務分析階段 .....	23
技術需求階段 .....	23
邏輯設計階段 .....	24
部署設計階段 .....	24
部署實作階段 .....	25
作業階段 .....	26
<b>第 2 章 業務分析 .....</b>	<b>27</b>
關於業務分析 .....	27
定義業務需求 .....	28

設定業務目標	28
範圍	28
優先性	28
關鍵特性	28
成長因素	28
安全空間	28
瞭解使用者需求	29
開發作業需求	30
支援現有的使用模式	30
瞭解公司文化	31
利害關係人	31
標準和策略	31
規管需求	31
安全性	31
站點分散	32
使用遞增方式	32
瞭解服務層級合約	32
定義業務限制	33
遷移問題	33
排程指定作業	33
預算限制	34
所有權成本	34
<b>第 3 章 技術需求</b>	<b>35</b>
關於技術需求	35
使用分析	36
使用實例	38
服務品質需求	38
效能	40
可用性	40
容錯系統	41
排定服務可用性優先順序	41
服務損失	42
延展性	42
估計成長	43
安全需求	44
安全性規劃的元素	45
潛在容量	46
可服務性需求	46
服務層級需求	47

<b>第 4 章 邏輯設計</b>	<b>49</b>
關於邏輯架構	49
設計邏輯架構	50
Java Enterprise System 元件	51
元件相依性	52
Web 容器支援	55
Messaging Server 提供的不同邏輯的服務	55
存取元件	56
多層架構設計	56
範例邏輯架構	58
Messaging Server 範例	58
Messaging Server 使用實例	59
基於身份識別的通訊範例	62
身份識別型通訊範例的使用實例	63
存取區域	65
部署方案	67
<b>第 5 章 部署設計</b>	<b>69</b>
關於部署設計	69
專案核准	70
部署設計輸出資料	70
影響部署設計的因素	71
部署設計方法	73
估計處理器需求	74
範例估計處理器需求	75
決定使用者進入點的基線 CPU 估計	76
包括服務相依性的 CPU 估計	77
研究尖峰負載使用的使用實例	77
修改其他負載情況的估計	78
更新 CPU 估計	79
估計安全交易的處理器需求	80
安全交易的 CPU 估計	80
處理 SSL 交易的專用硬體	82
決定可用性策略	83
可用性策略	83
單一伺服器系統	84
水平備援系統	84
Sun Cluster 軟體	86
可用性設計範例	87
Messaging Server 的負載平衡範例	87
使用 Sun Cluster 軟體的防故障備用範例	88
目錄服務複製的範例	89
單一主機複製	90

多重主機複製 .....	90
決定延展性的策略 .....	91
潛在容量 .....	92
延展性範例 .....	92
找出效能瓶頸 .....	93
最佳化磁碟存取 .....	95
設計最佳化資源使用 .....	96
管理風險 .....	97
範例部署架構 .....	98
<b>第 6 章 實作部署設計 .....</b>	<b>101</b>
關於實作部署設計 .....	101
安裝和配置軟體 .....	102
開發引導和原型 .....	103
測試引導和原型部署 .....	103
建置生產部署 .....	104
<b>索引 .....</b>	<b>105</b>

# 圖清單

圖 1-1	解決方案生命週期 .....	22
圖 4-1	Java Enterprise System 解決方案架構的三個要素 .....	50
圖 4-2	Java Enterprise System 元件 .....	51
圖 4-3	Java Enterprise System 元件相依性 .....	54
圖 4-4	多層架構模型 .....	57
圖 4-5	Messaging Server 部署的邏輯架構 .....	58
圖 4-6	顯示使用實例 1 的 Messaging Server 邏輯架構 .....	60
圖 4-7	顯示使用實例 2 的 Messaging Server 邏輯架構 .....	61
圖 4-8	顯示使用實例 3 的 Messaging Server 邏輯架構 .....	62
圖 4-9	身份識別型通訊方案的邏輯架構 .....	63
圖 4-10	顯示使用實例 1 的通訊方案邏輯架構 .....	64
圖 4-11	顯示使用實例 2 的通訊方案邏輯架構 .....	65
圖 4-12	放置在存取區域中的邏輯元件 .....	66
圖 5-1	身份識別型通訊方案的邏輯架構 .....	76
圖 5-2	單一伺服器系統 .....	84
圖 5-3	N+1 防故障備用系統和兩個伺服器 .....	85
圖 5-4	兩個伺服器之間的負載平衡和防故障備用 .....	85
圖 5-5	在 $n$ 個伺服器之間分配負載 .....	86
圖 5-6	範例的 Messaging Server 可用性 .....	88
圖 5-7	使用 Sun Cluster 軟體的防故障備用設計 .....	89
圖 5-8	單一主機複製範例 .....	90
圖 5-9	多重主機複製範例 .....	91
圖 5-10	水平和垂直調整的範例 .....	93
圖 5-11	範例部署架構 .....	99



# 表清單

表 1	Java Enterprise System 文件	13
表 1-1	Java Enterprise System 服務類別	18
表 3-1	使用分析因素	36
表 3-2	系統品質影響 QoS 需求	39
表 3-3	系統執行一整年 (8760 小時) 的未排程當機時間	41
表 3-4	依優先性排列的服務可用性	42
表 3-5	延展性因素	43
表 3-6	可服務性需求的主題	46
表 4-1	Java Enterprise System 元件相依性	52
表 4-2	Messaging Server 配置	55
表 4-3	Java Enterprise System 提供遠端存取的元件	56
表 4-4	多層架構 中的邏輯層	57
表 4-5	Messaging Server 邏輯架構中的元件	59
表 4-6	安全存取區域和其中所放置的元件	66
表 5-1	元件的 CPU 估計包含存取使用者進入點的基線 CPU 估計	76
表 5-2	支援元件的 CPU 估計	77
表 5-3	CPU 估計尖峰負載的調整	78
表 5-4	支援元件的 CPU 估計調整	79
表 5-5	修改 CPU 安全交易的估計	81
表 5-6	Portal Server 安全交易的 CPU 估計調整	82
表 5-7	支援元件的 CPU 估計調整	87
表 5-8	資料存取點	94
表 5-9	資源管理考量	96



# 前言

「Java Enterprise System 部署規劃指南」介紹如何規劃和設計以 Sun Java™ Enterprise System 為基礎的企業部署解決方案。本指南說明部署規劃和設計的一些基本概念及原則、討論解決方案生命週期（其中包含部署設計專案的階段及作業），以及提供您可在規劃整個企業的部署解決方案時，與 Java Enterprise System 搭配使用的高階範例和策略。

本前言包含以下各節：

- 第 12 頁的「本書適用對象」
- 第 12 頁的「相關文件」
- 第 14 頁的「線上存取 Sun 資源」
- 第 14 頁的「連絡 Sun 技術支援」
- 第 14 頁的「相關協力廠商網站參照」
- 第 15 頁的「Sun 歡迎您提出意見」

執行本指南描述的任何作業前，請先閱讀「Java Enterprise System 版本說明」(<http://docs.sun.com/doc/819-0818>)。

## 本書適用對象

本指南主要是為負責企業部署分析及設計的部署架構師和業務規劃人員所編寫。本指南也適用於系統整合師和其他負責企業應用程式各種層面的設計及實作人員。

本指南假設您熟悉企業層級應用程式的設計和安裝，也假設您已經讀過「Java Enterprise System 技術摘要」。

## 本書的編排方式

本指南以解決方案生命週期為基礎，描述部署規劃的各種階段。第 1 章，「部署規劃簡介」將介紹解決方案生命週期。

## 相關文件

<http://docs.sun.com><sup>SM</sup> 網站可讓您線上存取 Sun 技術文件。您可以瀏覽歸檔檔案或搜尋特定的書名或主旨。

## 此文件集所包含的書籍

Java ES 手冊以線上檔案方式提供，有可攜式文件格式 (PDF) 和超文字標記語言 (HTML) 兩種格式。殘障使用者可以使用輔助技術閱讀這兩種格式的檔案。可以從此處存取 Sun™ 文件網站：

<http://docs.sun.com>

Java ES 文件包括與整個系統相關的資訊及其元件的相關資訊。可以從以下位置存取此文件：

<http://docs.sun.com/prod/entsys.05q1> 與  
[http://docs.sun.com/prod/entsys.05q1?l=zh\\_TW](http://docs.sun.com/prod/entsys.05q1?l=zh_TW)

下表列出 Java ES 文件集中的系統層級手冊。左欄為每份文件的名稱及文件號碼位置，右欄為文件的一般性內容。

表 1 Java Enterprise System 文件

文件	內容
Java Enterprise System 版本說明 <a href="http://docs.sun.com/doc/819-0818">http://docs.sun.com/doc/819-0818</a>	包含有關 <b>Java Enterprise System</b> 的最新資訊，包括已知問題。此外，元件產品均有各自的版本說明。
Java Enterprise System 文件資訊指南 <a href="http://docs.sun.com/doc/819-1915">http://docs.sun.com/doc/819-1915</a>	提供與 <b>Java Enterprise System</b> 相關之文件的描述。包括與元件相關之文件的連結。
Java Enterprise System 技術摘要 <a href="http://docs.sun.com/doc/819-1929">http://docs.sun.com/doc/819-1929</a>	介紹 <b>Java Enterprise System</b> 中使用的技術及概念基礎。描述元件、架構、流程和功能。
Java Enterprise System 部署規劃指南 <a href="http://docs.sun.com/doc/819-1922">http://docs.sun.com/doc/819-1922</a>	介紹如何規劃和設計以 <b>Java Enterprise System</b> 為基礎的企業部署解決方案。說明部署規劃和設計的一些基本概念及原則、討論解決方案生命週期，以及提供您在規劃以 <b>Java Enterprise System</b> 為基礎的解決方案時使用的高階範例和策略。
Java Enterprise System 使用者管理指南 <a href="http://docs.sun.com/doc/819-2231">http://docs.sun.com/doc/819-2231</a>	協助您規劃、部署和管理 <b>Java Enterprise System</b> 解決方案之使用者的相關資訊。在解決方案生命週期的每個階段描述使用者管理問題，以補充「 <b>Java Enterprise System</b> 部署規劃指南」。
Java Enterprise System 部署範例系列：評估方案 <a href="http://docs.sun.com/doc/819-1669">http://docs.sun.com/doc/819-1669</a>	描述如何在一個系統上安裝 <b>Java Enterprise System</b> 、建立一組核心、共用的網路服務，以及設定可存取您所建立之服務的使用者帳號。
Java Enterprise System 安裝指南 <a href="http://docs.sun.com/doc/819-0811">http://docs.sun.com/doc/819-0811</a>	引導您完成安裝 <b>Solaris™</b> 作業系統或 <b>Linux</b> 作業系統的 <b>Java Enterprise System</b> 。告訴您如何選取要安裝的元件產品、如何在安裝後配置那些元件，以及如何確認配置的軟體是否運作正常。
Java Enterprise System 升級與遷移指南 <a href="http://docs.sun.com/doc/819-2238">http://docs.sun.com/doc/819-2238</a>	提供資訊和指示，來升級 <b>Solaris™</b> 作業系統或 <b>Linux</b> 作業環境的 <b>Java Enterprise System</b> 。
Java Enterprise System 字彙表 <a href="http://docs.sun.com/doc/819-1936">http://docs.sun.com/doc/819-1936</a>	定義 <b>Java Enterprise System</b> 文件中使用的術語。

## 線上存取 Sun 資源

如需產品下載、專業服務、修補程式和支援，以及其他開發人員資訊，請造訪下列網站：

- 下載中心  
<http://www.sun.com/software/download/>
- 專業服務  
<http://www.sun.com/service/sunps/sunone/index.html>
- Sun 企業服務、Solaris 修補程式和支援  
<http://sunsolve.sun.com/>
- 開發人員資訊  
<http://developers.sun.com/prodtech/index.html>

以下位置包含有關 Java ES 及其元件的資訊：

<http://www.sun.com/software/javaenterprisesystem/index.html>

## 連絡 Sun 技術支援

如果您遇到的技術問題在產品文件中沒有解答，則請造訪

<http://www.sun.com/service/contacting>。

## 相關協力廠商網站參照

Sun 不為本文件中所提及之協力廠商網站的可用性負責。對於透過或在此類網站或資源上取得的任何內容、廣告、產品或其他材料，Sun 概不認同，也不承擔責任或義務。對於因使用或依賴此類網站或資源取得的任何內容、商品或服務而造成或聲稱造成的實質或聲稱的損失，Sun 概不承擔責任或義務。

# Sun 歡迎您提出意見

Sun 致力於改善文件品質並歡迎您的批評與指教。

若要提出您的意見，請造訪 <http://docs.sun.com>，並按一下 [ 傳送您的回饋意見 ]。在線上表單中，請提供文件標題和文件號碼。文件號碼位於書本的標題頁或文件的頂部，通常是一組七位或九位數的數字。例如，本書的書名是「Sun Java Enterprise System 2005Q1 部署規劃指南」，且文件號碼為 819-1922。提出意見時您還需要在表格中輸入此文件的英文標題和文件號碼。例如，本文件的英文文件號碼為 819-0058，完整標題為「Sun Java Enterprise System 2005Q1 Deployment Planning Guide」。

Sun 歡迎您提出意見

# 部署規劃簡介

本章提供 Sun Java™ Enterprise System (Java ES) 的概觀，討論部署規劃概念並介紹解決方案生命週期，此週期會概述規劃和設計企業軟體系統的各種步驟。本章包含以下各節：

- 第 17 頁的「關於 Java Enterprise System」
- 第 21 頁的「關於部署規劃」

## 關於 Java Enterprise System

Java Enterprise System 是一套軟體架構，其提供一組完整介體服務支援跨越網路或網際網路環境發佈的企業應用程式。提供服務的 Java Enterprise System 元件皆透過一個公用的安裝程式進行安裝，與一組共用程式庫保持同步，並且共享一個整合的使用者身份識別與安全性管理系統。

### 系統服務

Java Enterprise System 元件所提供的主要基礎架構服務可分類如下：

- **入口網站服務。** 這些服務使得行動辦公員工、在家工作者、知識工作者、業務夥伴、供應商以及客戶，可以從企業網路外部透過網際網路，安全地存取他們的個人化公司入口網站。這些服務向使用者社群提供隨時隨地的存取能力、提供整合、集成、個人化、安全、行動存取以及搜尋功能。
- **通訊與協作服務。** 這些服務允許不同使用者社群之間安全地交換資訊。具體功能包括使用者的業務環境中的訊息傳送、即時協作以及行事曆排程。

- **網路身份識別與安全服務。** 這些服務確保在全域基礎上對所有社群、應用程式和服務實作適當的存取控制策略，以提高企業重要資訊資產的安全性並加強保護。這些服務與儲存庫配合，用於儲存和管理身份設定檔、存取權限以及應用程式和網路資源資訊。
- **Web 與應用程式服務。** 這些服務可讓分散式元件間彼此聯繫，並廣泛地支援伺服器、用戶端和裝置之應用程式的開發、部署以及管理。這些服務以 Java 2 Platform, Enterprise Edition (J2EE™) 技術為基礎。
- **可用性服務。** 這些服務為應用程式和 Web 服務提供近乎不間斷的可用性與延展性。

下表列出了上述的服務類別，並說明為每個類別提供服務的 Java Enterprise System 元件。

**表 1-1** Java Enterprise System 服務類別

服務類別	Java Enterprise System 元件
入口網站服務	Portal Server Portal Server Secure Remote Access Access Manager Directory Server Application Server 或 Web Server
通訊服務和協作服務	Messaging Server Calendar Server Instant Messaging Access Manager Directory Server Application Server 或 Web Server
網路身份識別服務	Access Manager Directory Server Web Server
Web 服務和應用服務	Application Server Message Queue Web Server
可用性服務	Sun Cluster Sun Cluster Agents

如需 Java Enterprise System 服務、元件和 Java Enterprise System 架構概念的更多資訊，請參閱「Java Enterprise System 技術摘要」  
(<http://docs.sun.com/doc/819-1929>)。

## 內建服務和自訂開發服務

以 Java Enterprise System 為基礎的部署解決方案通常分為兩種類別：

- **80:20 部署。** 這些解決方案主要包含 Java Enterprise System 提供的服務。Java Enterprise System 提供 80% 或以上的服務。
- **20:80 部署。** 這些解決方案包含大量的自訂服務和協力廠商應用程式。

80:20 和 20:80 類別包含許多項目。所提供服務類型的確實百分比並不重要。不過，這個百分比可顯示出一個解決方案中所包含自訂的數量。

因為 Java ES 提供豐富的服務組合，所以 Java Enterprise System 非常適合 80:20 部署。例如，相對而言，其可使用 Java Enterprise System 提供的服務，輕易部署整個企業的通訊系統或是整個企業的入口系統。

對於需要自訂開發的部署來說，Java Enterprise System 支援建立及整合自訂開發的服務和應用程式。

[第 18 頁的表 1-1](#) 所列出的多數服務類別都可用於提供 80:20 部署。例如，通訊服務和協作服務為一般使用者提供電子郵件、行事曆和即時訊息服務，讓他們整合及個人化內容。同樣的，網路身份識別和企業入口網站類別的服務可讓您安裝和配置企業範圍內的應用程式，而無須開發或整合自訂服務。

需要自訂開發 J2EE 平台服務的企業解決方案可以調配 Java Enterprise System Web 和應用程式服務提供的 Application Server、Message Queue 或 Web Server。

企業部署所需要的自訂部署服務數量有相當大的變化。因為 Java Enterprise System 服務間的互通功能，您可針對您特定的企業需求建立您自己的服務套件。

## 遷移到 Java Enterprise System

規劃、設計和實作 Java Enterprise System 的企業解決方案，絕大部分取決於目前的部署策略。對於首次規劃部署解決方案的企業來說，其規劃、設計和實作絕大部分是以企業的特定需求為導向的。不過，首次部署解決方案並不是典型的。較典型的情況是解決方案使用 Java Enterprise System 以增強現有企業解決方案或是從舊版 Java Enterprise System 元件升級。

取代或升級現有解決方案時，您必須採取額外的規劃、設計和實作步驟，確保已保留現有資料且將軟體正確升級到目前的版本。當您繼續執行本指南概述的分析和設計時，請注意取代和升級現有軟體系統需要的準備和規劃。

如需有關升級到 Java Enterprise System 目前版本的更多資訊以及從其他應用程式遷移的策略，請參閱「Java Enterprise System 升級與遷移指南」。

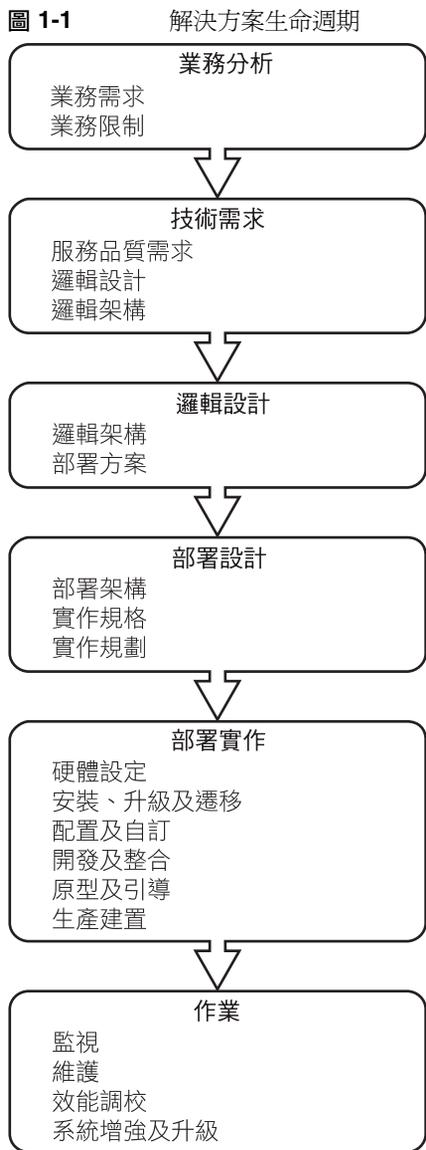
## 關於部署規劃

部署規劃是成功實作 Java Enterprise System 解決方案的關鍵步驟。每個企業都有各自的目標、需求和要考量的優先性。成功的規劃始於分析企業目標和決定達成這些目標的需求。接著必須將業務需求轉換為技術需求，技術需求可作為設計和實作達成企業目標之系統的基礎。

成功的部署規劃是仔細準備、分析和設計的結果。在規劃程序期間所發生的任何錯誤和不當步驟都會導致系統在許多方面無法正確運作。規劃不當的系統可能會造成重大的問題。例如，系統的效能表現不如預期、維護困難、作業成本過高、浪費資源，或是無法調整以符合日益增加的需求。

## 解決方案生命週期

下圖所示的解決方案生命週期描述規劃、設計和實作以 Java Enterprise System 為基礎之企業軟體解決方案的步驟。生命週期對掌握部署專案而言是有用的工具。



生命週期由循序的階段組成。每個階段都由相關的作業組成，其輸出資料會成爲下一個階段的輸入資料。每個階段中的作業都是反覆式的，在產生該階段的輸出資料前需要徹底的分析 and 設計。早期的階段也是反覆式的。例如，在部署設計階段期間，您可能會發現在早期階段的分析不足且需要進行更多作業。

本章的下列章節簡短地描述每個生命週期階段。

## 業務分析階段

在業務分析階段期間，您定義部署專案的業務目標並確定達成這些目標必須符合的業務需要。當確定業務需求時，需考慮任何會影響達成業務目標的業務限制。在整個生命週期期間，您可以根據業務分析階段中執行的分析，來評量部署規劃的可行性以及部署解決方案最終的可行性。

在業務分析階段期間，您需要建立稍後會作為技術需求階段輸入資料的業務需求文件。

如需有關業務分析階段的更多資訊，請參閱第 27 頁的第 2 章「業務分析」。

## 技術需求階段

技術需求階段以您在業務分析階段期間定義的業務需求和業務限制開始，並轉譯這些需求成為之後可用來設計部署架構的技術規格。技術需求指定服務品質 (QoS) 功能，例如效能、可用性、安全性及其他。

在技術需求階段期間，您建立的文件會包含下列資訊：

- 使用者任務和使用模式分析
- 塑造使用者與規劃系統互動的使用實例
- 服務品質需求來自業務需求，可能會考慮使用者任務和使用模式的分析

使用分析、使用實例和 QoS 需求文件的結果系列是解決方案生命週期邏輯設計階段的輸入資料。使用分析也在部署設計階段中扮演重要的角色。

在技術需求階段期間，您也要指定服務層級需求，此為之後建立服務層級合約 (SLA) 的基礎。服務層級合約指定提供客戶系統維護支援的條款，且通常會在部署設計階段中簽訂為專案核准的一部份。

如需有關技術需求的更多資訊，請參閱第 35 頁的第 3 章「技術需求」。

## 邏輯設計階段

在邏輯設計期間，將技術需求階段的使用實例作為輸入資料，以識別實作解決方案所需的 Java Enterprise System 元件。識別支援這些 Java ES 元件的元件，也識別符合業務需求所需的任何額外的自訂開發元件。然後您可以對應邏輯架構中的元件，此架構會顯示元件之間的相互關係。邏輯架構不會指定實作解決方案需要的任何硬體。

邏輯設計階段的輸出資料就是邏輯架構。只有邏輯架構還不足以開始部署設計。您還需要來自技術需求階段的 QoS 需求。來自技術需求階段的邏輯架構和 QoS 需求會形成部署方案。此部署方案是部署設計階段的輸入資料。

如需有關邏輯設計的更多資訊，請參閱第 49 頁的第 4 章「邏輯設計」。

## 部署設計階段

在部署設計期間，您可以將邏輯架構中指定的元件對應到產生高階部署架構的實體環境。您也可以建立實作規格，其提供如何建立部署架構的低階詳細資訊。此外，您建立一系列的規劃和規格，詳細說明實作軟體解決方案的不同層面。

專案核准會在部署設計階段期間發生。在專案核准期間，會評估部署的成本。核准後，就會簽訂實作部署的合約，以及取得建立專案的資源。通常，在詳細說明實作規格後就會核准專案。不過，也可能在完成部署架構時核准專案。

部署階段的輸出資料包括下列項目：

- **部署架構**。一種高階設計文件，表示網路硬體和軟體元件間的對應關係。
- **實作規格**。詳細的規格，可作為建立部署的藍圖。
- **實作規劃**。一組規劃和規格，涵蓋實作企業軟體解決方案的各個層面。實作規劃包括遷移規劃、安裝規劃、使用者管理規劃、測試規劃及其他。

如需有關部署設計的更多資訊，請參閱第 69 頁的第 5 章「部署設計」。

## 部署實作階段

在部署實作階段期間，您可以執行部署設計時所建立的規格與規劃來建立部署架構和實作解決方案。根據您部署專案的特性，此階段會包括以下一部分或是全部的作業：

- 安裝和配置硬體基礎架構
- 安裝和配置軟體
- 建立 LDAP 目錄設計中的使用者和資源的模型
- 根據使用者管理規劃，從現有的目錄和資料庫遷移資料
- 在測試環境中建立和部署引導及原型部署
- 設計並執行功能測試來評量系統需求的合格性
- 設計並執行加強測試來評量尖峰負載下的效能
- 開發及整合任何自訂企業應用程式
- 建立生產部署、逐步執行各階段

一旦部署成形，您必須繼續進行解決方案生命週期的階段作業。

如需有關部署實作階段的更多資訊，請參閱第 101 頁的第 6 章「實作部署設計」。

## 作業階段

作業階段涵蓋讓部署可以正常實作的必要作業。此階段包括下列項目：

- 監視部署以確保系統按規劃執行
- 調校效能以確保部署軟體以最佳狀態執行
- 排定維護作業使作業運作順利，且視需要執行未排程的維護作業
- 視需要升級軟體和硬體

關於作業階段的詳細資訊不屬於本指南的涵蓋範圍。

# 業務分析

在解決方案生命週期的業務分析階段期間定義業務目標，方法是透過分析業務問題以及找出達成該目標的業務需求和業務限制。

本章包含以下各節：

- [第 27 頁的「關於業務分析」](#)
- [第 28 頁的「定義業務需求」](#)
- [第 33 頁的「定義業務限制」](#)

## 關於業務分析

業務分析始於陳述業務目標。接著您要分析業務問題，您必須解決並找出要達到業務目標必須完成的業務需求。同時考慮任何限制您達成目標的業務限制。根據業務需求和限制的分析可產生一組業務需求文件。

您可以將這組業務需求文件作為技術需求階段中產生技術需求的基礎。在整個解決方案生命週期間，您可以根據業務分析階段中執行的分析，來衡量部署規劃的可行性以及解決方案最終的可行性。

# 定義業務需求

業務需求定義並沒有簡單的公式。業務需求的決定應根據與軟體解決方案利害關係人的合作、您對業務網域的相關知識以及實用的創意思考。

本節提供定義業務需求時應考慮的幾個因素。

## 設定業務目標

業務分析應該說明部署專案的目標。清楚的目標有助於集中設計判斷，亦可避免專案偏離主題。用目前的作業對照業務目標可協助決定設計的判斷。

### 範圍

業務需求應該要陳述部署專案的範圍。確定您定義的範圍可以被處理，並且避免「無限制」的敘述使目標不清楚或無法達成。不清楚的定義範圍會導致部署設計無法充分滿足業務需求或是浪費資源。

### 優先性

排出目標的優先順序，來確保部署最重要的層面可以最先達成。如果資源有限時，則可能需要延遲或修改某些目標。一般而言，大型複雜的部署需要階段性實作解決方案。透過確定優先性，您可以提供決策（可能需要為您的部署設計建立以便獲得利害關係人承認）的指導。

### 關鍵特性

找出成功的關鍵領域，以便利害關係人和設計者可以將重心放在在最重要的標準。

### 成長因素

設定業務目標時，您不只要考慮組織目前的需要，也要預估這些需求長期下會如何變更和成長。您應該避免決定一個很快就不適用的解決方案。

### 安全空間

解決方案的設計以業務分析階段期間所做的假設為基礎。有許多原因可能造成這些假設不正確，例如，資料不足、判斷錯誤，或是未預期的外部事件。請確保您同時在業務目標和整個規劃中進行安全空間規劃，如此，您所設計的解決方案才能處理未預期的事件。

## 瞭解使用者需求

進行必要的研究，以瞭解解決方案鎖定的使用者類型、他們的需求以及他們的預期利益。例如，下列清單提供一個分類使用者的方法：

- 僅限於現任員工
- 現任和離職員工
- 管理員
- 活躍的客戶
- 所有的客戶
- 會員場所
- 公眾
- 限制存取

清楚指出使用者的預期利益可幫助促成設計決定。例如，這裡是解決方案可提供給使用者的一些好處：

- 遠端存取公司資源
- 企業合作
- 單純化日常作業
- 由遠端團隊分享資源
- 增加產能
- 一般使用者的自我管理

## 開發作業需求

將作業需求以一組包含明確目標的功能需求來表示。一般而言，您會為下列的區域建立作業需求，例如：

- 一般使用者功能
- 減少回應時間
- 可用性和正常執行時間
- 減少錯誤率
- 資訊存檔和保留

以所有利害關係人都能瞭解的適當術語來表達作業需求。避免含糊不清的語言，例如「適當的一般使用者回應時間」。作業需求的範例如下所示：

- 電源中斷時，在 10 分鐘之內修復服務的能力
- 重新傳送過去 48 小時的傳入訊息的能力
- 在尖峰期間，在 60 秒內完成線上交易
- 在尖峰期間，在 4 秒內完成一般使用者認證

## 支援現有的使用模式

以可清楚評量的目標表示現有的使用模式。下列問題可協助決定此類的目標。

- 如何使用目前的服務？
- 使用模式為何（例如，偶爾使用、時常使用或大量使用）？
- 您的使用者通常會連線到哪個網站？
- 使用者通常傳送的訊息大小為何？
- 使用者每日或每小時通常會完成多少交易？

研究會存取服務的使用者。使用者何時會存取現有的服務及存取時間長度，此類的因素是識別目標的關鍵。如果組織的經驗無法提供這些模式，請研究類似組織的經驗。

## 瞭解公司文化

需求分析應該將公司文化和政策的各種層面考慮在內。如果不注意公司文化會導致解決方案不容易被接受，或是難以實作。

### 利害關係人

確認在提議解決方案的成功當中具有既得利益的個人和組織。所有的利害關係人都應該主動參加業務目標和需求的定義。如果利害關係人沒有參加或是對規劃的變更一無所知，則規劃可能會有很嚴重的缺失。此類的利害關係人甚至會阻礙部署的實作。

### 標準和策略

確定您瞭解組織要求解決方案的標準和策略。這些標準和策略可能會影響設計、產品選項和部署方法的技術面。

一個範例是人事資料的機密性，這可能是由人事資源組織或部門主管所擁有或控制的資訊。另一個範例則是改變經營的公司程序。改變經營的公司程序可能會嚴重地影響解決方案的接受度、實作方法和時間表。

### 規管需求

規管需求有很大的程度是取決於業務的特性。研究和瞭解可能會影響部署的規管需求。許多公司和政府機構需要遵循存取標準。部署全球性解決方案時，亦需考慮到外國的法律和法規。例如，許多歐洲國家對儲存個人資訊有嚴格的控制。

### 安全性

您之前確認的目標，可能有需要特別注意的潛在安全性問題。調出部署所需的特定安全性目標。例如：

- 專屬資訊的授權存取
- 以角色架構存取機密資訊
- 遠端位置間的安全通訊
- 在本機系統呼叫遠端應用程式
- 與協力企業和組織安全交易
- 強制執行安全性策略

## 站點分散

站點間的地理分散與頻寬會影響設計決定。此外，某些站點可能需要本機管理。

這些地理考量會提高專案的訓練成本和複雜性等等。清楚指出站點地理分散造成的需求。強調哪些站點是設計成功的關鍵。

## 使用遞增方式

通常您會將軟體解決方案視為一個全體、廣泛的系統。然而，您經常以評量步驟逐步達成完整系統的部署。

當採用遞增方式時，您通常會設計藍圖來提供引導到最終、廣泛解決方案的重大事件。此外，您可能要考慮延遲到後期實作的廣泛解決方案層面的短期規劃。

遞增方式提供這些優點：

- 您可以配合業務成長帶來的需求變更。
- 您可以調配現有的架構作為最終部署實作前的過渡階段。
- 您可以調度資本支出需求。
- 您可以調配小型的人力資源供給。
- 您可以考慮合夥可能。

## 瞭解服務層級合約

服務層級合約 (SLA) 指定最低的效能需求以及依據達成那些需求造成的失敗指定必須提供的客戶支援的層級和範圍。服務層級合約 (SLA) 是以業務分析期間定義的業務需求為基礎，稍後會將這些需求指定為技術需求階段期間的服務層級需求。SLA 會在專案核准 (也就是部署設計階段) 期間簽訂。

您應該以下列項目為中心開發 SLA，例如正常執行時間、回應時間、訊息傳送時間和損壞修復。SLA 應該考慮這些項目，例如，系統概觀、支援組織的角色和責任、如何衡量服務層級、改變要求等等。以系統可用性為中心來識別組織的預期是決定 SLA 範圍的關鍵。

## 定義業務限制

業務限制在決定部署專案的特性中扮演重要的角色。成功的部署設計關鍵在於在已知業務限制中找到符合業務需求的最佳方式。業務限制可以是財務限制、實體限制（例如網路容量）、時間限制（例如完成重大事件（如下次年度會議）之前），或是任何其他限制（您預期會對完成業務目標造成影響的因素）。

本節提供定義業務限制時要考慮的幾個因素。

## 遷移問題

通常，部署專案會取代或補充現有的軟體基礎架構和資料。任何新的解決方案必須能夠將資料和程序從現有基礎架構遷移到新的解決方案，通常會保留與現有應用程式的互通功能。需要目前基礎架構分析，才能決定遷移問題在提議解決方案中的影響範圍。

## 排程指定作業

解決方案的實作排程會影響設計決定。積極的排程可能會導致目標的縮小、改變優先性或是採取遞增的解決方法。在排程中，可能也有重要的重大事件值得考慮。重大事件可以以內部事件（例如排程的服務建置），或是外部事件（例如學校學期的開學日）來設定。

## 預算限制

大部分的部署專案必須依附預算。考慮建立提議解決方案的成本，還有在特定的使用期間內維護專案需要的資源包括下列項目：

- **現有硬體和網路基礎架構。** 信賴現有基礎架構會影響系統設計。
- **實作解決方案需要的開發資源。** 有限的開發資源，包括硬體、軟體和人力資源，可能會建議遞增式部署。您可能要對每個您所實作的遞增階段重新使用相同的資源或開發團隊。
- **維護、管理和支援。** 分析在系統上管理、維護和支援使用者可用的資源。有限的資源可能會影響設計決定。

## 所有權成本

除了維護、管理和支援，請分析其他可能會影響所有權成本的因素。可能需要的硬體和軟體升級，對電源供應網上的解決方案的影響，電信成本和其他影響暫墊費用的因素。指定解決方案可用性層級的服務層級合約也會因為要求較多的備援而影響所有權的成本。

解決方案的實作應該提供投資利潤。投資利潤的分析通常包含資金支出獲得的經濟利益。

估計解決方案的經濟利益包含要達成的目標與達成目標的替代方案相比較，或是與甚麼都不做的成本相比較的詳細分析。

# 技術需求

在解決方案生命週期的技術需求階段，您會執行使用分析、識別使用實例，以及決定提議部署解決方案的服務品質需求。

本章包含以下各節：

- 第 35 頁的「關於技術需求」
- 第 36 頁的「使用分析」
- 第 38 頁的「使用實例」
- 第 38 頁的「服務品質需求」
- 第 47 頁的「服務層級需求」

## 關於技術需求

技術需求分析以解決方案生命週期的業務分析階段期間所建立的業務需求文件開始著手進行。使用業務分析為基礎，執行以下步驟：

- 執行使用分析來協助決定預期的負載。
- 建立塑造典型使用者與系統互動的使用實例。
- 建立一組服務品質需求 (QoS)，以定義部署解決方案在回應時間、可用性、安全性及其他領域中應有的表現。

服務品質需求來自使用分析和使用實例，請記住先前已確認業務需求和限制。

稍後，會在邏輯設計階段中將服務品質需求與邏輯架構搭配以形成部署方案。部署方案是解決方案生命週期之部署設計階段的主要輸入資料。

在業務分析中，對於技術需求分析而言，並沒有產生使用分析、使用實例與系統需求的簡單公式。技術需求分析需要瞭解業務領域、業務目標以及系統科技的構成。

## 使用分析

使用分析包含確認您設計的解決方案的各種使用者以及決定使用者的使用模式。您所收集的資訊是估計系統負載情況的基礎。使用分析資訊在分配使用實例的權重時也非常的有用，如第 38 頁的「使用實例」中所述。

在使用分析期間您應該盡可能地約談使用者，研究使用模式現存的資料，以及約談設計者和之前系統的管理員。下表列出執行使用分析時應該考慮到的因素。

**表 3-1** 使用分析因素

主題	描述
使用者的數量與類型	<p>決定您的解決方案必須支援多少使用者，並在需要時分類使用者。</p> <p>例如：</p> <ul style="list-style-type: none"> <li>「企業到消費者 (B2C)」解決方案可能有大量的訪問者，不過只有少數的使用者會登記並從事商務交易。</li> <li>「企業到員工 (B2E)」解決方案典型上可容納每一個員工，不過有些員工可能需要從公司網路外部進行存取。</li> </ul> <p>在 B2E 解決方案中，負責人可能需要得到一般員工不能存取的特定區域授權。</p>
作用中和非作用中的使用者	<p>確認使用模式以及作用中和非作用中使用者的比例。</p> <p>作用中的使用者指的是登入系統並且與系統服務互動的使用者。非作用中的使用者則可能是未登入系統的使用者、登入系統但沒有與系統元件互動的使用者，或是在資料庫中卻從未登入的使用者。</p>
管理使用者	<p>確認能進入部署系統監視、更新以及支援部署的使用者。</p> <p>決定任何會影響技術需求的特定管理使用模式 (例如防火牆外部的部署管理)。</p>

表 3-1 使用分析因素 (續)

主題	描述
使用模式	<p>確認不同類型的使用者如何存取系統並提供目標給預期的使用。</p> <p>例如：</p> <ul style="list-style-type: none"> <li>• 使用錯誤時是否有尖峰時段？</li> <li>• 一般業務時段為何？</li> <li>• 使用者是否分散全球？</li> <li>• 使用者連線的預期持續期間為何？</li> </ul>
使用者成長	<p>確定使用者基礎規模是否固定或者部署是否預期使用者數量的成長。</p> <p>如果預期使用者基礎將會成長，試著針對此一成長做出合理的規劃。</p>
使用者交易	<p>確認必須支援的使用者交易類型。這些使用者交易可以轉化成使用實例。</p> <p>例如：</p> <ul style="list-style-type: none"> <li>• 使用者會執行什麼作業？</li> <li>• 使用者何時登入，他們會維持登入狀態嗎？或他們只是執行一些作業後就登出？</li> <li>• 使用者之間是否需要一般的行事曆、網路會議及內部網頁的部署等值得注意的協作？</li> </ul>
使用者研究及統計資料	<p>利用既有的使用者研究及其他來源來決定使用者的行為模式。</p> <p>通常企業或工業組織擁有使用者的研究資料，從這裡您可以取得關於使用者的有用資訊。現有應用程式的記錄檔可能包含可用於評估系統的有用統計資料。</p>

## 使用實例

使用實例以您設計的解決方案塑造出典型的使用者互動，從一般使用者的角度描繪出作業的完整流程。決定完整使用實例的設計優先順序可確保持續集中焦點在達成預期功能上。使用實例是邏輯設計的主要輸入資料。

指定相對的權重給使用實例，權重最高的使用實例代表最常見的使用者作業。使用實例的權重可讓您將設計的重點放在最常使用的系統服務上。

使用實例可描述為兩種層級。

- **使用實例報告。** 個別實例的說明，包括主要和替代的事件流程。
- **使用實例圖。** 使用實例圖描述行為者和使用實例之間的關係、並以較正式的編排方式來表示事件流程。使用實例圖例有助於塑造較大型或複雜的使用實例。一般而言，您可以使用統一塑模語言 (UML) 標準來描繪使用實例圖。

## 服務品質需求

服務品質需求 (QoS) 是技術規格，可指定系統功能品質，例如，效能、可用性、延展性及服務性。QoS 需求以業務需求中指定的業務需要為導向。例如，如果這些服務必須 24 小時全天候提供，可用性需求就必須解決這項業務需求。

下表列出通常會形成 QoS 需求基礎的系統品質。

表 3-2 系統品質影響 QoS 需求

系統品質	描述
效能	衡量相對於使用者負載條件的回應時間和處理能力。
可用性	衡量一般使用者可存取系統資源與服務的頻率，常以系統的 <b>正常執行時間</b> 表示。
延展性	隨時間推移在已部署系統中新增容量（及使用者）的能力。延展性通常牽涉在系統中新增資源，但不應要求對部署架構做出變更。
安全性	描述系統及其使用者的完整性之複雜因子組合。安全性包括使用者的認證和授權、資料安全性以及部署系統的安全存取。
潛在容量	系統在不新增資源的情況下處理少見的尖峰負載的能力。潛在容量是可用性、效能和延展性品質中的一個因素。
服務性	對已部署系統進行維護的易行度，維護包括系統監視、修復發生的問題及升級硬體與軟體元件等作業。

系統品質彼此息息相關。對某個系統品質的需求可能會影響對其他系統品質的需求與設計。例如，較高層級的安全性可能會影響效能，而效能又會影響可用性。新增額外伺服器來解決可用性問題卻可能影響到服務性（維護成本）。

瞭解系統品質如何產生關聯以及其利弊，是設計能夠成功滿足業務需求和業務限制的系統最大的關鍵。

下節更進一步描述影響部署設計的系統品質，提供公式化 QoS 需求時的因素指導。關於服務層級需求的章節，其中亦包含形成服務層級合約的基礎。

## 效能

業務需求通常以指定回應時間的非技術名詞表示效能。例如，以 Web 架構存取的業務需求可能強調以下：

使用者在登入時可預期合理的回應時間，通常不會大於四秒。

以業務需求開始，檢查所有使用實例來決定如何在系統層級表達此需求。在某些情況下，您可能想要包含在使用分析期間所決定的使用者負載條件。以特定負載條件下的回應時間或是回應時間加上處理能力來表示每個使用實例的效能需求。您還可以指定可容許的錯誤數量。

這是兩個如何指定用於效能的系統需求範例：

當日網頁重新整理的回應不得大於四秒，每隔 15 分鐘進行取樣一次，每百萬次交易要少於 3.4 個錯誤。

在定義的尖峰期間內，系統必須每秒允許 25 個安全登入，且任何使用者的回應時間不得大於 12 秒，每百萬次交易要少於 3.4 個錯誤。

效能需求與可用性需求（故障轉移如何影響效能）以及潛在容量（處理不尋常尖峰負載的容量有多少）息息相關。

## 可用性

可用性是指定部署系統正常執行時間的一種方式，且通常以系統可供使用者存取時間的百分比為測量基準。系統無法存取的時間（當機時間）可能歸咎於硬體、軟體、網路錯誤或者其他造成系統當機的因素（例如斷電）。規劃的服務當機時間（維護和升級）不列入當機時間計算。以正常執行時間的比率計算系統可用性的基本方程式為：

$$\text{可用性} = \text{正常執行時間} / (\text{正常執行時間} + \text{當機時間}) * 100\%$$

通常您會用可以達成的「九」的數量來評量可用性。例如，99% 的可用性為兩個「九」。指定額外的「九」會明顯地影響部署設計。下表顯示增加可用性額外的「九」到系統的未排程當機時間，系統為一整年 24x7 小時執行，合計為 8760 小時。

**表 3-3** 系統執行一整年 (8760 小時) 的未排程當機時間

「九」的數量	可用百分比	未排程的當機時間
2	99%	88 小時
3	99.9%	9 小時
4	99.99%	45 分鐘
5	99.999%	5 分鐘

## 容錯系統

四或五個「九」的可用性需求通常需要容錯系統。容錯系統必須能夠在硬體或軟體錯誤期間繼續服務。通常，容錯是透過硬體 (例如 CPU、記憶體和網路裝置) 和軟體兩者的備援提供關鍵服務。

錯誤單點是一個硬體或軟體元件，為重要路徑的一部分，但沒有透過備援元件備份。此元件的錯誤會導致系統服務的損失。在設計容錯系統時，您必須確認並排除潛在的錯誤單點。

容錯系統的實作和維護可能很昂貴。確定您瞭解用於可用性的業務需求特性，並考慮符合這些需求的可用性解決方案的策略和成本。

## 排定服務可用性優先順序

從使用者觀點來看，比起整體系統的可用性，可用性更常應用在個別的服務上。例如，無法使用即時訊息傳送服務，通常不會對其他服務的可用性造成太大的影響。然而，如果是無法使用其他許多服務所依賴的服務 (例如 Directory Server)，對系統就會有較大的影響。較高的可用性規格應該清楚地參照到需要較高可用性的特定使用實例和使用分析。

根據優先性的順序組合列出可用性需求會很有幫助。下表排出服務的不同類型的可用性順序。

表 3-4 依優先性排列的服務可用性

優先性	服務類型	描述
1	關鍵任務	必須隨時能用的服務。例如，應用程式的資料庫服務（如 LDAP 目錄）。
2	必須可用	必須能用的服務，但是在降低效能時可以利用。例如，訊息傳送可用性在某些業務環境中可能不是關鍵的。
3	可以延遲	在給定期間內可以使用的服務。例如，行事曆服務可用性在某些業務環境中可能不是必要的。
4	選擇性	可以無限延遲的服務。例如，在某些環境中，即時訊息傳送服務可能被視為有用但不是必要的服務。

## 服務損失

可用性設計會考慮可用性降低或元件遺失時可能發生的情況。同時也會考慮相關的使用者是否必須重新開始階段作業，以及一個區域的錯誤對系統其他區域所造成的影響程度。QoS 需求應該要考慮這些方案，並指定部署如何對這些解決方案做出回應。

## 延展性

延展性是增加容量到您系統的能力，如此系統才能支援來自現有使用者或增加的使用者基礎的額外負載。延展性通常需要額外的資源，但是不應該需要變更部署架構的設計，或是因為需要加入額外資源的時間而損失服務。

如同可用性，延展性更多應用在系統提供的個別服務上而不是整個系統。然而，在其他許多服務所依賴的服務之上（例如 Directory Server），延展性可能會影響整個系統。

您不需要以 QoS 需求來指定延展性需求，除非部署的計畫性成長清楚地說明在業務需求中。在解決方案生命週期的部署設計階段期間，部署架構應該有增加調整系統規模的容錯比率，即使您未指定延展性的 QoS 需求。

## 估計成長

評估系統的成長以決定延展性需求包含預測、評估和可能無法滿足的假設。可延展系統之開發需求的三個關鍵，如下所示。

- **高效能設計策略。** 在效能需求的規格期間，納入潛在容量處理日後可能增加的負載。同時，在預算限制內將可用性最大化。這項策略可讓您吸收成長並提高調整系統規模的排程重大事件。
- **遞增式部署。** 遞增式部署可協助排程額外的資源。指定調整系統的清清楚重大事件。重大事件通常是會與特定日期相互協調以評估延展性的負載型需求。
- **大規模的效能監視。** 監視效能可協助決定何時要新增資源到部署中。監視效能需求可為負責維護及升級的作業人員和管理員提供指導。

下表列出在決定延展性需求時要考慮的因素。

**表 3-5** 延展性因素

主題	描述
分析使用模式	透過研究現有資料，瞭解目前 (或是計畫) 使用者基礎的使用模式。缺少目前的資料時，分析工業資料或市場估計。
合理的最大規模設計	為已知和可能的需要設計朝向最大需求規模的目標。 通常，根據現有使用者負載和未來負載的合理預期，這會是一個 24 個月的評估。估計的時期絕大部分依賴設計的可信度。
設定適當的重大事件	遞增式實作部署設計來符合短期需求，並使用緩衝以允許預期外的成長。設定增加系統資源的重大事件。 例如： <ul style="list-style-type: none"> <li>• 取得資金 (例如季度或年度)</li> <li>• 購買硬體和軟體的前置時間 (例如一到六個星期)</li> <li>• 緩衝 (10% 到 100%，根據成長預期)</li> </ul>
吸收新興技術	瞭解新興技術，例如較快的處理器和 Web 伺服器，以及這項技術會如何影響基礎架構的效能。

## 安全需求

安全性是一個複雜的主題，與部署系統的所有層級相關。開發安全性需求會以識別安全性威脅和開發對抗策略為中心。此安全性分析包括下列步驟：

1. 識別重要資產
2. 識別那些資產的威脅
3. 識別會造成這些威脅的漏洞，這些威脅會對組織形成風險
4. 開發可減輕組織風險的安全性規劃

安全性需求的分析應該要包含組織內各種利益關係人，包括經理、業務分析師和資訊技術專業人員。通常，組織會指定安全性架構師領導安全性措施的設計和實作。

下列章節描述安全性規劃所涵蓋的區域。

## 安全性規劃的元素

系統的安全性規劃是部署設計的一部份，是成功實作不可或缺的要素。規劃安全性時，請考慮下列項目：

- **實體安全性。** 實體安全性是對路由器、伺服器、伺服器機房、資料中心和基礎架構其他部分的實體存取。如果未經授權的人可以進入伺服器機房和拔除路由器，則其他的安全性措施就會遭到危害。
- **網路安全性。** 網路安全性是透過防火牆、安全存取區域、存取控制清單和連接埠存取，存取您的網路。為求網路安全性，您開發未經授權存取、篡改和拒絕 (DoS) 服務攻擊的策略。
- **應用程式和應用程式資料安全性。** 應用程式和應用程式資料安全性包含透過認證及授權程序和策略，對使用者帳戶、公司資料和企業應用程式的存取 此區域包括定義下列策略：
  - 密碼策略
  - 存取權限，例如相對於管理員存取的使用者委託管理
  - 帳戶關閉
  - 存取控制
  - 加密策略，包含資料的安全傳輸以及使用認證簽定資料
- **個人安全性措施。** 整個組織的安全性策略定義作業環境以及所有使用者必須遵守的措施，藉以確定安全性措施將如同設計運作。通常，您會開發關於安全性指南或手冊，並為使用者提供安全性措施的訓練。為了取得有效的整體安全性策略，良好的安全性措施必須成為組織文化的一部份。

## 潛在容量

潛在容量是部署處理不尋常尖峰負載使用而無須額外資源的能力。一般而言，您不會直接根據潛在容量指定 QoS 需求，但是這項系統品質是決定可用性、效能和延展性需求的因素。

## 可服務性需求

可服務性是維護已部署系統的輕易程度，包括監視系統、修復產生的問題、從系統新增和移除使用者以及升級硬體和軟體元件的作業。

當規劃可服務性需求時，請考慮下表列出的主題。

**表 3-6** 可服務性需求的主題

主題	描述
當機時間規劃	<p>確認需要特定服務無法使用或是部分無法使用的維護作業。</p> <p>在其他需要中斷對使用者的服務時，有些維護和升級可以不間斷地進行。如果有可能，請將維護活動時需要當機時間的使用者納入排程，讓使用者規劃當機時間。</p>
使用模式	<p>識別使用模式以決定排程維護的最佳時間。</p> <p>例如，在尖峰使用為一般業務時間的系統上，將維護作業排定在晚上或週末。對於地理分散的系統來說，決定這些時間可能非常具有挑戰性。</p>
可用性	<p>可服務性經常是您可用性設計的反映。維護和升級的最小化當機時間策略以您的可用性策略為中心。需要高度可用性的系統進行維護、升級和修復的機會有限。</p> <p>處理可用性需求的策略影響您處理維護和升級的方式。例如，在地理上分散的系統，服務在維護期間可以依靠傳送工作量到遠端伺服器的能力。</p> <p>同時，需要高度可用性的系統可能需要更多成熟的解決方案，使用少量人力介入的系統就可以自動重新啟動。</p>
診斷與監視	<p>透過定期執行診斷和監視工具來確認問題區域，您可以改善系統的穩定性。</p> <p>定期監視系統可以避免問題發生，根據可用性策略幫助平衡工作量並改善維護和當機時間規劃。</p>

## 服務層級需求

服務層級合約 (SLA) 指定最低的效能需求以及依據達成那些需求造成的失敗指定必須提供的客戶支援的層級和範圍。服務層級需求是系統需求，會指定以 SLA 為基礎的情況。

如同 QoS 需求，服務層級需求來自業務需求，並代表關於部署系統必須達到的整體系統品質的擔保。因為服務層級合約被視為合約，所以服務層級需求的規格必須相當明確。服務層級需求精確定義何種狀況下測試需求，以及明確地說明何種組成不滿足需求。



# 邏輯設計

在解決方案生命週期的邏輯設計階段期間，您會設計邏輯架構，顯示解決方案邏輯元件的相互關係。來自技術需求階段的邏輯架構和使用分析會形成部署方案，其為部署設計階段的輸入資料。

本章包含以下各節：

- [第 49 頁的「關於邏輯架構」](#)
- [第 50 頁的「設計邏輯架構」](#)
- [第 58 頁的「範例邏輯架構」](#)
- [第 65 頁的「存取區域」](#)
- [第 67 頁的「部署方案」](#)

## 關於邏輯架構

邏輯架構會識別實作解決方案需要的軟體元件，顯示元件之間的相互關係。技術需求階段期間決定的邏輯架構和服務品質需求會形成部署方案。部署方案是設計部署架構的基礎，會在下個階段（亦即部署設計）發生。

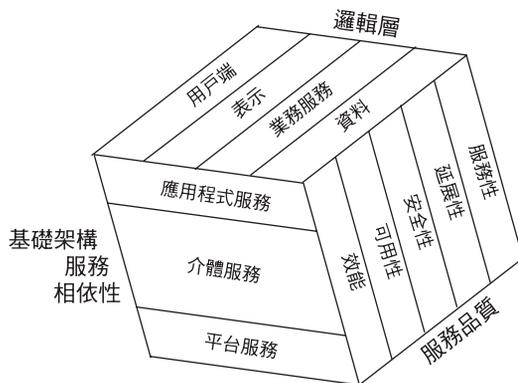
開發邏輯架構時，您不但要識別為使用者提供服務的元件，還要識別那些提供必要介體和平台服務的其他元件。基礎架構服務相依性和邏輯層提供兩種互補的方法來執行這個分析。

基礎架構服務相依性和邏輯層是解決方案架構三個要素中的其中兩個要素，該架構為 Sun Java™ Enterprise System 的基礎。三個要素如下所示，在[第 50 頁的圖 4-1](#) 也有介紹。

- **基礎架構服務相依性。** 與提供企業服務的軟體元件互動。軟體元件需要一套基本的基礎架構服務，讓分散式元件可以彼此通訊以及交互操作。

- **邏輯層**。軟體元件的邏輯組織，分屬代表軟體元件邏輯和實體獨立性的階層，以其提供的服務特性為基礎。
- **服務品質**。系統服務品質，例如，效能、可用性、延展性和其他代表軟體解決方案設計和操作特定層面的特性。

圖 4-1 Java Enterprise System 解決方案架構的三個要素



**備註** 如需 Java Enterprise System 架構觀念的更多資訊，請參閱「Java Enterprise System 技術摘要」(<http://docs.sun.com/doc/819-1929>) 的「Java Enterprise System 架構」一章。

邏輯架構藉由顯示必要的元件及其相依性，來描述基礎架構服務層級。邏輯架構也會將元件分配於邏輯層之間，這些邏輯層代表最終可由用戶端層存取的表示、業務和資料服務。服務品質需求並不是在邏輯架構中成形，而是與部署方案的邏輯架構相互搭配。

## 設計邏輯架構

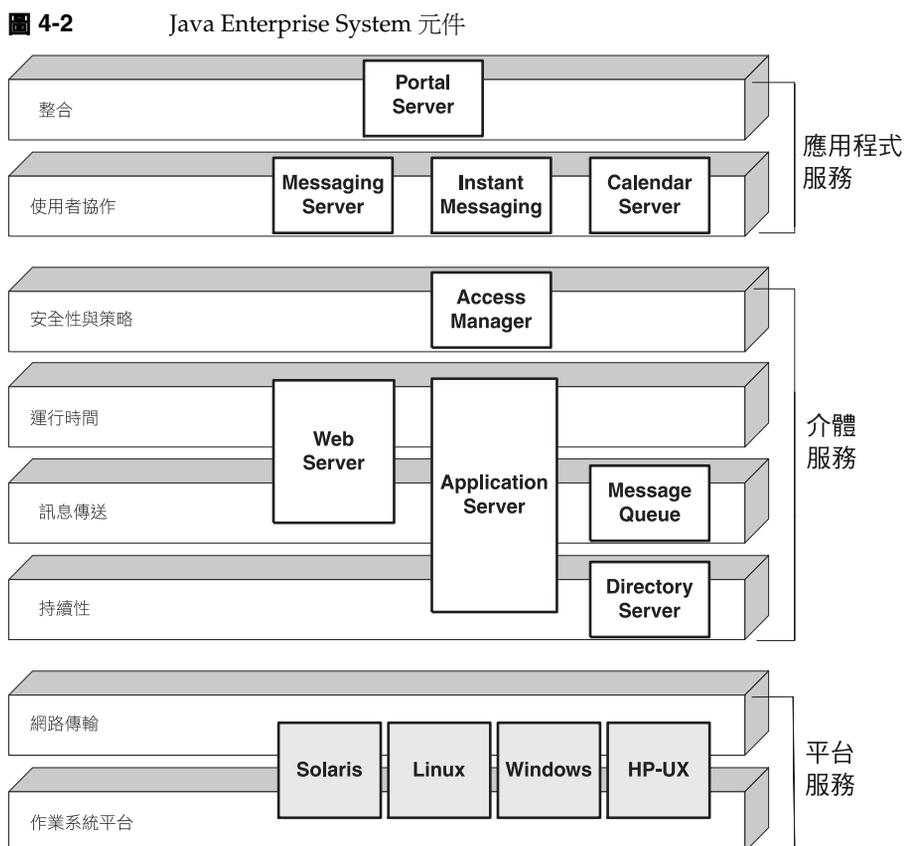
設計邏輯架構時，請使用技術需求階段期間所識別的使用實例，以決定可為解決方案提供必要服務的 Java Enterprise System 元件。您也必須識別可為最初識別之元件提供服務的任何元件。

根據元件所提供的服務類型，將 Java Enterprise System 元件放置在多層架構的環境中。瞭解元件是多層架構的一部份，可協助您稍後決定如何分配元件提供的服務，也可協助決定實作服務品質的策略（例如延展性、可用性及其他）。

此外，您可以以其他觀點檢視邏輯元件，將元件放置在安全存取區域內。[第 65 頁的「存取區域」](#)一節提供安全存取區域的範例。

## Java Enterprise System 元件

Java Enterprise System 由互動式軟體元件組成，提供您可用來建立企業解決方案的企業服務。下圖顯示隨附於 Java Enterprise System 的關鍵軟體元件。「Java Enterprise System 技術摘要」(<http://docs.sun.com/doc/819-1929>) 提供 Java Enterprise System 元件及其提供之服務的其他資訊。



## 元件相依性

識別邏輯架構的 Java Enterprise System 元件時，您也必須識別支援的元件。例如，如果您將 Messaging Server 視為邏輯架構的必要元件，則您的邏輯架構也必須包括 Directory Server 和 Access Manager ( 若有可能 )。Messaging Server 依賴目錄服務的 Directory Server 並依賴需要單次登入之解決方案的 Access Manager。

下表列出 Java Enterprise System 元件的相依性。請參閱第 54 頁的圖 4-3，此圖以視覺方式表示關鍵元件彼此間的相依性。設計邏輯架構時，使用這個表格及隨附的圖表來決定設計中的附屬元件。

**表 4-1** Java Enterprise System 元件相依性

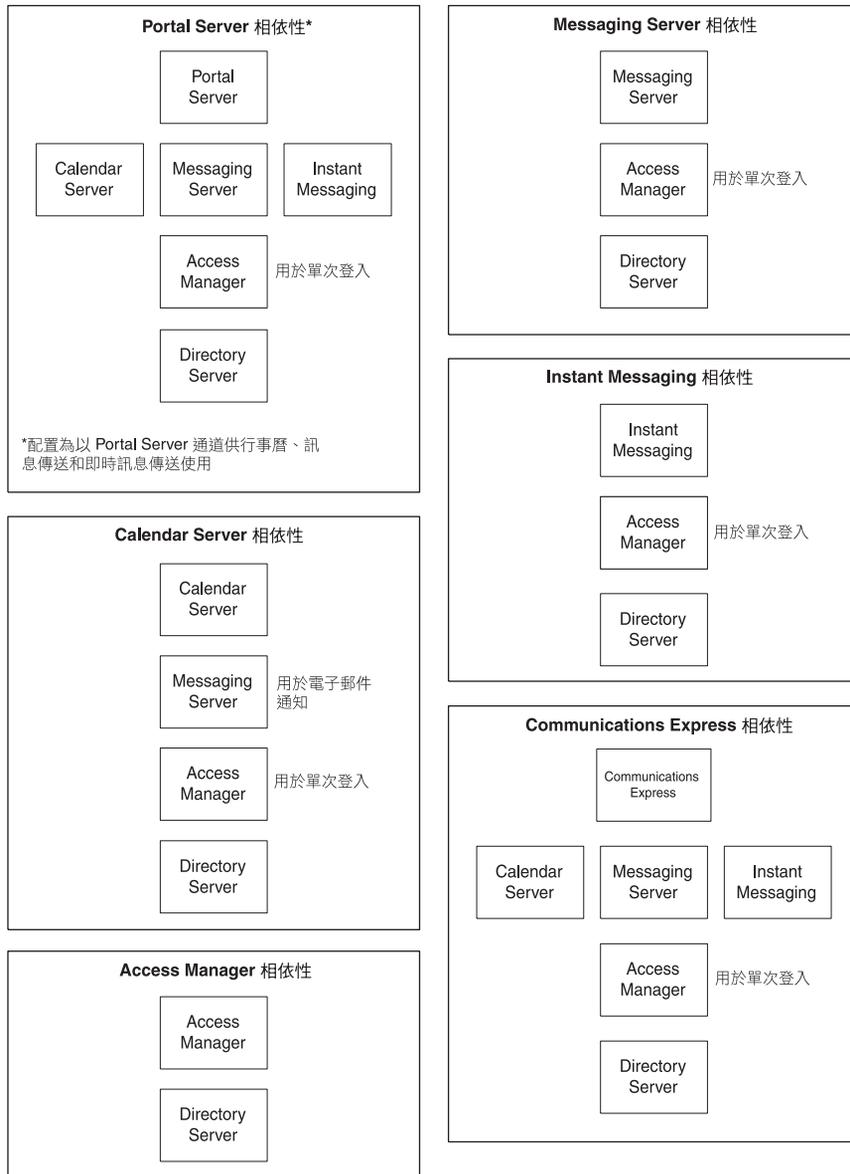
Java Enterprise System 元件	依賴
Application Server	Message Queue Directory Server ( 選擇性 )
Calendar Server	Messaging Server ( 用於電子郵件通知服務 ) Access Manager ( 用於單次登入 ) Web Server ( 用於 Web 介面 ) Directory Server
Communications Express	Access Manager ( 用於單次登入 ) Calendar Server Messaging Server Instant Messaging Web Server ( 用於 Web 介面 ) Directory Server
Directory Proxy Server	Directory Server
Directory Server	無
Access Manager	Application Server 或 Web Server Directory Server
Instant Messaging	Access Manager ( 用於單次登入 ) Directory Server
Message Queue	Directory Server ( 選擇性 )
Messaging Server	Access Manager ( 用於單次登入 ) Web Server ( 用於 Web 介面 ) Directory Server
Portal Server	如果配置為使用 Portal Server 通道： Calendar Server Messaging Server Instant Messaging  Access Manager ( 用於單次登入 ) Application Server 或 Web Server Directory Server
Portal Server Secure Remote Access	Portal Server

**表 4-1** Java Enterprise System 元件相依性 (續)

Java Enterprise System 元件	依賴
Web Server	Access Manager (選擇性，用於存取控制)

**備註** 在表 4-1 中所列出的 Java Enterprise System 元件之間的相依性並未涵蓋所有的元件相依性。表 4-1 沒有列出您在規劃安裝時必須考慮的相依性。如需完整的 Java Enterprise System 相依性的清單，請參閱「Java Enterprise System 安裝指南」(<http://docs.sun.com/doc/819-0811>)。

圖 4-3 Java Enterprise System 元件相依性



## Web 容器支援

前一節 ( [元件相依性](#) ) 沒有考慮到 Portal Server 和 Access Manager 在其中執行的 Web 容器。此 Web 容器可能是由 Application Server、Web Server 或協力廠商產品所提供。設計包含 Portal Server 或 Access Manager 的邏輯架構時，請確實考慮這些元件需要的 Web 容器。

## Messaging Server 提供的不同邏輯的服務

可將 Java Enterprise System Messaging Server 配置為提供個別的實例，這些實例會提供下列不同邏輯的服務：

- Message Transfer Agent (MTA)
- Message Multiplexor
- Message Express Multiplexor
- 訊息儲存

Messaging Server 的各種配置提供的功能可部署在個別實體伺服器上，也可以在邏輯架構的不同層上加以表示。因為這些 Messaging Server 的配置代表個別層上不同邏輯的服務，在設計邏輯架構時請將這些配置視為不同邏輯的元件。[第 58 頁的「範例邏輯架構」](#)一節提供邏輯獨立元件的範例。

下表描述 Messaging Server 不同邏輯的配置。

**表 4-2** Messaging Server 配置

子元件	描述
Message Transfer Agent (MTA)	可支援電子郵件的傳送，方法是透過處理 SMTP 連線、路由電子郵件以及將訊息傳遞到適當的訊息儲存區。可將 MTA 元件配置為支援從企業外部傳送的電子郵件 (內送)，或是從企業內部傳送的電子郵件 (外傳)。
訊息儲存 (STR)	提供電子郵件的擷取和儲存。
Message Multiplexor (MMP)	可支援電子郵件的擷取，方法是使用 IMAP 或 POP 協定，來存取電子郵件用戶端的訊息儲存。
Messenger Express Multiplexor (MEM)	支援電子郵件的擷取，方法是代表 Web 型 (HTTP) 用戶端存取訊息儲存。

## 存取元件

Java Enterprise System 也包含提供系統服務存取的元件，通常是來自企業防火牆外部。Messaging Server 的某些配置也會提供網路存取，例如為訊息多重訊號組合器配置的 Messaging Server。下表描述提供系統服務之遠端存取的 Java Enterprise System 元件。

**表 4-3** Java Enterprise System 提供遠端存取的元件

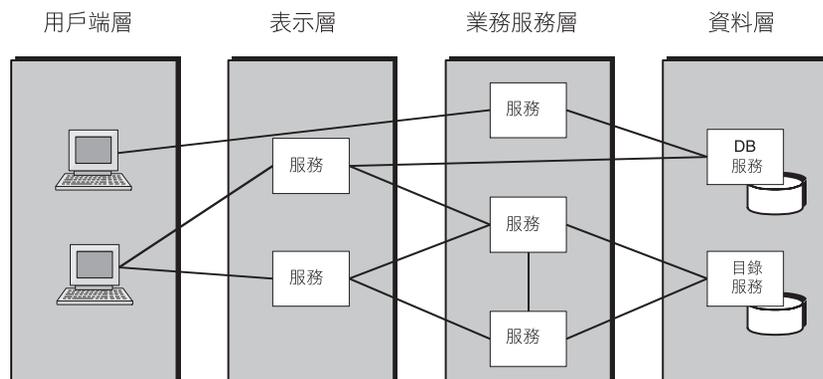
元件	描述
Directory Proxy Server	為多重 Directory Server 實例提供增強的目錄存取、模式相容性、路由與負載平衡。
Portal Server, Portal Server Secure Remote Access	提供從公司防火牆外部對 Portal Server 內容與服務 (包括內部入口網站或網際網路應用程式) 的安全網際網路存取。
Portal Server, Portal Server Mobile Access	提供從行動裝置到 Portal Server 的無線存取和語音存取。
Messaging Server Message Multiplexor (MMP)	支援電子郵件的擷取，方法是代表 Web 型 (HTTP) 用戶端存取訊息儲存。

提供遠端存取的元件通常會部署在安全存取區域中，如第 65 頁的「存取區域」一節的範例所示。

## 多層架構設計

Java Enterprise System 非常適合多層架構設計，因為其中的服務都根據其所提供的功能而放置在不同的層中。每個服務都有獨立的邏輯，而且可供同一層或不同層的服務存取。下圖描述企業應用程式的多層架構，說明了用戶端、表示、業務服務和資料層。

圖 4-4 多層架構模型



下表描述圖 4-4 所述的邏輯層。

表 4-4 多層架構中的邏輯層

層	描述
用戶端層	包含為一般使用者呈現資訊的用戶端應用程式。對 <b>Java Enterprise System</b> 而言，這些應用程式通常是郵件用戶端、 <b>Web</b> 瀏覽器或行動存取用戶端。
表示層	提供為一般使用者顯示資料的服務，可讓使用者處理和控制顯示方式。例如，一個 <b>Web</b> 郵件用戶端或 <b>Portal Server</b> 元件可讓使用者修改所接收資訊的顯示方式。
業務服務層	提供後端服務，這些服務通常在資料層接收資料，再將資料提供到表示層或業務服務層中的其他服務，或是直接將資料提供到用戶端層。例如， <b>Access Manager</b> 將身份識別服務提供到 <b>Java Enterprise System</b> 元件。
資料層	提供可由表示層或業務服務層內服務存取的資料庫服務。例如， <b>Directory Server</b> 提供其他服務的 <b>LDAP</b> 目錄存取。

多層架構設計有數個優點。在部署設計階段期間，根據多層架構中的功能來放置服務，可協助您決定如何分散網路中的服務。您也可以瞭解架構中的元件如何存取其他元件的服務。這個視覺化的表示方法可協助您規劃可用性、延展性、安全性和其他服務品質解決方案。

## 範例邏輯架構

本節提供 Java Enterprise System 解決方案之邏輯架構的某些範例。這些範例會顯示如何在多層架構的適當層中放置邏輯元件，然後再藉由研究使用實例來分析元件之間的關係。將本節的邏輯架構範例作為瞭解 Java Enterprise System 解決方案之邏輯架構設計的基礎。

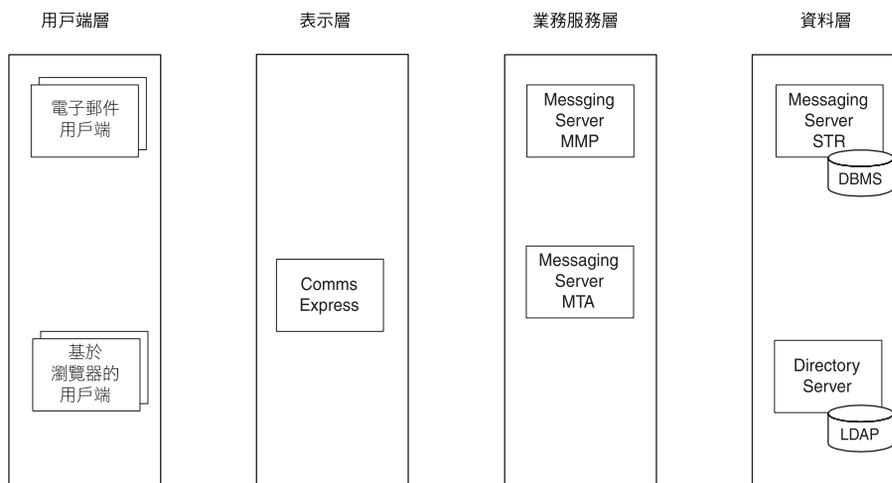
第一個範例是基本的 Messaging Server 解決方案，說明了 Messaging Server 不同邏輯的元件如何與其他元件互動。第二個範例顯示基於身份識別部署的邏輯架構，可能適用於擁有 1000 至 5000 名員工的中型企業。

## Messaging Server 範例

下圖顯示 Messaging Server 部署的基本邏輯架構。此邏輯架構只會顯示 Messaging Server 需要的邏輯獨立元件。之後的圖例會說明這些元件彼此間的關係。

**備註** 一般而言，Messaging Server 部署是企業解決方案的一部份，包括其他的 Java Enterprise System 元件，如第 62 頁的「基於身份識別的通訊範例」所述。

**圖 4-5** Messaging Server 部署的邏輯架構



下表描述圖 4-5 所述的元件。

**表 4-5** Messaging Server 邏輯架構中的元件

元件	描述
電子郵件用戶端	閱讀和傳送電子郵件的用戶端應用程式。
Messaging Server MTA	將 Messaging Server 配置為 Message Transfer Agent (MTA) 以擷取、安排、傳輸和傳送電子郵件。
Messaging Server MMP	將 Messaging Server 配置為 Message Multiplexor (MMP)，可將連線路由到適當的訊息儲存以進行擷取和儲存。MMP 存取 Directory Server 查詢目錄資訊，以決定適當的訊息儲存。
Messaging Server STR	將 Messaging Server 配置為擷取和儲存電子郵件的訊息儲存區。
Directory Server	提供 LDAP 目錄資料的存取。

邏輯架構不會指定 Messaging Server 元件的服務複製。例如，企業部署通常會建立個別的內送和外傳的 MTA 實例，但第 58 頁的圖 4-5 只會顯示一個 MTA 元件。將邏輯元件複製到數個實例中，是您在部署設計階段期間所做的設計決定。

## Messaging Server 使用實例

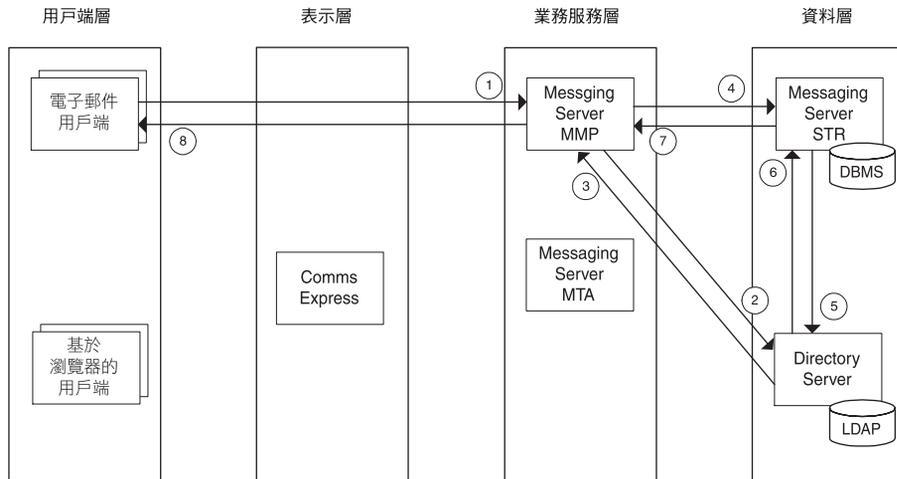
使用實例可協助您識別架構中邏輯元件彼此間的關係。根據使用實例對應元件之間的互動，透過這個動作，您可以看見元件互動的方式，這對部署設計相當有幫助。

一般而言，您會分析每個使用實例，以便在部署設計之前決定元件的互動。下列三個使用實例是 Messaging Server 的典型實例，會顯示邏輯元件中的互動。

### 使用實例 1：使用者成功登入 Messaging Server

1. 電子郵件用戶端將登入資訊傳送到 Messaging Server Multiplexor (MMP)
2. MMP 要求認證 Directory Server 的使用者 ID 和密碼。
3. Directory Server 傳回 MMP 的認證。
4. MMP 要求 Messaging Server 訊息儲存 (STR) 的訊息清單。
5. STR 要求 Directory Server 的使用者 LDAP 記錄。
6. Directory Server 將使用者 LDAP 記錄傳回 STR。
7. STR 將訊息清單傳回 MMP。
8. MMP 將訊息清單轉送到電子郵件用戶端。

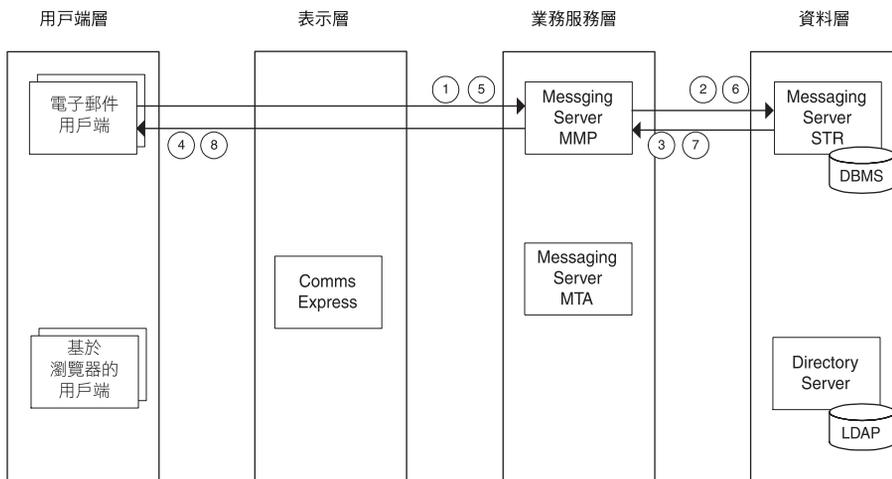
圖 4-6 顯示使用實例 1 的 Messaging Server 邏輯架構



### 使用實例 2：登入的使用者閱讀和刪除郵件

1. 電子郵件用戶端要求從 Messaging Server Multiplexor (MMP) 閱讀訊息。
2. MMP 要求 Messaging Server 訊息儲存 (STR) 的訊息。
3. STR 將訊息傳回 MMP。
4. MMP 將訊息轉送到電子郵件用戶端。
5. 電子郵件用戶端傳送「刪除郵件」動作到 MMP。
6. MMP 轉送「刪除郵件」動作到 STR。
7. STR 將郵件自資料庫中刪除，並傳送確認到 MMP。
8. MMP 將刪除確認轉送到電子郵件用戶端。

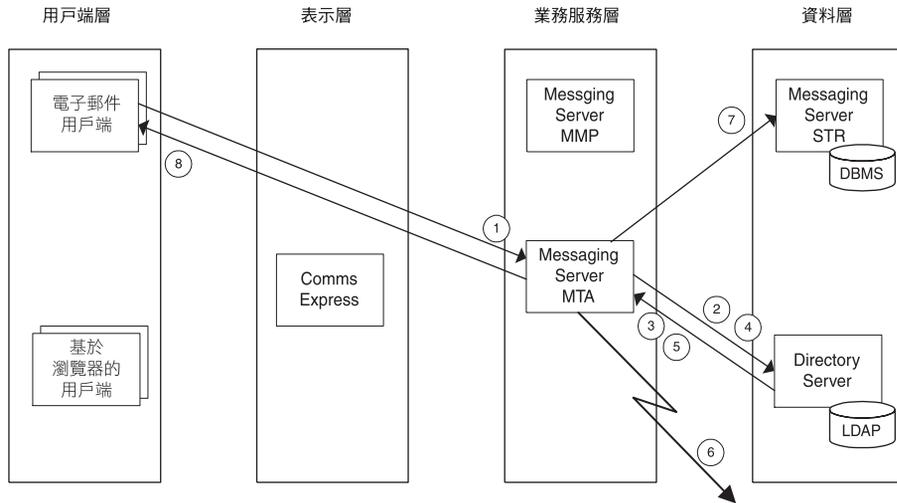
圖 4-7 顯示使用實例 2 的 Messaging Server 邏輯架構



### 使用實例 3：登入的使用者傳送電子郵件

1. 電子郵件用戶端將在用戶端撰寫的訊息傳送到 Messaging Server Message Transfer Agent (MTA)。
2. MTA 要求認證 Directory Server 的使用者 ID 和密碼。
3. Directory Server 傳回 MTA 的認證。
4. MTA 會檢查 Directory Server 以確定每個收件者的目的地網域。
5. Directory Server 會將每個收件者的目的地網域傳回 MTA。
6. MTA 會將訊息轉送給每個收件者。
7. MTA 將訊息轉送到 Messaging Server 訊息儲存 (STR)，以將訊息儲存在寄件匣中。
8. MTA 會將確認傳送到電子郵件用戶端。

圖 4-8 顯示使用實例 3 的 Messaging Server 邏輯架構



## 基於身份識別的通訊範例

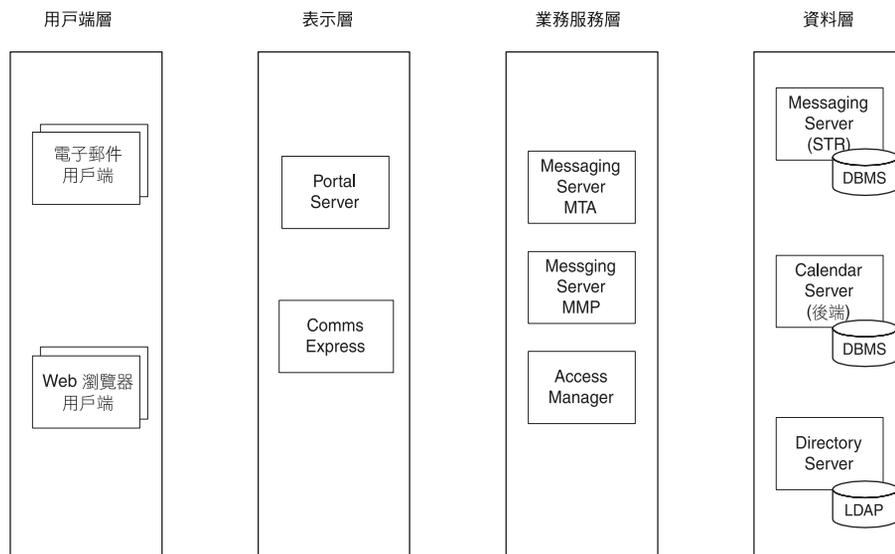
此範例顯示基於身份識別的通訊解決方案，適用於擁有 1000 至 5000 名員工的中型企業。一般而言，設計邏輯架構需要徹底的業務分析，這個分析會在詳細的技術需求分析之後產生。不過，這只是理論性的範例，因此會假設已決定下列的業務需求。

- 企業的員工需要以個人化方式存取內部網站、通訊服務、行事曆服務和其他資源。
- 整個企業的認證和授權會提供內部網站和其他服務的存取。
- 所有企業服務之間將追蹤單次身分識別，啓用可提供內部網站和其他服務存取的單次登入 (SSO)。

此範例的使用實例會詳細說明登入程序、閱讀電子郵件、傳送電子郵件、個人化入口網站、同步化行事曆和其他類似的使用者活動。

下圖顯示此類基於身份識別的通訊解決方案之邏輯架構。

圖 4-9 身份識別型通訊方案的邏輯架構



## 身份識別型通訊範例的使用實例

對於具備此特性的部署解決方案而言，通常有許多個詳細的使用實例，描繪使用者與此解決方案提供之服務間的互動。此範例強調使用者從 Web 瀏覽器用戶端登入入口網站時，元件之間的互動。本範例會將登入方案分為兩個使用實例：

- 使用者登入，認證成功後，Portal Server 就會擷取使用者的入口網站配置。
- Portal Server 擷取電子郵件和行事曆資訊以便在 Web 用戶端中顯示。

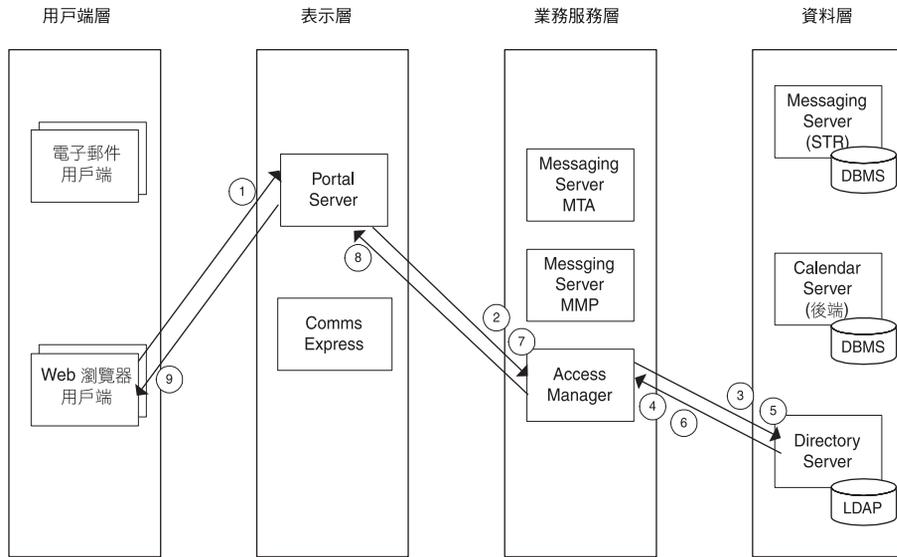
可將這兩個使用實例視為一個延伸的使用實例。不過，此範例分別討論這兩個使用實例以達到簡化的目的。

### 使用實例 1：使用者成功登入，且入口網站擷取使用者配置

1. Web 瀏覽器用戶端將使用者 ID 和密碼傳送到 Portal Server。
2. Portal Server 要求認證 Access Manager。
3. Access Manager 要求認證 Directory Server 的使用者 ID 和密碼。
4. Directory Server 認證使用者 ID 和密碼。
5. Access Manager 要求 Directory Server 的使用者設定檔。
6. Directory Server 傳回使用者設定檔。

7. Portal Server 要求 Access Manager 的使用者顯示設定檔。
8. Access Manager 傳回入口網站配置。
9. 入口網站配置顯示於 Web 瀏覽器用戶端。

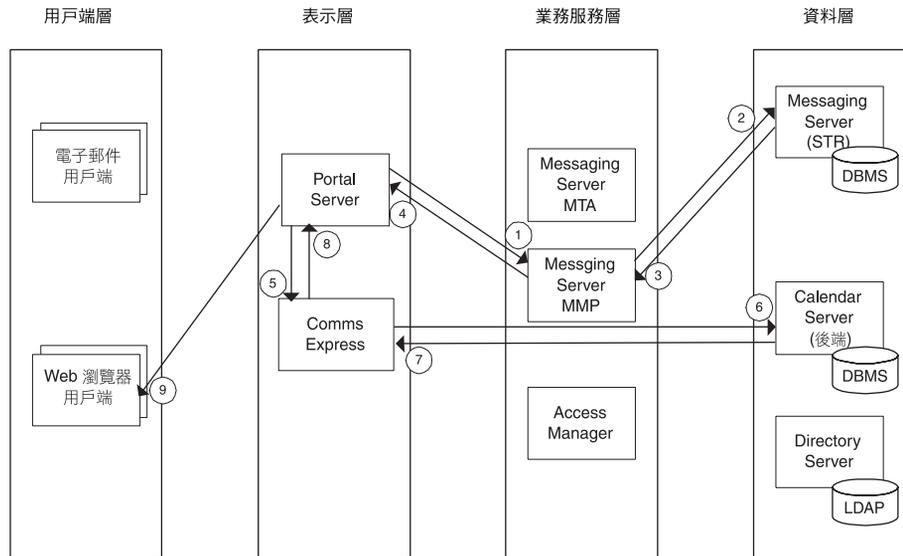
圖 4-10 顯示使用實例 1 的通訊方案邏輯架構



### 使用實例 2：Portal Server 顯示電子郵件和行事曆資訊

1. 在成功登入、認證和擷取入口網站配置之後，Portal Server 要求來自 Messaging Server MMP 的電子郵件。
2. MMP 要求 Messaging Server STR 的訊息清單。
3. STR 將訊息清單傳回 MMP。
4. MMP 將訊息轉送到 Portal Server。
5. Portal Server 要求 Communications Express 的行事曆資訊。
6. Communications Express 要求 Calendar Server 後端的行事曆資訊。
7. Calendar Server 後端將行事曆資訊傳回 Communications Express。
8. Communications Express 將行事曆資訊轉送到 Portal Server。
9. Portal Server 將所有通道資訊傳送到 Web 瀏覽器用戶端。

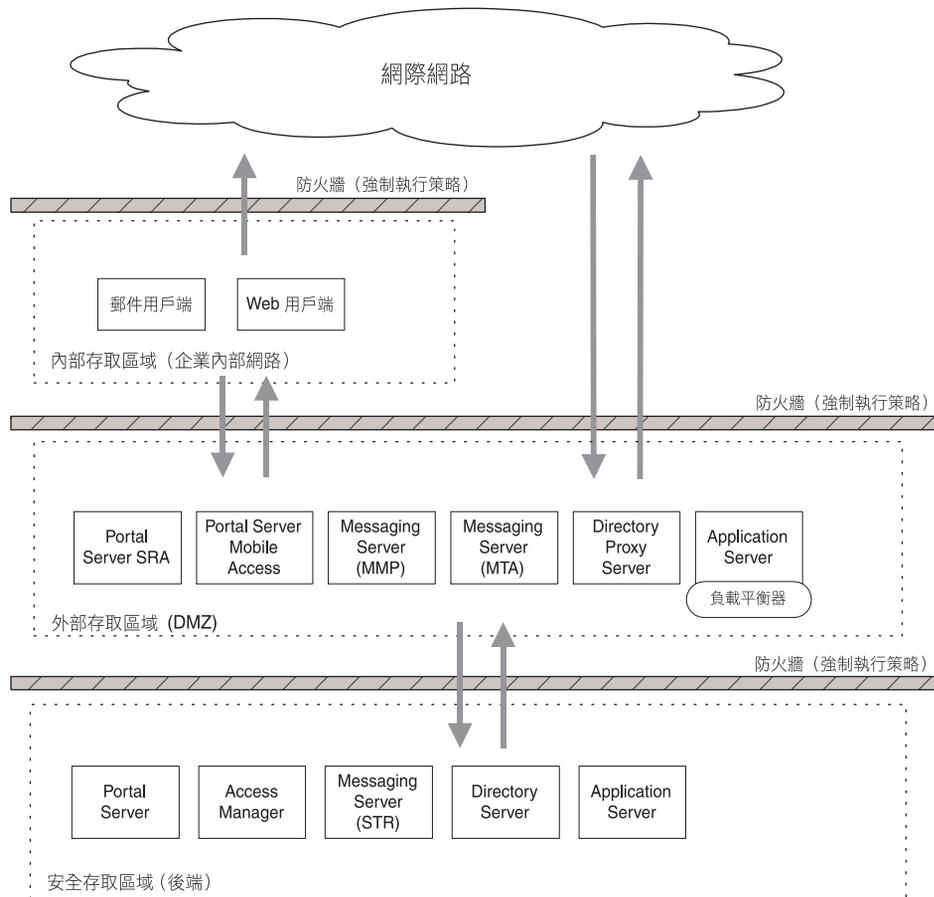
圖 4-11 顯示使用實例 2 的通訊方案邏輯架構



## 存取區域

另一種表示邏輯架構的元件的方法就是將這些元件放置在存取區域中，顯示架構如何提供安全的存取。下圖描述部署 Java Enterprise System 元件的存取區域。每個存取區域都會顯示元件如何在網際網路和企業內部網路之間提供安全的遠端存取。

圖 4-12 放置在存取區域中的邏輯元件



下表描述圖 4-12 所述的存取區域。

表 4-6 安全存取區域和其中所放置的元件

存取區域	描述
內部存取區域 (企業內部網路)	透過企業內部網路和網際網路之間的防火牆所強制執行的策略來存取網際網路。內部存取區域通常會被一般使用者用來進行 Web 瀏覽和傳送電子郵件。  在某些情況下，也會允許直接存取網際網路以進行 Web 瀏覽。不過，網際網路之間的安全存取都是透過外部存取區域而提供的。
外部存取區域 (DMZ)	提供網際網路之間的安全存取，作為關鍵後端服務的安全性緩衝。

**表 4-6** 安全存取區域和其中所放置的元件 (續)

存取區域	描述
安全存取區域 (後端)	提供關鍵後端服務的限制存取，只能從外部的存取區域進行存取。

圖 4-12 不描述先前的範例所述的邏輯層，而是將焦點放在哪個元件提供遠端和內部存取、安全性措施 (例如防火牆) 中這些元件彼此間的關係，並以視覺化方式描述必須強制執行的存取規則。將多層結構設計和可顯示存取區域的設計搭配使用，可提供您規劃部署的邏輯模型。

## 部署方案

僅有完整的邏輯架構設計並不足以開始進行解決方案生命週期的部署設計階段。您必須將邏輯架構與技術需求階段期間決定的服務品質 (QoS) 需求相互搭配。邏輯架構和 QoS 需求的搭配就會組成部署方案。部署方案是設計部署架構的起始點，如第 5 章，「部署設計」中所解釋。



# 部署設計

在解決方案生命週期的部署設計階段期間，您設計高階部署架構和低階實作規格，以及準備實作解決方案需要的一系列規劃和規格。專案核准會在部署設計階段發生。

本章包含以下各節：

- 第 69 頁的「關於部署設計」
- 第 73 頁的「部署設計方法」
- 第 74 頁的「估計處理器需求」
- 第 80 頁的「估計安全交易的處理器需求」
- 第 83 頁的「決定可用性策略」
- 第 91 頁的「決定延展性的策略」
- 第 96 頁的「設計最佳化資源使用」
- 第 98 頁的「範例部署架構」

## 關於部署設計

部署設計以部署方案開始，而部署方案是在解決方案生命週期的邏輯設計和技術需求階段期間所設計的。部署方案包含解決方案的邏輯架構和服務品質 (QoS) 需求。您對應邏輯架構 (跨實體伺服器和其他網路裝置) 中識別的元件以建立部署架構。QoS 需求提供效能、可用性、延展性和其他 QoS 相關規格之硬體配置的指導。

設計部署架構是一個反覆式程序。通常您會重新審視 QoS 需求並重新檢查最初的設計。您會考慮 QoS 需求的相互關係、設法平衡利弊和所有權成本的問題以找出一個能夠達到專案的最終業務目標的最佳解決方案。

## 專案核准

專案核准會在部署設計階段期間發生，通常是在您建立部署架構之後。使用部署架構和以下所述的實作規格（若有可能），則可估計部署的實際成本並將結果提交給利害關係人以進行核准。一旦核准專案後，則會簽訂部署完成的合約並取得及分配實作專案的資源。

## 部署設計輸出資料

在部署設計階段期間，您可能準備下列任何規格和規劃：

- **部署架構**。一個高階架構描述邏輯架構到實體環境的對應關係。實體環境包括企業內部網路或是網際網路環境中的計算節點、處理器、記憶體、儲存裝置和其他硬體及網路裝置。
- **實作規格**。詳細的規格，可作為建立部署的藍圖。這些規格提供電腦和網路硬體的規格，以取得和描述部署的網路佈局。實作規格也包括目錄服務的規格，包含目錄資訊樹狀結構 (DIT) 以及為目錄存取定義的群組和角色的詳細資訊。
- **實作規劃**。一組規劃，涵蓋實作企業軟體解決方案的各個層面。實作規劃包括下列項目：
  - **遷移規劃**。描述遷移企業資料和升級企業軟體的策略和程序。遷移資料必須符合最近安裝之企業應用程式的格式和標準。所有的企業軟體必須都是正確的版本層級，才能進行交互操作。
  - **安裝規劃**。衍生自部署架構，指定硬體伺服器名稱、安裝目錄、安裝順序、每個節點的安裝類型以及安裝和配置分散式部署必要的配置資訊。

- **使用者管理規劃**。包括在現有目錄和資料庫中遷移資料的策略、目錄設計規格 (需考慮部署架構中指定的複製設計) 以及使用新內容佈建目錄的程序。
- **測試規劃**。描述測試部署軟體的程序, 包括開發原型和引導實作的特定規劃、決定處理專案負載能力的加強測試、以及決定規劃的功能是否如預期運作的功能測試。
- **建置規劃**。描述將實作從規劃和測試環境移動到生產環境的程序及排程。將實作移動到生產通常會發生在各個階段中。例如, 第一個階段可能會為有限群組的使用者部署軟體, 並在完成整個部署之前於每個階段增加使用者基礎。在完成整個部署之前, 階段式實作也可以包括特定軟體套件的排程實作。
- **損壞修復規劃**。描述在意外的整個系統失敗時如何還原系統的程序。修復規劃包括大型和小型失敗的程序。
- **作業規劃 (操作手冊)**。作業手冊會描述監視、維護、安裝和升級的程序。
- **訓練規劃**。包含在最近安裝的企業軟體上訓練作業人員、管理員和一般使用者的過程和程序。

## 影響部署設計的因素

有數個因素會影響您在部署設計期間所做的決定。請考慮下列關鍵因素：

- **邏輯架構**。邏輯架構會詳細說明提議解決方案中的功能服務以及提供那些服務之元件的相互關係。將邏輯架構作為決定分配服務之最佳作法的關鍵。部署方案包含與服務品質元件需求相互搭配的邏輯架構 (如下所示)。

- **服務品質需求。** 服務品質 (QoS) 需求會指定解決方案作業的各種層面。使用 QoS 需求可協助開發策略，以達成效能、可用性、延展性、服務性和其他服務品質的目標。部署方案包含與服務品質需求相互搭配的邏輯架構 (如先前描述)。
- **使用分析。** 在解決方案生命週期的技術需求期間開發的使用分析會提供使用模式的資訊，這可協助估計部署系統的負載和加強。利用使用分析來協助隔離效能瓶頸並開發可滿足 QoS 需求的策略。
- **使用實例。** 在解決方案生命週期的技術需求期間開發的使用實例會列出部署的不同使用者互動，通常是找出最常見的使用實例。雖然使用實例屬於使用分析的一部份，但在評估部署設計時，您應該要參閱使用實例，確保已經有正確說明。
- **服務層級合約。** 服務層級合約 (SLA) 會指定最低的效能需求、無法滿足這些需求的時段，以及必須提供的用戶支援之層級和範圍。部署設計應該可以很輕易地達到服務層級合約指定的效能需求。
- **所有權的總成本。** 在部署設計階段期間，您分析潛在的解決方案，其會說明可用性、效能、延展性和其他功能的 QoS 需求。不過，針對每個您列入考慮的解決方案，您也必須考慮該解決方案的成本，以及該成本對所有權總成本的影響。請確實考量您的決定之利弊，以及您已經使資源最佳化以達成業務限制中的業務需求。
- **業務目標。** 在解決方案生命週期的業務分析階段將說明業務目標，且業務目標會包含達成這些目標所需的業務需求和業務限制。最後會依部署設計達成業務目標的能力決定其優劣。

# 部署設計方法

如同部署規劃的其他層面，部署設計同時兼具藝術與科學的特性，無法以特定的程序和過程來詳細說明。成功部署設計的因素包含了過去的設計經驗、系統架構的知識、網域知識和實用的創意思考。

部署設計通常以達成效能需求為中心並同時滿足其他 QoS 需求。您所使用的策略必須能平衡設計決定的利弊，才能最佳化解決方案。您使用的方法通常會包含下列作業：

- **估計處理器需求。** 部署設計通常是從估計邏輯架構中每個元件所需要的 CPU 數目開始。以代表最大負載的使用實例開始，再繼續進行每個使用實例。考慮支援使用實例之所有元件的負載，並依實際情況修改估計。另外也要將您之前設計企業系統的經驗列入考慮。
- **估計安全傳輸的處理器需求。** 研究需要安全傳輸的使用實例並依實際情況修改 CPU 估計。
- **可用性和延展性的複製服務。** 如果您對處理器的估計感到滿意，即可修改負責可用性和延展性 QoS 需求的設計。考慮處理可用性和防故障備用問題的負載平衡解決方案。

在分析期間，請考慮設計決定的利弊。例如，可用性及延展性策略對系統的服務性（維護）有什麼影響？策略的其他成本為何？

- **找出瓶頸。** 當您繼續進行分析時，請檢查部署設計以找出任何使資料傳輸無法達到需求標準的瓶頸後再進行判斷。
- **最佳化資源。** 審核資源管理的部署設計，並考慮在滿足需求時可將成本降到最低的選項。
- **管理風險。** 重新檢視關於設計的業務和技術分析，進行修改以應對早前規劃階段未預見的事件和情況。

# 估計處理器需求

本節會討論評估支援部署系統中服務需要的 CPU 處理器數目和對應記憶體的過程。本節也會為您說明評估過程，並以通訊部署方案為範例說明。

評估 CPU 計算能力是一項反覆式過程，需要考慮到下列項目：

- 邏輯元件及其互動（如邏輯架構中的元件相依性所列）
- 已識別的使用實例的使用分析
- 服務品質需求
- 過去使用部署設計和 Java Enterprise System 的經驗
- 向具有設計和實作各種類型部署方案經驗的 Sun 專業服務部門諮詢

評估過程包括下列步驟。這些步驟的順序並不是關鍵 — 這只是提供一個考慮影響最後結果之因素的方法。

## 1. 決定作為系統的使用者進入點元件的基線 CPU 估計。

其中一項是決定要完整載入或部分載入 CPU。完整載入的 CPU 可將系統的容量最大化。若要增加容量，可能會因為增加額外的 CPU 而產生維護成本和可能的當機時間。在某些情況下，您可以選擇增加額外的機器以符合增加的效能需求。

部份載入的 CPU 可允許處理額外效能需求的空間，而不會立即產生維護成本。不過，運用不足的系統會有額外的前置費用。

## 2. 調整 CPU 估計來處理元件之間的互動。

研究邏輯架構中元件彼此間的互動，以決定因為獨立元件而產生的額外負載。

3. 研究特定使用實例的使用分析，以決定系統的尖峰負載，然後再對處理尖峰負載的元件進行調整。

從權重最高的使用實例（亦即需要最多負載）開始，再繼續研究每個使用實例，確實考慮所有的計畫性使用方案。

4. 調整 CPU 估計來反映安全性、可用性和延展性需求。

此評估過程提供決定您需要的實際處理能力的起點。一般而言，您會先建立以這些評估值為基礎的原型部署，然後再對預期的使用實例執行嚴格的測試。只有在經過反覆的測試之後，您才能決定部署設計的實際處理需求。

## 範例估計處理器需求

本節說明一個方法，來評估範例部署所需的處理能力。範例部署是以身份識別型通訊解決方案的邏輯架構為基礎，適用於擁有 1000 至 5000 名員工的中型企業，如第 62 頁的「基於身份識別的通訊範例」章節所述。

本範例中使用的 CPU 和記憶體為隨意估計，僅限為圖解使用。這些圖例以理論性範例所使用的隨意資料為根據。需要徹底分析各種因素才能評估處理器需求。此分析會包括（但不限於）下列資訊：

- 以徹底業務分析為基礎的詳細使用實例和使用分析
- 由業務需求分析所決定的服務品質需求
- 處理和網路硬體的特定成本和規格
- 過去實作類似部署的經驗

---

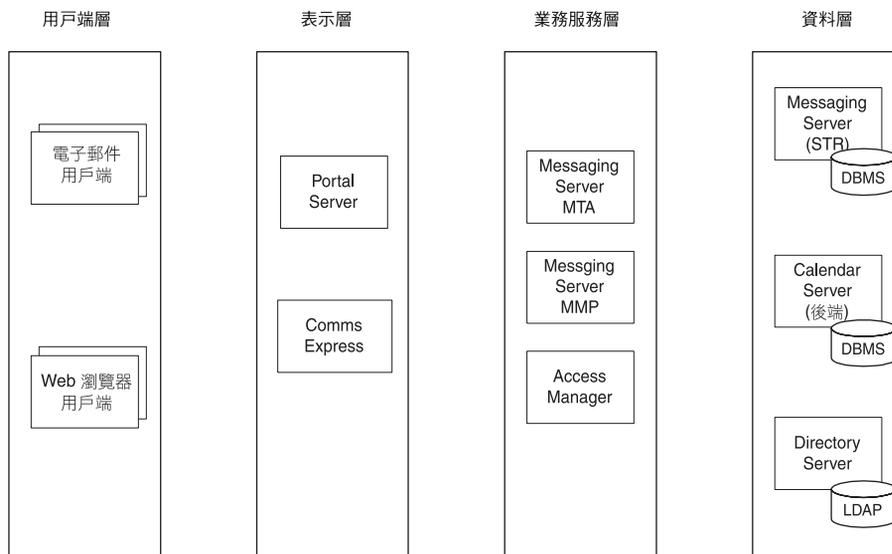
**警告** 除了描述您在設計系統時可能使用的過程外，這些範例所示的資訊不代表任何特定的實作建議。

---

## 決定使用者進入點的基線 CPU 估計

從估計處理每個作為使用者進入點的元件上預期的負載所需的 CPU 數量開始。下圖顯示在第 62 頁的第 4 章「邏輯設計」中描述的身份識別型通訊方案的邏輯架構。

圖 5-1 身份識別型通訊方案的邏輯架構



下表列出邏輯架構之表示層中的元件，其會直接與部署的一般使用者聯繫。這個表格包括基線 CPU 估計，這些估計源自於技術需求分析、使用實例、特定使用分析和過去處理此類型部署的經驗。

表 5-1 元件的 CPU 估計包含存取使用者進入點的基線 CPU 估計

元件	CPU 的數目	描述
Portal Server	4	作為使用者進入點的元件。
Communications Express	2	將資料路由到 Portal Server 訊息傳送和行事曆通道。

## 包括服務相依性的 CPU 估計

提供使用者進入點的元件需要 Java Enterprise System 元件的支援。若要繼續指定效能需求，請納入效能估計以考慮其他元件需要的支援。在設計邏輯架構時，應該詳細說明元件間的互動類型，如第 58 頁的「範例邏輯架構」章節的邏輯架構範例所述。

表 5-2 支援元件的 CPU 估計

元件	CPU	描述
Messaging Server MTA (內送)	1	路由來自 Communications Express 和電子郵件用戶端的傳入電子郵件。
Messaging Server MTA (外傳)	1	路由外送的電子郵件給收件者。
Messaging Server MMP	1	存取電子郵件用戶端的 Messaging Server 郵件儲存。
Messaging Server STR (訊息儲存)	1	擷取和儲存電子郵件。
Access Manager	2	提供授權與認證服務。
Calendar Server (後端)	2	擷取和儲存 Communications Express (Calendar Server 前端) 的行事曆資料。
Directory Server	2	提供 LDAP 目錄服務。
Web Server	0	提供 Portal Server 和 Access Manager 的 Web 容器支援。 (不需要額外的 CPU 循環。)

## 研究尖峰負載使用的使用實例

回到使用實例和使用分析，以找出尖峰負載使用的區域，並調整您的 CPU 估計。

例如，在此例中您發現下列的尖峰負載情況：

- 當使用者同時登入時，開始累計使用者
- 在特定的時間範圍內交換電子郵件

若要應付這個尖峰負載使用，請調整提供這些服務的元件。下表會概述您可以針對應付這個尖峰負載所進行的調整。

表 5-3 CPU 估計尖峰負載的調整

元件	CPU (已調整)	描述
Messaging Server MTA 內送	2	為尖峰傳入電子郵件新增 1 個 CPU
Messaging Server MTA 外傳	2	為尖峰外送電子郵件新增 1 個 CPU
Messaging Server MMP	2	為額外負載新增 1 個 CPU
Messaging Server STR (訊息儲存)	2	為額外負載新增 1 個 CPU
Directory Server	3	為額外 LDAP 查詢新增 1 個 CPU

## 修改其他負載情況的估計

繼續進行 CPU 估計，考慮其他會影響負載的服務品質需求：

- **安全性。** 在技術需求階段中，決定資料的安全傳輸會如何影響負載需求，再對您的估計進行對應的修改。下列章節 — [第 80 頁的「估計安全交易的處理器需求」](#) — 會描述進行調整的過程。
- **複製服務。** 調整 CPU 估計以滿足可用性、負載平衡和延展性問題的複製服務需求。下列章節 — [第 83 頁的「決定可用性策略」](#) — 描述如何調整可用性解決方案的大小。此節 [第 91 頁的「決定延展性的策略」](#) 討論包含可存取目錄服務的解決方案。
- **潛在容量和延展性。** 視需要修改 CPU 估計，使部署上可接受意外的大型負載潛在容量。注意調整的預期重大事件以及一段期間中計畫性負載的增加，確定您可以達成調整系統 (不管是以水平或垂直調整的方式) 的所有計畫性重大事件。

## 更新 CPU 估計

通常，您將 CPU 數目集中為偶數。集中為偶數數目的 CPU 可讓您平均地將 CPU 估計分散到兩個實體伺服器之間，並為潛在容量增加一點空間。不過，請根據您對複製服務的特定需求進行集中。

一般的規則是每個 CPU 應有 2 GB 的記憶體。所需的實際記憶體是依您特定的使用為依據，並可在測試時決定。

下表列出身份識別型通訊範例的最終估計。這些估計沒有包含任何專為安全性和可用性增加的額外運算能力。安全性和可用性的總和將新增於下列章節中。

**表 5-4** 支援元件的 CPU 估計調整

元件	CPU	記憶體
Portal Server	4	8 GB
Communications Express	2	4 GB
Messaging Server (MTA, 內送)	2	4 GB
Messaging Server (MTA, 外傳)	2	4 GB
Messaging Server (MMP)	2	4 GB
Messaging Server (訊息儲存)	2	4 GB
Access Manager	2	4 GB
Calendar Server	2	4 GB
Directory Server	4	8 GB (由 3 個 CPU 集中 / 6 GB 的記憶體)
Web Server	0	0

## 估計安全交易的處理器需求

資料的安全傳輸包括通過安全傳輸協定處理交易，例如 Secure Sockets Layer (SSL) 和傳輸層安全性 (TLS)。通過安全傳輸處理的交易通常需要額外的計算能力，首先要建立安全階段作業（稱為訊號交換），然後再加密和解密傳輸資料。根據使用的加密演算法（例如 40 位元或 128 位元加密演算法），可能另外需要充足的計算能力。

考慮在非安全交易的同層級上執行安全交易時，您必須規劃額外的計算能力。根據交易的特性以及處理交易的 Sun Java™ Enterprise System 服務而定，安全交易可能需要比非安全交易多出四倍以上的計算能力。

當評估處理安全交易的處理能力時，請分析使用實例來決定需要安全傳輸的交易百分比。如果安全交易的效能需求與非安全交易相同，則需修改 CPU 數量的估計值，將安全交易所需的額外計算能力考慮在內。

在某些使用方案中，安全傳輸可能只需要用來認證。一旦使用者經由系統認證，就不需要額外的資料傳輸安全性評量。在其他方案中，所有的交易可能都需要安全傳輸。

例如，在線上電子商務網站瀏覽產品目錄時，直到用戶完成選擇並準備「結帳」前，所有的交易可能都是不安全的。然而，有些使用方案，例如銀行或證券經紀商的部署則非常需要安全交易或希望全部的交易都能安全並應用相同的效能標準到安全和非安全交易。

### 安全交易的 CPU 估計

本節繼續以範例部署說明如何計算理論性使用實例的 CPU 需求，實例包括安全和非安全交易。

若要估計安全交易的 CPU 需求，請進行以下計算：

1. 從 CPU 估計的基礎計算開始，如同您在前一節第 75 頁的「範例估計處理器需求」所說明的。
2. 計算需要安全傳輸的交易百分比，並計算安全交易的 CPU 估計。
3. 計算非安全交易已減少的 CPU 估計。
4. 清點安全和非安全估計以計算總 CPU 估計。
5. 將總 CPU 估計集中為偶數。

表 5-5 顯示以 Portal Server 的使用實例和使用分析為基礎的範例計算，其假設如下：

- 所有的登入都需要安全認證。
- 所有的登入帳戶使用總 Portal Server 負載的 10%。
- 安全交易的效能需求與非安全交易的效能需求相同。

若要說明處理安全交易的額外計算能力，處理這些交易的 CPU 數目將以 4 倍增加。如同此範例中的其他 CPU 圖例，這個係數是隨意的，且僅限於描述之用。

表 5-5 修改 CPU 安全交易的估計

步驟	描述	計算	結果
1	從所有的 Portal Server 交易的基線估計開始。	第 78 頁的表 5-3 的基線估計是 4 個 CPU。	-----
2	計算安全交易的額外 CPU 估計。假設安全交易需要的 CPU 能力是非安全交易的 5 倍。	10% 的基礎估計需要安全傳輸： $0.10 \times 4 \text{ CPU} = 0.4 \text{ CPU}$ 增加安全交易的 CPU 能力 4 倍： $4 \times 0.4 = 1.6 \text{ CPU}$	1.6 CPU
3	計算非安全交易已減少的 CPU 估計。	90% 的基線估計是不安全的： $0.9 \times 4 \text{ CPU} = 3.6 \text{ CPU}$	3.6 CPU
4	計算安全和非安全交易的已調整總 CPU 估計。	安全估計 + 非安全估計 = 總計： $1.6 \text{ CPU} + 3.6 \text{ CPU} = 5.2 \text{ CPU}$	5.2 CPU
5	集中為偶數。	$5.2 \text{ CPU} \Rightarrow 6 \text{ CPU}$	6 CPU

從此範例的安全交易的計算，您可以在第 81 頁的表 5-5 修改總 CPU 估計，方法是增加 2 個 CPU 和 4 GB 的記憶體，即可取得下列的 Portal Server 總計。

表 5-6 Portal Server 安全交易的 CPU 估計調整

元件	CPU	記憶體
Portal Server	6	12 GB

## 處理 SSL 交易的專用硬體

專門硬體裝置，例如 SSL 加速卡和其他裝置，可用來提供處理安全階段作業建立的計算能力，和資料的加密和解密。在使用專門的 SSL 作業硬體時，計算能力專屬於 SSL 計算的某一部份，這部份通常是建立安全階段作業的「SSL 訊號交換」作業。

此硬體可能對您最後的部署架構有益。然而，因為硬體的專門特性，請先評估 CPU 能力方面的安全交易效能需求，然後再考慮使用專門硬體處理額外負載的利益。

當使用專門硬體時要考慮的一些因素是使用實例是否支援使用硬體（例如，需要大量「SSL 訊號交換」作業的使用實例），以及此類硬體帶給設計的複雜性附加層面。該複雜性包括安裝、配置、測試和裝置管理。

# 決定可用性策略

開發可用性需求的策略時，請研究元件互動和使用分析來決定要考慮哪個可用性解決方案。以基於元件的方式來進行分析，決定最能符合可用性和防故障備用需求的解決方案。

下列項目是您蒐集的資訊類型範例，可用來決定可用性策略：

- 可用性中指定了多少個「九」？
- 哪些是與防故障備用狀況相關的效能規格（例如在防故障備用期間最少為效能的 50%）？
- 使用分析是否區分尖峰和非尖峰使用的時間？
- 地理考量為何？

您選擇的可用性策略也要考慮服務性需求，如第 91 頁的「決定延展性的策略」中所討論。避免需要大量管理和維護的複雜型解決方案。

## 可用性策略

Java Enterprise System 部署的可用性策略包括下列項目：

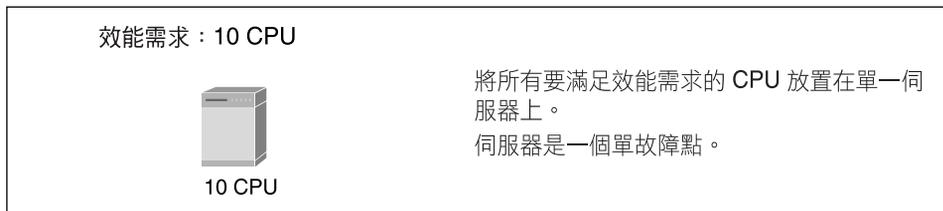
- **負載平衡。** 使用備援硬體和軟體元件以共用處理負載。負載平衡器會將服務的任何要求導向到該服務的其中一個多重對稱式實例。如果任何一個實例失敗，還可以使用其他實例繼續較大量的負載。
- **防故障備用。** 包含管理備援的硬體和軟體，以便在任何元件失敗時，能夠持續存取服務和安全性以取得關鍵的資料。  
Sun Cluster 軟體為關鍵資料提供防故障備用解決方案，這些資料是由後端元件，例如 Messaging Server 的訊息儲存和 Calendar Server 的行事曆資料所管理。
- **複製服務。** 複製服務提供數個來源，讓您存取相同的資料。Directory Server 為 LDAP 目錄存取提供了許多個複製和同步化策略。

下列章節提供數個可用性解決方案的範例，這些解決方案會提供數個層級的負載平衡、防故障備用和複製服務。

## 單一伺服器系統

將所有服務的計算資源放置在單一伺服器上。如果伺服器發生錯誤，則整個服務會失敗。

圖 5-2 單一伺服器系統



Sun 提供高階伺服器，其提供下列效益：

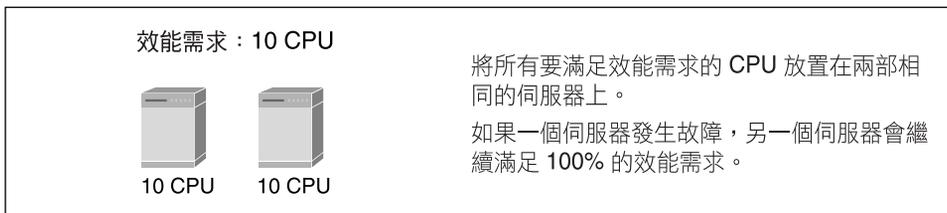
- 在系統執行時硬體元件可取代和重新裝配
- 在伺服器上的錯誤隔離網域中執行多個應用程式的能力
- 無須重新啟動系統即可升級容量、效能速度和 I/O 組態的能力

高階伺服器通常比較的多伺服器系統花費更大。然而，單一伺服器提供管理上的節約、監視和控管資料中心內伺服器的成本。負載平衡、防故障備用和單一失敗點的移除比多伺服器系統更有彈性。

## 水平備援系統

有數種方式可使用平行備援伺服器（同時提供負載平衡和防故障備用）增加可用性。下圖說明兩個重複伺服器提供 N+1 防故障備用系統。N+1 系統在一個伺服器錯誤時有額外的伺服器可提供 100% 的容量。

圖 5-3 N+1 防故障備用系統和兩個伺服器

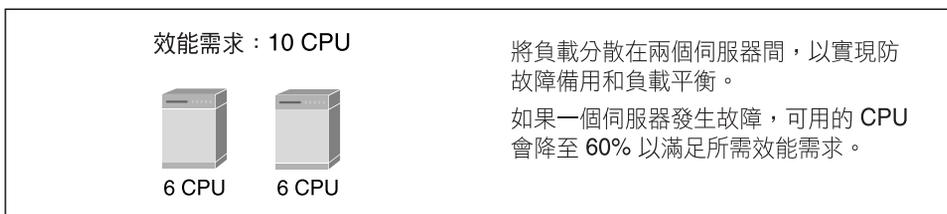


在上述圖 5-3 當中每個伺服器的計算能力是相同的。一個伺服器單獨處理效能需求。其他伺服器在呼叫至服務中做為備份時提供 100% 的效能。

N+1 防故障備用設計的優點是在防故障備用情況下還能擁有 100% 的效能。缺點包括增加硬體成本而沒有獲得對應的整體效能（因為有一個作為備用的伺服器，僅限在防故障備用情況中使用）。

下圖說明實作負載平衡和防故障備用的系統，該系統會在兩部伺服器之間分配效能。

圖 5-4 兩個伺服器之間的負載平衡和防故障備用

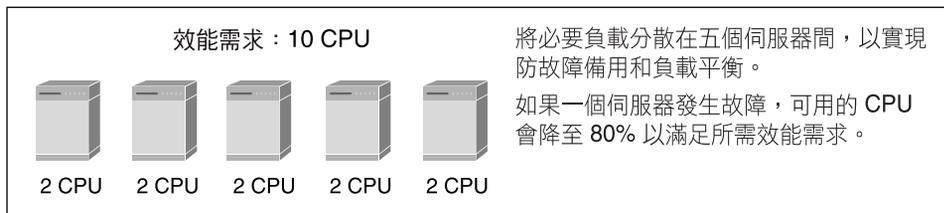


在上圖 5-4 所述的系統中，如果一個伺服器錯誤，所有服務仍然可使用，儘管只是全部容量的一部分。剩餘的伺服器提供 6 CPU 的計算能力，為 10 CPU 需求的 60%。

此設計的優點為兩個伺服器都可使用時有額外的 2 CPU 潛在容量。

下圖說明在一些伺服器之間的效能和負載平衡的分配。

圖 5-5 在  $n$  個伺服器之間分配負載



因為在圖 5-5 中的設計有五個伺服器，如果一個伺服器錯誤，剩餘的伺服器提供的計算能力合計為 8 CPU，其為 10 CPU 效能需求的 80%。如果您增加 2 CPU 容量的額外伺服器到設計中，實際上您已具有一個 N+1 設計。如果一個伺服器錯誤，100% 的效能需求會由剩餘的伺服器達成。

本設計包含以下優點：

- 增加單一伺服器錯誤時的效能
- 當一個以上的伺服器當機時依然可用
- 伺服器可轉出服務以便維護和升級
- 多個低階伺服器通常花費低於單一高階伺服器

然而，使用額外的伺服器會明顯地增加管理和維護費用。您也必須考慮在資料中心中控管這些伺服器的成本。在某些時候您會遇到新增額外伺服器造成的利潤縮減。

## Sun Cluster 軟體

在需要高度可用性的情況（例如四或五個「九」），您可能會考慮 Sun Cluster 軟體作為您的可用性設計的一部份。叢集系統是備援伺服器、儲存裝置和其他網路資源的結合。叢集中的伺服器保持互相通訊。如果一個伺服器離線，叢集中剩餘的裝置會隔離該伺服器，並將任何應用程式或資料從錯誤節點故障轉移到其他節點。錯誤轉移過程相對上會快速的完成，並且只對使用者的系統造成極小的服務中斷。

Sun Cluster 軟體需要額外的專用硬體和專門技術來配置、管理和維護。

## 可用性設計範例

本節包含兩個可用性策略範例，主要是以基於身份識別部署的通訊解決方案，適用於擁有 1000 至 5000 名員工的中型企業，如第 62 頁的「基於身份識別的通訊範例」中所述。第一個可用性策略說明 Messaging Server 的負載平衡。第二個範例則說明使用 Sun Cluster 軟體的防故障備用解決方案。

### Messaging Server 的負載平衡範例

下表列出邏輯架構中，每個邏輯 Messaging Server 元件的 CPU 能力估計。這個表格會重複第 79 頁的「更新 CPU 估計」章節中計算的最終估計。

表 5-7 支援元件的 CPU 估計調整

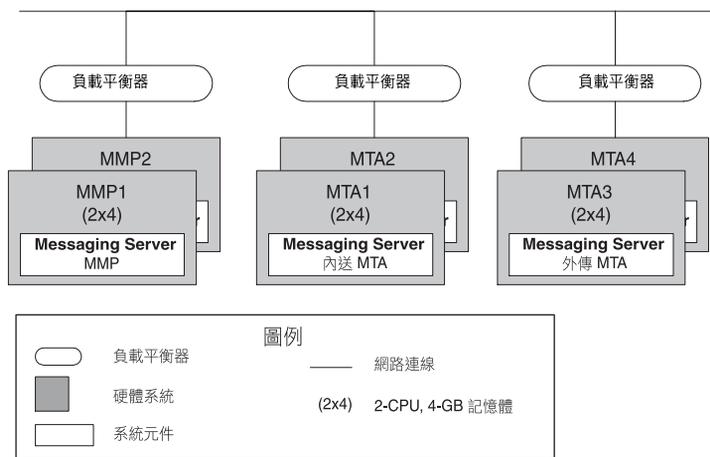
元件	CPU	記憶體
Messaging Server (MTA, 內送)	2	4 GB
Messaging Server (MTA, 外傳)	2	4 GB
Messaging Server (MMP)	2	4 GB
Messaging Server (訊息儲存)	2	4 GB

對此範例而言，假設在技術需求階段期間，您已經指定下列的服務品質需求：

- **可用性。** 整體的系統可用性應該是 99.99% (不包含排程的當機時間)。個別電腦系統的失敗應該不會導致伺服器失敗。
- **延展性。** 在每日尖峰負載下，不應該讓任何伺服器使用超過 80%，且系統必須能夠長期地應付每年 10% 的成長。

若要滿足可用性需求，針對每個 Messaging Server 元件提供兩個實例，每一個實例都位於個別的伺服器上。如果一個元件的伺服器失敗，其他伺服器會繼續提供服務。下表說明此可用性策略的網路圖表。

圖 5-6 範例的 Messaging Server 可用性



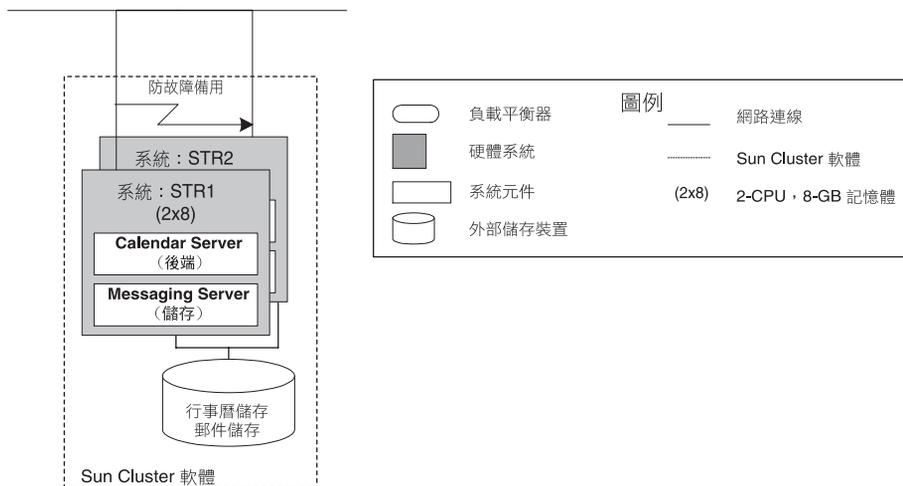
在之前的圖表中，CPU 的數目已經比原本的估計增加了一倍。CPU 增加一倍的理由如下：

- 在一個伺服器失敗的情況下，其餘的伺服器會繼續提供處理負載的 CPU 能力。
- 對於延展性需求 (在尖峰負載下，不應該讓單一伺服器使用超過 80%)，新增的 CPU 能力會提供這個安全性極限。
- 對於延展性需求 (系統必須容納每年 10% 的成長)，新增的 CPU 能力會增加潛在容量，以便在需要額外調整時能夠處理增加的負載。

### 使用 Sun Cluster 軟體的防故障備用範例

下表顯示 Calendar Server 後端和 Messaging Server 訊息儲存的防故障備用策略的範例。Calendar Server 後端和訊息儲存會在個別的伺服器上複製，並使用 Sun Cluster 軟體配置為防故障備用。將於 Sun Cluster 的每個伺服器上複製 CPU 數目和對應的記憶體。

圖 5-7 使用 Sun Cluster 軟體的防故障備用設計



## 目錄服務複製的範例

可將目錄服務複製到不同伺服器的分散式交易中，提供高可用性。Directory Server 提供複製服務的各種策略，包括下列項目：

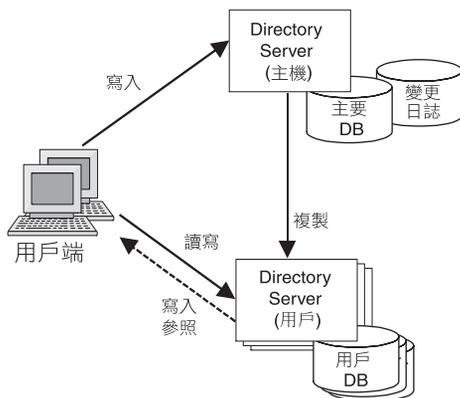
- **多個資料庫。** 在個別的資料庫中儲存不同部分的目錄樹狀結構。
- **鏈接和參照。** 將分散式資料連結到單一的目錄樹狀結構。
- **單一主機複製。** 提供主要資料庫的集中來源，之後再分散到用戶的複本。
- **多重主機複製。** 將主要資料庫分散在數個伺服器之間。每個主要資料庫接著都會將他們的資料庫分散到用戶的複本。

Directory Server 的可用性策略是一個複雜的主題，不屬於此指南的涵蓋範圍。下列章節（[單一主機複製](#)和[多重主機複製](#)）提供基本複製策略的高階檢視。如需 Directory Server 之可用性策略的詳細資訊，請參閱「Directory Server Deployment Planning Guide」（<http://docs.sun.com/doc/817-7607>）。

## 單一主機複製

下圖顯示說明基本複製概念的單一主機複製策略。

圖 5-8 單一主機複製範例



在單一主機複製中，Directory Server 的一個實例會管理主要目錄資料庫並記錄所有變更。主要資料庫會複製到所有的用戶資料庫中。Directory Server 的用戶實例將最佳化以進行讀取和搜尋作業。用戶接收的任何寫入作業將會被導回主要資料庫中。主要資料庫會定期更新用戶資料庫。

單一主機複製的優點包括：

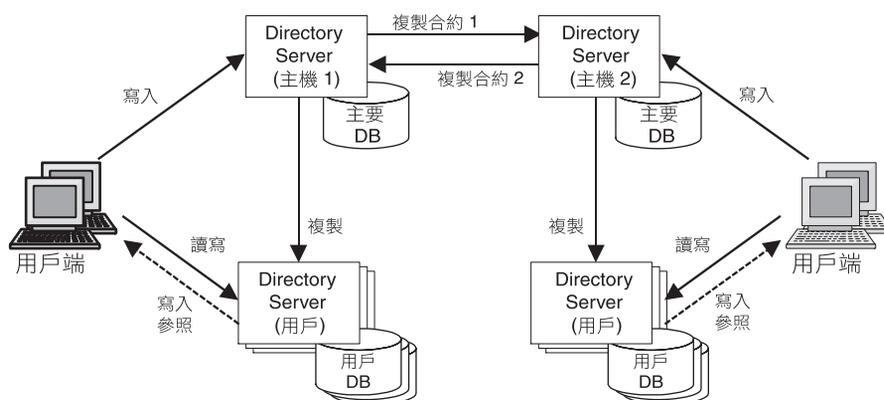
- Directory Server 的單一實例將最佳化以進行資料庫讀取和寫入作業
- 所有 Directory Server 的用戶實例將最佳化以進行讀取和搜尋作業
- Directory Server 用戶實例的水平延展性

## 多重主機複製

下圖顯示多重主機複製策略，可用來全域分散目錄存取。

在多重主機複製中，一或多個 Directory Server 實例會管理主要的目錄資料庫。每個主要資料庫都有複製協定，指定同步化主要資料庫的程序。每個主要資料庫複製為所有的用戶資料庫。和單一主機複製一樣，也會將 Directory Server 的用戶實例最佳化，以進行讀取和搜尋存取。用戶接收的任何寫入作業將會被導回主要資料庫中。主要資料庫會定期更新用戶資料庫。

圖 5-9 多重主機複製範例



多重主機複製策略擁有單一主機複製的所有優點，並具有可提供更新主要資料庫負載平衡的可用性策略。您也可以實作可用性策略，提供目錄作業的本端控制，這對資料中心分佈於全世界的企業而言是一個很重要的考量。

## 決定延展性的策略

延展性是增加容量到您系統的能力，通常增加系統資源，但是不變更部署架構。在需求分析期間，您通常會根據業務需求和之後的使用分析來建立系統預期成長的設計。這些系統使用者數目的設計，以及符合其需求的系統容量，經常是與實際部署系統的數目會有顯著差異的評估。您的設計應該有足夠的彈性容許規劃中的差異。

可延伸的設計包括足夠的潛在容量，以便在系統使用額外資源升級之前，能夠處理增加的負載。可延伸的設計能可輕易調整以應付增加的負載，而無須重新設計系統。

## 潛在容量

潛在容量是您可以將額外的效能和可用性資源容納進系統中的延展性層面，以便系統可以處理不尋常的尖峰負載。您也可以監視在部署的系統中如何使用潛在容量，協助決定何時增加資源來調整系統。潛在容量是將安全性帶入您設計中的一種方式。

使用實例的分析可協助找出會造成不尋常尖峰負載的方案。使用這項不尋常尖峰負載的分析，加上容納非預期成長的因素，設計出可將安全性帶入系統的潛在容量。

您的系統設計應該能夠處理計畫容量一段合理的時間，通常是作業的最初 6 到 12 個月。維護週期可在需要時用來增加資源或增加容量。理想狀況下，您應該可以定期排程系統的升級，但是預測需要的容量增加通常會很困難。依靠資源的仔細監視以及業務設計來決定何時要升級系統。

如果您規劃在遞增式階段中實作解決方案，您必須在每個遞增式階段，將系統容量的增加和其他排程的改善功能排程為同時發生。

## 延展性範例

本節的範例說明以水平和垂直方式調整實作 Messaging Server 的解決方案。若要以水平方式調整，請將額外的 CPU 增加到伺服器中以處理增加的負載。若要以垂直方式調整，請增加額外的伺服器來處理增加的負載，從而分散負載。

範例基準假設為由兩個訊息儲存實例支援 50,000 個使用者基礎，且已分散這些實例以取得負載平衡。每個伺服器有兩個 CPU，所以總共有 4 個 CPU。下圖顯示如何調整這個系統，以便為 250,000 個使用者和 2,000,000 使用者處理增加的負載。

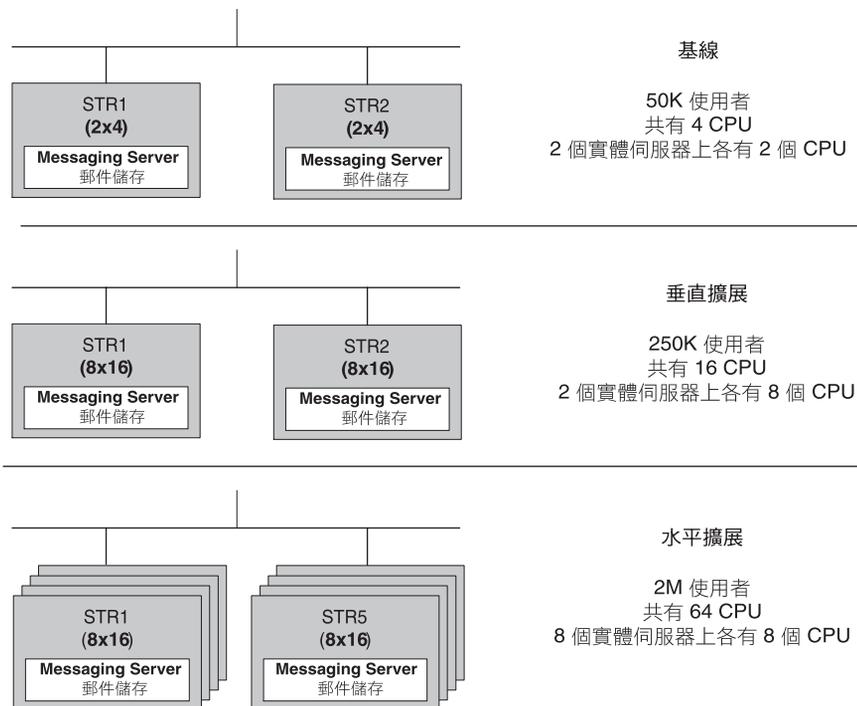
---

### 備註

[圖 5-10](#) 顯示水平調整和垂直調整的差異。這個圖表不會顯示調整時需考慮的其他因素，例如負載平衡、防故障備用和使用模式中的變更。

---

圖 5-10 水平和垂直調整的範例



## 找出效能瓶頸

成功部署設計的其中一個關鍵就是找出潛在的效能瓶頸，以及開發避免瓶頸的策略。當存取資料的速率無法符合指定的系統需求時，效能瓶頸就會發生。

瓶頸可分為不同的硬體類別，如以下系統中資料存取點表格所示。此表也會建議每個硬體類別中瓶頸的潛在解決方法。

**表 5-8** 資料存取點

硬體類別	相對的存取速度	效能改善功能的解決方法
處理器	十億分之一秒	垂直調整：增加更多處理能力，改善處理器快取能力 水平調整：增加平行的處理能力以進行負載平衡
系統記憶體 (RAM)	一百萬分之一秒	將系統記憶體指定給特定的作業 垂直調整：增加額外的記憶體 水平調整：建立額外的實例，以進行平行處理和負載平衡
磁碟讀取和寫入	毫秒	使用磁碟陣列 (RAID) 使磁碟存取最佳化 將磁碟存取指定給特定功能，例如唯讀或唯寫 快取系統記憶體中經常存取的資料
網路介面	因網路上的頻寬和節點的存取速度而異	增加頻寬 傳輸安全資料時，增加加速卡硬體 改善網路中節點的效能，使得能夠輕鬆隨時使用資料

**備註** 表 5-8 根據相對的存取速度列出硬體類別，代表慢速的存取點（如磁碟）可能就是造成瓶頸的原因。不過，無法處理大量負載的處理器也可能是造成瓶頸的原因。

一般而言，部署設計通常是從部署中每個元件及其相依性的基線處理能力估計開始。然後，您再決定如何避免與系統記憶體和磁碟存取相關的瓶頸。最後，檢查網路介面以決定潛在的瓶頸，再將焦點放在可以解決瓶頸的策略上。

## 最佳化磁碟存取

部署設計的關鍵元件是對於時常存取的資料集 (如 LDAP 目錄) 的磁碟存取速度。磁碟存取提供最慢的資料存取，可能是造成效能瓶頸的原因。

最佳化磁碟存取的方法就是將寫入作業和讀取作業分開。不單是因為寫入作業比讀取作業費時，而且讀取作業 (查詢 LDAP 目錄的作業) 發生的機率通常也比寫入作業 (更新 LDAP 目錄中的資料) 來得大。

另一個最佳化磁碟存取的方法是將磁碟指定給不同的 I/O 作業。例如，提供個別存取給 Directory Server 記錄作業，如交易記錄檔和事件記錄檔以及 LDAP 讀取和寫入作業。

此外，考慮實作一或多個指定給讀取和寫入作業的 Directory Server 實例，並使用分散到本機伺服器的複製實例以進行讀取和搜尋存取。也可以使用鏈接和連結選項來最佳化目錄服務的存取。

「Directory Server Deployment Planning Guide」的「System Sizing」章節 (<http://docs.sun.com/doc/817-7607>) 會討論規劃磁碟存取的各種因素。本章的主題包括：

- **最低的記憶體和磁碟空間要求。** 為各種大小的目錄提供所需的磁碟和記憶體估計。
- **調整實體記憶體以進行快取存取。** 根據 Directory Server 的使用規劃，提供估計快取大小和規劃總記憶體使用的指導。
- **調整磁碟子系統的大小。** 根據目錄尾碼和影響磁碟使用的 Directory Server 因素，提供規劃磁碟空間需求以及將檔案分散在磁碟間 (包括各種磁碟陣列選項) 的資訊。

## 設計最佳化資源使用

部署設計不只是估計滿足 QoS 需求需要的資源。在部署設計期間，您也要分析所有可用的選項，並選擇最佳的解決方案，這個解決方案必須能將成本降到最低但仍然可以滿足 QoS 需求。您必須分析每個設計決定的利弊，以確保一個領域中的獲利不會被另一個領域中的成本抵銷。

例如，可用性的水平調整可能會增加整體可用性，然而代價是增加維護和服務。效能的垂直調整可能會廉價地增加計算能力，但是額外的能力可能會被某些裝置沒有效率地使用。

在完成設計策略之前，審核決定以確認您已經從提議解決方案的整體利益平衡資源的使用。這個分析通常包含檢查一個區域的系統品質如何影響其他的系統品質。下表列出某些系統品質和資源管理的對應考量。

**表 5-9** 資源管理考量

系統品質	描述
效能	就集中 CPU 在個別伺服器的效能解決方案而言，服務是否能有效地利用計算能力？（例如，某些裝置對能夠有效使用的 CPU 具有數目限制。）
潛在容量	您的策略是否能處理超過效能估計的負載？ 在伺服器上使用垂直調整處理過度的負載，平分負載到其他伺服器，或是兩者同時使用？ 潛在容量足夠處理不尋常的尖峰負載，一直到您達成調整部署的下一個重大事件嗎？
安全性	您曾經充分地說明處理安全交易需要的效能費用嗎？
可用性	就水平備援解決方案而言，您曾經充分地估計長期維護的費用嗎？ 您曾經考慮過維護系統必要的排程停工嗎？ 您是否平衡過高階伺服器和低階伺服器的成本？
延展性	您曾經評估過調整部署的重大事件嗎？ 您是否有策略可提供足夠的潛在容量處理負載的設計增加，一直到您達成調整部署的重大事件為止？
服務性	您有將管理、監視和維護成本考慮到您的可用性設計中嗎？ 您曾經考慮過委託管理解決方案（讓一般使用者執行某些管理作業）來減少管理成本嗎？

## 管理風險

部署設計所根據的大部分資訊，例如服務品質需求和使用分析，並非經驗資料而是基於評估和規劃。可能造成這些設計不正確的原因很多，包括業務環境中未預期的情況、收集資料的方法錯誤或是人為疏失。在完成部署設計前，重新檢視設計所依據的分析，並確定您的設計考慮到任何來自評估或規劃的合理誤差。

例如，如果使用分析低估系統的實際使用，您將面臨無法應付流量的系統風險。在表現水準下的設計無疑將被視為失敗。

另一方面，如果您建立一個超出需要的多層級系統，則會將可用於別處的資源分散掉。關鍵在於將安全性的極限包含於需求之上，但避免浪費資源。

浪費資源可能也會導致設計失敗，因為沒利用到的資源原本應該應用在其他領域。此外，浪費的解決方案可能會被利害關係人視為不良的合約履行。

## 範例部署架構

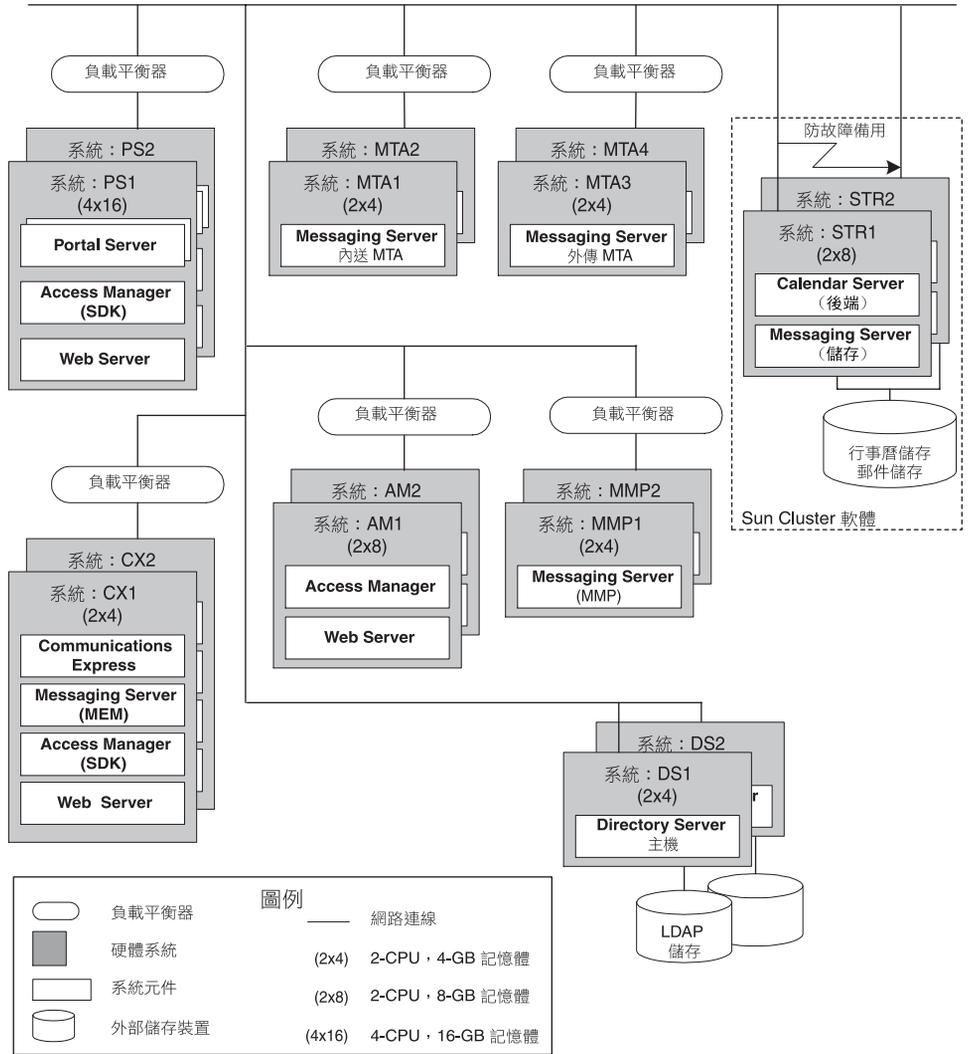
下圖表示本白皮書稍早介紹的範例部署的完整部署架構。本圖提供一個表現部署架構的概念。

---

**警告** 下圖中的部署架構只限於圖解說明使用。其並不代表已實際設計、建立或測試過的部署，而且不應視為部署規劃的建議。

---

圖 5-11 範例部署架構



## 範例部署架構

# 實作部署設計

在解決方案生命週期的實作階段，您可以執行部署設計時建立的規格與規劃來建立和測試部署架構，最後再將部署建置到生產中。雖然實作不在本指南涵蓋範圍內，本章仍提供此階段的高階檢視。

本章包含以下各節：

- [第 101 頁的「關於實作部署設計」](#)
- [第 102 頁的「安裝和配置軟體」](#)
- [第 103 頁的「開發引導和原型」](#)
- [第 103 頁的「測試引導和原型部署」](#)
- [第 104 頁的「建置生產部署」](#)

## 關於實作部署設計

在核准部署架構且完成了詳細的設計規格之後，您便進入解決方案生命週期的實作階段。實作是一套複雜的過程和程序，您必須仔細地規劃，才能確保實作可以順利完成。實作包括下列作業：

- 建立網路和硬體基礎架構
- 根據安裝規劃來安裝和配置軟體

- 將資料從現有的應用程式遷移到目前的解決方案
- 實作使用者管理規劃。
- 根據測試規劃，在測試環境中設計和部署引導或原型
- 根據測試規劃，設計和執行功能和加強測試
- 根據建置規劃，將解決方案從測試環境建置到生產環境
- 根據訓練規劃，訓練部署的管理員和使用者

實作的詳細資訊不屬於本指南的涵蓋範圍。不過，下列章節會提供這些作業的概要資訊。

## 安裝和配置軟體

為分散式企業應用程式安裝和配置 Sun Java™ Enterprise System，您必須規劃和協調許多作業及程序。在部署設計階段期間，您可以根據高階部署架構來建立安裝規劃，此架構可提供安裝 Java Enterprise System 軟體所需的安裝和配置資訊。

此安裝規劃的重點包括：

- 決定安裝的順序和類型
- 調查先前安裝軟體的主機和安裝是否已準備就緒
- 為每個您正在安裝的 Java Enterprise System 元件收集配置資訊

「Java Enterprise System 安裝指南」(<http://docs.sun.com/doc/819-0811>) 提供如何收集安裝規劃的詳細資訊。安裝指南會提供您可以用來記錄資訊的詳細配置資訊及試算表。安裝指南會提供一般安裝方案的指導，這些方案包含數個 Java Enterprise System 元件。如需更多資訊，請參閱「Java Enterprise System 安裝指南」中的「準備安裝」一節(<http://docs.sun.com/doc/819-0811>)。

## 開發引導和原型

Java Enterprise System 部署通常分為兩種類型，主要根據 Java Enterprise System 提供的服務類型，以及整合 Java Enterprise System 服務的自訂服務的有效數量。您可將前者的部署類型視為 80:20 的部署 (Java Enterprise System 提供 80% 的服務)，同樣地，前者為 20:80 部署。

就 80:20 部署而言，在實作階段期間，您通常會開發一個引導部署以供測試。因為 80:20 部署使用成熟的 Java Enterprise System 服務，其提供「基本即開即用」的功能，引導部署相對上會快速地從開發、測試和修改步驟移動到產品部署。引導部署驗證解決方案的功能，但也會提供系統執行情況的資訊。

在另一方面，20:80 部署會介紹新的自訂服務，其中並未包含來自 80:20 部署的互通功能歷程。為此，您建立一個原型，這是概念驗證部署，在成為產品前通常需要一個更嚴格的開發、測試、修改循環。原型可讓您決定提議的解決方案在測試環境中解決問題的能力。一旦原型證明該功能運作正常且功能完整，您就可以進行更嚴格的測試，然後再移至引導部署。

---

**備註** 實際企業部署在其需要的服務的自訂開發數量方面可以有非常大的變化。您使用引導或原型部署進行測試的方法，會根據部署的複雜性和特性而定。

---

## 測試引導和原型部署

測試引導和原型部署的目的在於確定，在盡可能最好的測試條件下，部署是否能滿足系統需求並符合業務目標。

理想狀態下，功能測試應該根據所有定義過的使用實例塑造方案，發展一套公式來評量合格性。功能測試也可將限制性的部署納入測試使用者的選取群組中，決定是否可滿足企業需要。

加強測試可評量尖峰負載下的效能。這些測試通常使用一系列的模擬環境及負載產生器來評量資料和效能的處理能力。部署的系統需求通常是設計和通過加強測試的根據。

---

**備註** 功能和加強測試對於大型部署特別重要，其中的系統需求可能沒有完善定義，也沒有舊的實作可供基本估計，而部署需要大量的新開發。

---

測試可指出部署設計規格的問題，並且在您將部署建置到生產環境前可能包含數個設計、建立和測試重複。測試原型部署時，您可能會發現部署設計的問題，如此一來您可以重複解決方案生命週期中的早期階段來解決這些問題。

在進行引導部署之前，請確定您已經徹底測試過部署設計。引導部署顯示您已使用之前的許多測試來驗證部署設計。您在測試引導部署期間所發現的問題，一般而言，必須在部署設計中的參數說明。

因為測試無法完全模擬生產環境，但也由於部署解決方案的特性可能會演進和變更，所以您應該繼續監視部署的系統，以找出是否有任何領域需要調校、維護或服務。

## 建置生產部署

一旦引導或概念驗證部署通過測試標準，您已準備就緒可將部署建置到生產環境。一般而言，您會分階段建置至生產環境。分階段建置對影響大量使用者的大型部署特別重要。

分階段建置部署可從一小組使用者開始，最後擴展到使用者基礎，直到所有使用者都可使用部署。分階段的部署也可從有限的服務組開始，最後建置到其餘的服務中。分階段服務可協助隔離、識別和排除一項服務在生產環境中可能遇到的問題。

## NUMERICS

- 20:80 部署 19
  - 實作階段 103
- 80:20 部署 19, 103

## A

- Access Manager 52, 77

## C

- Calendar Server 52, 77
- Communications Express 52

## D

- Directory Proxy Server 52, 56
- DMZ
  - 外部存取區域 66

## I

- Instant Messaging 52

## J

- Java Enterprise System
  - 20:80 部署 19
  - 80:20 部署 19
  - 三個要素架構 49
  - 元件 51
  - 元件相依性 52
  - 存取元件 56
  - 安裝 102
  - 自訂服務 19
  - 系統服務 17
  - 服務 19
  - 建置生產部署 104
  - 遷移問題 20
  - 關於 17

## M

- Messaging Server 52
  - Message Multiplexor (MMP) 55, 56, 59, 77
  - Message Transfer Agent (MTA) 55, 59
  - Messenger Express Multiplexor (MEM) 55
  - 不同邏輯的服務 55
  - 使用實例 59
  - 負載平衡範例 87
  - 訊息儲存 (STR) 55, 59, 77
  - 範例邏輯架構 58

## N

## N

N+1 防故障備用系統 84

## P

Portal Server 52, 56  
Mobile Access 56  
Secure Remote Access 52, 56

## Q

QoS (服務品質需求) 38

## S

SLA 32  
Solaris  
支援 14  
修補程式 14  
Sun Cluster 軟體 86  
防故障備用範例 88

## W

Web 伺服器 53, 77

## 三畫

三個要素架構 49

## 四畫

元件相依性 52  
Web 容器支援 55  
內部存取區域 (企業內部網路) 66  
公司文化 31  
引導 103  
測試 103  
支援  
Solaris 14  
文件 12  
安裝指南 53, 102  
技術摘要 18, 50, 51  
概要 12  
水平備援系統 84

## 五畫

加強測試 103  
功能測試 103  
可用性  
N+1 防故障備用系統 84  
水平備援系統 84  
防故障備用 83  
服務品質需求 40  
負載平衡 83  
最佳化資源 96  
範例 87  
複製服務 83  
優先順序 41  
可用性策略  
決定 83  
外部存取區域 (DMZ) 66  
用戶端層  
多層架構模型 57  
目錄伺服器 52, 59, 77  
多重主機複製 89, 90  
單一主機複製 89, 90

## 六畫

- 多重主機複製 89
  - 範例 90
- 多層架構設計 56
- 存取區域 65
- 安全存取區域 67
- 安全性
  - 估計處理器需求 73
  - 服務品質需求 44
  - 最佳化資源 96
- 安裝 Java Enterprise System 102
- 安裝規劃 70

## 七畫

- 估計處理器需求 73, 74
  - 安全交易 80
  - 使用實例 77
  - 範例 75
- 作業規劃 (操作手冊) 71
- 作業階段 26
- 作業需求 30
- 技術需求
  - 可用性 40
  - 安全性 44
  - 延展性 42
  - 服務性 46
  - 服務層級需求 47
  - 效能 40
  - 潛在容量 46
- 技術需求階段 23
  - 使用分析 36
  - 使用實例 38
  - 服務品質需求 38
  - 關於 35
- 找出瓶頸
  - 部署設計 73
- 防故障備用 83
  - Sun Cluster 軟體 86

範例 88

## 八畫

- 使用分析 36
  - 影響部署設計 72
- 使用者管理規劃 71
- 使用實例 38
  - Messaging Server 範例 59
  - 估計處理器需求 77
  - 基於身份識別的通訊範例 63
  - 影響部署設計 72
- 使用模式 30
- 延展性
  - 估計成長 43
  - 服務品質需求 42
  - 最佳化資源 96
  - 策略 91
  - 範例 92
- 所有權成本 34
  - 影響部署設計 72
- 服務性
  - 服務品質需求 46
  - 最佳化資源 96
- 服務品質需求 38, 46
  - 在部署設計中的角色 69
  - 影響部署設計 72
- 服務層級合約 32
  - 需求 47
  - 影響部署設計 72
- 服務層級需求 47
- 表示層
  - 多層架構模型 57

## 九畫

- 建置規劃 71
- 負載平衡 83

## 十畫

範例 85  
風險管理 97

## 十畫

原型 103  
測試 103  
容錯系統 41  
效能  
找出瓶頸 93  
服務品質需求 40  
最佳化資源 96  
訊息佇列 52  
訓練規劃 71

## 十一畫

基於身份識別的通訊範例 62  
估計處理器需求 75  
使用實例 63  
專案核准 70  
排程指定作業 33  
處理器需求  
估計 74  
規管需求 31  
部署方案 49, 67, 69  
部署架構 70  
範例 98  
部署計劃  
解決方案生命週期 21  
關於 21  
部署規劃  
遞增方式 32  
部署設計  
方法 73  
因素 71  
使用分析 72  
使用實例 72

所有權成本 72  
服務品質需求 72  
服務層級合約 72  
專案核准 70  
處理器需求 73  
業務目標 72  
複製服務 73  
輸出資料 70  
關於 69  
部署設計階段 24  
部署實作階段 25

## 十二畫

最佳化  
資源使用 96  
磁碟存取 95  
最佳化資源  
部署設計 73  
單一主機複製 89  
範例 90  
測試  
引導和原型 103  
加強測試 103  
功能測試 103  
測試規劃 71

## 十三畫

損壞修復規劃 71  
業務分析階段 23  
關於 27  
業務目標  
定義 28  
影響部署設計 72  
業務服務層  
多層架構模型 57  
業務限制 33

- 所有權成本 34
- 排程指定作業 33
- 預算限制 34
- 遷移問題 33
- 業務需求
  - 公司文化 31
  - 安全性目標 31
  - 作業需求 30
  - 使用模式 30
  - 定義 28
  - 服務層級合約 32
  - 規管需求 31
  - 業務目標 28
  - 瞭解使用者 29
- 解決方案生命週期 21
  - 作業階段 26
  - 技術需求階段 23, 35
  - 部署設計階段 24, 69
  - 部署實作階段 25
  - 業務分析階段 23, 27
  - 實作階段 101
  - 邏輯設計階段 24, 49
- 資料層
  - 多層架構模型 57
- 預算限制 34

## 十四畫

- 實作規格 70
- 實作規劃 70
- 實作階段 103
  - 開發引導和原型 103
  - 關於 101
- 管理風險 97
  - 部署設計 73

## 十五畫

- 潛在容量 46
  - 延展性考慮 92
- 範例
  - Messaging Server 邏輯架構 58
  - 可用性設計 87
  - 目錄伺服器 89
  - 多重主機複製 90
  - 存取區域 65
  - 估計安全交易的處理器需求 80
  - 估計處理器需求 75
  - 防故障備用 88
  - 延展性 92
  - 負載平衡 85, 87
  - 基於身份識別的通訊 62
  - 部署架構 98
  - 單一主機複製 90
  - 複製服務 89
  - 邏輯架構 58
- 複製服務 73
  - 可用性策略 83
  - 目錄伺服器範例 89
- 遷移問題 20
  - 業務限制 33
- 遷移規劃 70

## 十六畫

- 操作手冊 71

## 十七畫

- 應用程式伺服器 52

## 二十三畫

邏輯架構 49

基於身份識別的通訊範例 62

設計 50

影響部署設計 71

範例 58

邏輯設計

關於 49

邏輯設計階段 24

邏輯層

多層架構模型 57