



Sun Java™ System

Sun Java Enterprise System

部署規劃白皮書

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

文件號碼 : 817-7595

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 版權所有。

本文件所介紹產品中涉及的技術的相關智慧產權歸 Sun Microsystems, Inc. 所有。需特別指出的是 (但不僅限於)，這些智慧產權可能包含 <http://www.sun.com/patents> 上列出的一項或多項美國專利以及在美國和其他國家 / 地區的一項或多項其他專利或待批的專利申請。

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。

本發行軟體可能包括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 商標的使用均已獲得許可，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) 圖形使用者介面由 Sun Microsystems, Inc. 為其使用者和被授權者開發。Sun 感謝 Xerox 在研究和設計電腦業中視覺化或圖形使用者介面這個觀念上所作的開創成就。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。

本說明文件以「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為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

圖目錄	7
表目錄	9
第 1 章 部署規劃簡介	11
關於 Java Enterprise System	12
Java Enterprise System 套裝服務	13
Java Enterprise System 的優點	15
關於部署規劃	17
商業分析階段	18
技術需求階段	18
邏輯設計階段	19
部署設計階段	19
實施階段	20
第 2 章 商業分析	21
商業需求	21
商業限制	24
遞增方式的部署	25
第 3 章 技術需求	27
使用分析	28
使用實例	30
系統需求	31
可用性	32
容錯系統	32
Sun Cluster 3.1 4/04	33
服務的優先可用性	33
潛在容量	34
效能	34

延展性	35
安全需求	36
認證	37
授權	37
識別管理	37
可服務性需求	38
服務層級需求	39
第 4 章 設計邏輯架構	41
部署規劃範例	42
Java Enterprise System 服務	43
範例部署的邏輯架構	47
範例部署的資料流程	48
部署方案	49
第 5 章 設計部署架構	51
調整規劃部署大小	52
調整效能大小	52
決定使用者進入點的基線 CPU 估計	53
調整服務依賴性的 CPU 估計	54
調整潛在容量、延展性和可用性的 CPU 估計	56
調整安全性大小	58
計算安全交易的效能	59
處理 SSL 交易的專門硬體	60
調整可用性大小	61
複雜系統的目錄設計	61
硬體和軟體錯誤	62
可用性的一般方式	62
範例部署的可用性設計	65
可服務性問題	66
調整延展性大小	67
潛在容量	67
升級系統容量	67
最佳化資源	68
風險管理	68
管理資源	68
範例部署架構	70
詳細設計規格	72
第 6 章 實施部署設計	73
開發引導和原型	74
測試引導和原型部署	74

轉出產品部署 75

索引 **77**

圖目錄

圖 1-1	部署規劃階段	17
圖 3-1	技術需求階段以及其他部署規劃階段	28
圖 4-1	與其他部署規劃階段相關的邏輯設計	42
圖 4-2	Java Enterprise System 元件	44
圖 4-3	Java Enterprise System 邏輯架構中的元件	47
圖 4-4	範例部署的資料邏輯流程	48
圖 5-1	提供使用者進入點的元件的基線 CPU 估計	54
圖 5-2	為支援服務調整的 CPU 估計	56
圖 5-3	效能圖例包括記憶體需求	57
圖 5-4	計算安全交易的 CPU 估計的試算表	60
圖 5-5	單一伺服器	62
圖 5-6	兩個重複伺服器	63
圖 5-7	在兩個伺服器之間分配負載	63
圖 5-8	在 n 個伺服器之間分配負載	64
圖 5-9	範例部署中的 Calendar Server 的可用性設計	66
圖 5-10	範例部署架構	71

表目錄

表 1-1	Java Enterprise System 元件	12
表 1-2	Java Enterprise System 套裝服務	14
表 1-3	Java Enterprise System 優點	15
表 2-1	分析商業需求的主題	22
表 2-2	分析商業限制的主題	24
表 3-1	使用分析主題	29
表 3-2	系統品質影響部署設計	31
表 3-3	系統執行一整年 (8,760 小時) 的當機時間	32
表 3-4	服務的優先可用性	33
表 3-5	延展性考慮	35
表 3-6	可服務性需求的主題	38
表 4-1	範例部署的使用實例	43
表 4-2	Java Enterprise System 元件相互依賴性	45
表 4-3	Java Enterprise System 元件支援範例使用實例	46
表 4-4	額外元件支援範例使用實例	46
表 5-1	支援服務的 CPU 估計	55
表 5-2	資源管理主題	69

部署規劃簡介

本白皮書提供基於 Sun Java™ Enterprise System 的規劃大規模部署的簡介。其提出部署規劃的一些基本概念和原則並且介紹當設計企業範圍內的部署時，您能初步使用的一些過程。

如果您是根據 Java Enterprise System 評估 Java Enterprise System 或者規劃建立及部署大規模應用程式，請使用這篇報告作為部署規劃過程的指南。

本章提供簡要的 Java Enterprise System 概觀，並介紹在稍後章節會討論到的部署規劃概念。本章包含以下各節：

- [第 12 頁的「關於 Java Enterprise System」](#)
- [第 17 頁的「關於部署規劃」](#)

關於 Java Enterprise System

Java Enterprise System 是一套軟體架構，其提供的服務支援跨越網路或網際網路環境發佈的企業級應用程式。下表列出 Java Enterprise System 的元件，以及其提供的架構服務。

表 1-1 Java Enterprise System 元件

系統元件	提供的服務
Application Server	提供 Java 2 平台版、企業版 (J2EE™ 平台版) 容器服務給 Enterprise JavaBeans™ (EJB) 元件，例如階段作業 Bean、實體 Bean 和訊息導引 Bean。該容器提供需要的架構服務給緊耦合的分散式元件以產生互動，使其成為電子商務應用程式和 Web 服務執行與部署的平台。Application Server 也提供 Web 容器服務。
Calendar Server	提供一般使用者和群組使用者行事曆和排程服務。Calendar Server 提供與伺服器互動的瀏覽器架構用戶端。
Directory Proxy Server	從公司防火牆外部提供對 Directory Server 的安全服務。Directory Proxy Server 提供增強的目錄存取控制、綱目相容性、路由和多個 Directory Server 實例負載平衡。
Directory Server	提供用於儲存和管理企業內部網路、網際網路資訊的中央儲存庫，例如身份設定檔 (員工、客戶、供應商等等)、(公開金鑰證書、密碼及 PIN 號碼)、存取權限、應用程式資訊和網路資源資訊。
Identity Server	提供存取管理及數位識別管理服務。存取管理服務包括認證 (包括單一簽入) 和存取應用程式及 / 或服務的角色架構授權。管理服務包括對個別使用者設定檔、角色、群組以及策略的集中管理。
Instant Messaging	提供一般使用者之間的安全的、即時通訊，例如即時訊息傳送 (聊天)、會議、警示、新聞、輪詢和檔案傳輸。服務包括線上狀態管理者 (可告知使用者目前誰在線上) 和瀏覽器架構的用戶端 (可與伺服器互動)。
Message Queue	提供鬆耦合的分散式元件與應用程式間可靠、非同步的訊息。Message Queue 實施 Java Message Service (JMS) API 規格並新增企業功能，例如安全性、延展性及遠端管理。
Messaging Server	提供安全可靠並可大容量儲存與轉寄的訊息傳送功能，可支援電子郵件、傳真、傳呼、語音和影像。其可同時存取多個郵件儲存區並提供內容過濾，以協助拒絕垃圾電子郵件和預防病毒的攻擊。
Portal Server	為存取企業應用程式或服務的瀏覽器架構用戶端提供主要門戶網站服務 (如內容集成和個性化)。Portal Server 也提供可配置的搜尋引擎。

表 1-1 Java Enterprise System 元件 (續)

系統元件	提供的服務
Secure Remote Access	提供從公司防火牆外部對 Portal Server 內容和服務 (包括內部門戶網站或網際網路應用程式) 的安全網際網路存取。
Web Server	為 Java web 元件提供 J2EE 平台 web 容器服務，例如 Java Servlet 和 JavaServer Pages™ (JSP™) 元件。Web Server 也支援其他 web 應用程式技術以傳送靜態和動態內容，例如 CGI 程序檔和 Active Server Pages。
Sun Cluster	為 Java Enterprise System 提供高可用性和延展性的服務，執行在 Java Enterprise System 架構頂層的應用程式，以及兩者均部署的硬體環境。

Java Enterprise System 套裝服務

Java Enterprise System 部署通常分為兩種一般類別，主要由 Java Enterprise System 提供的服務組成的類別，以及整合大量自訂部署服務和協力廠商應用程式的類別。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，後者為 *20:80 的部署*。

注意 實際企業部署在其需要的自訂部署服務數量方面可以有非常大的變化。

Java Enterprise System 因為其豐富的服務組合，特別適合 80:20 的部署。例如，相對而言，其可簡單的部署企業範圍內的通訊系統或是企業範圍的入口系統。

然而，對於需要自訂部署的部署來說，Java Enterprise System 仍具有提供建立及整合自訂開發的服務和應用程式的能力。

下列表格將 Java Enterprise System 元件分組為可以傳送企業部署的套件。有些元件位於一個以上的套件中。

表 1-2 Java Enterprise System 套裝服務

套裝	Java Enterprise System 元件
網路身份識別服務	Identity Server Directory Server Web Server
企業入口網站服務	Portal Server Secure Remote Access Identity Server Directory Server Application Server 或 Web Server
企業通訊服務和協作服務	Messaging Server Calendar Server Instant Messaging Identity Server Directory Server Application Server 或 Web Server
Web 服務和應用服務	Application Server Message Queue Web Server
可用性服務	Sun Cluster 3.1 4/04 Sun Cluster Agents

表 1-2 中所述的大部分套件可傳送 80:20 類型的部署。例如，企業通訊服務和協作套件可用於建立提供電子郵件、行事曆和即時訊息給一般使用者的部署，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服務套件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。

可用性服務套件提供高可用性給企業應用程式的大規模部署。如果您的企業應用程式需要執行在應用程式伺服器或 Web 伺服器中的 J2EE 平台服務的自訂開發，請使用 Web 和應用程式服務套件。

因為 Java Enterprise System 服務間的互通功能，您可對特定的企業需求建立您自己修改過的服務套件。

Java Enterprise System 的優點

企業部署有三個成功關鍵。

- 傳送時間
- 傳送成本
- 功能

Java Enterprise System 提供您達到成功關鍵的工具，如下表所列。

表 1-3 Java Enterprise System 優點

優點	描述
簡單好用	<p>Java Enterprise System 提供一般安裝程式，讓您方便安裝、配置及升級。</p> <p>Java Enterprise System 從一個整合性獨立開發的單一產品和介體提供一項轉換到整合的平台服務的系統，服務可使用小型的自訂來部署和配置。</p>

表 1-3 Java Enterprise System 優點 (續)

優點	描述
可預測的	<p>Java Enterprise System 版次週期考慮到 Java Enterprise System 元件中的相容性。當您升級到新版次時，您可以避免元件間的不相容性和錯誤結合。</p> <p>Java Enterprise System 元件使用一系列共享的平台元件，可讓服務更容易溝通。</p> <p>已排程版次的 Java Enterprise System 傳送模組提供部署規劃中的可預測性。</p>
可負擔性	<p>商業授權的 Java Enterprise System 單一價格單位模組可減少安裝和升級部屬的複雜性及成本。單一價格單位包括支援、維護和諮詢服務。</p> <p>OEM 和教育授權還可採用其他的價格模組。</p>

關於部署規劃

成功的部署規劃是仔細準備、分析和通過一系列階段設計的結果，如下列圖表說明。

圖 1-1 部署規劃階段

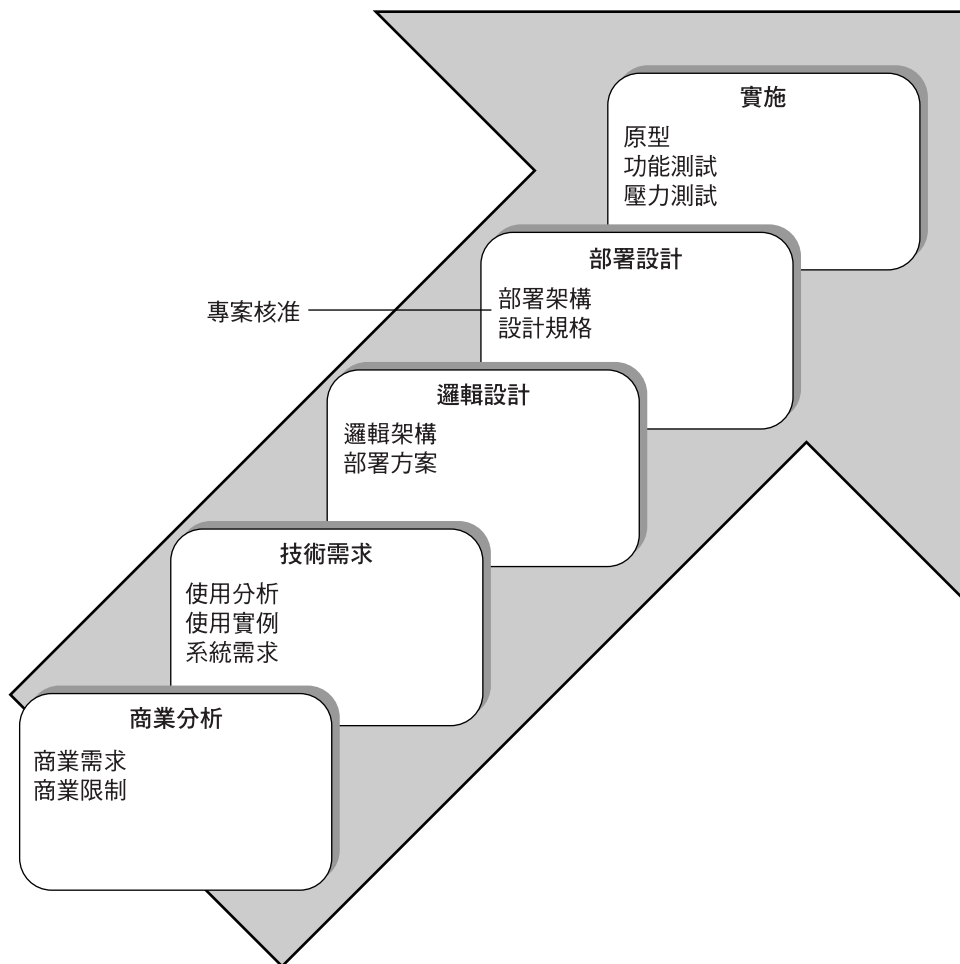


圖 1-1 中描述的每個階段都有自己的系列分析和程序，產生的規格和設計引導出隨後的階段。本章的下節提供每個部署規劃階段的摘要描述。

商業分析階段

在商業分析階段中，您定義部署專案的商業目標並確定達成目標必須符合的商業需要。當確定商業需求時，需考慮任何會影響達成商業目標能力的商業限制。商業分析階段會產生*商業需求文件*，您稍後會在技術需求階段使用到，並且依此評量部署設計的成功性。

如需有關商業分析階段的更多資訊，請參閱第 2 章，第 21 頁的「商業分析」。

技術需求階段

技術需求階段以您在商業分析階段建立的商業需求開始，並轉譯這些需求成為可用來設計部署架構的技術規格。在技術需求階段您要準備以下資訊：

- 使用者任務和使用模式分析
- 使用模型使用者與規劃部署互動的實例
- 系統需求來自商業需求，考慮使用者任務和使用模式的分析

使用分析、*使用實例*和*系統需求文件*的結果系列是邏輯設計階段的輸入資料。

在技術需求分析期間，您可能也會指定*服務層級需求*，其為部署系統不符合系統需求時，必須提供用於補救的顧客支援下的條目。服務層級需求是*服務層級協定*的基礎，在專案核准時簽訂。

如需有關技術需求階段的更多資訊，請參閱第 3 章，第 27 頁的「技術需求」。

邏輯設計階段

部署設計始於邏輯設計階段。在此階段中，您設計的邏輯架構提供 Java Enterprise System 服務和依靠，其滿足您在技術需求階段所定義的使用實例。

邏輯架構連同系統需求文件，描繪部署方案的特性。邏輯架構不指定實施部署方案需要的實際硬體。

如需有關邏輯設計階段的更多資訊，請參閱第 4 章，第 41 頁的「設計邏輯架構」。

部署設計階段

在部署設計階段，您建立一個部署架構，其將部署方案的藍圖表現為實體環境。實體環境是部署的網路架構，包含計算節點、每個節點的硬體需求、防火牆和其他網路設備。

對映的過程包含估計執行系統需求需要的實際硬體的部署，並決定符合預算考慮的最佳化部署架構策略。

部署專案的核准通常接在部署架構的建立之後。在專案核准期間，會評估部署的成本，如果核准，會簽訂部署的實施合約並取得建立專案的資源。

詳細的設計規格也是部署設計階段的一部份。設計規格提供實施部署架構需要的細節，例如實際硬體、作業系統、網路設計、和其他實體環境的層面。詳細的設計規格也包含指定供應使用者存取系統服務時需要的目錄服務資料架構。根據您部署專案的策略過程而定，設計規格會在專案核准前後產生。

如需有關部署設計階段的更多資訊，請參閱第 5 章，第 51 頁的「設計部署架構」。

實施階段

在實施階段期間，您會建立部署架構。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步驟：

- 在測試環境中建立和部署引導及 / 或模範部署
- 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
- 建立產品部署、將成品逐步執行

一旦部署成形，您必須持續監控、測試及調整部署以確保其符合商業目標。

如需有關實施階段的更多資訊，請參閱第 6 章，第 73 頁的「[實施部署設計](#)」。

商業分析

本章提供一些如何分析商業問題、識別商業需求和限制，以及表達商業目標的指南。

商業分析始於部署專案的商業目標確認。接著您要分析商業問題，您必須解決並識別必須達成的*商業需求*以達到商業目標。同時考慮任何限制您達成目標的*商業限制*。您識別出的商業需求和限制是您稍後在技術需求階段用來獲得系統需求的*商業需求文件*根據。

識別商業需求沒有簡單的公式，要根據您與客戶的合作以及您自己關於商業領域的知識來決定需求。這裡介紹的指南提供您一個開始商業分析的方式。

本章包含以下各節：

- [「商業需求」](#)
- [第 24 頁的「商業限制」](#)
- [第 25 頁的「遞增方式的部署」](#)

商業需求

商業問題的敘述與專案的執行摘要非常相似，可描繪出專案的最終目標。您在商業問題敘述中建立專案的商業實例（為何需要專案或值得一試）並定義專案範圍（哪些在專案範圍內，哪些在外）。您也要決定專案的哪些功能是其成功的關鍵。

商業需求分析的結果應該是一個定義部署如何滿足商業目標的文件。下表列出商業需求分析期間通常會著重的主題。

表 2-1 分析商業需求的主题

主题	描述
商業目標	<p>清楚表達專案的目標。清楚理解目標可幫助集中設計決定。</p> <p>這裡是一些範例目標：</p> <ul style="list-style-type: none"> • 企業合作，包括訊息傳送、通訊錄、即時訊息和行事曆服務的功能 • 企業入口網站，可讓使用者整合及個人化內容，並提供電子郵件存取、行事曆、即時訊息和其他企業服務。 • 企業資源排程程式，可排定會議室、辦公室和其他共用的實體資源 • 啓用線上商務 <p>用目前的作業對照規劃部署的目標可協助以後決定設計的判斷。</p>
部署類型	<p>確認以下哪些是您設想的部署類型：</p> <ul style="list-style-type: none"> • 企業到客戶 • 企業到員工 • 企業到企業 • 企業員工到員工通訊 • 某些上述類型的組合 <p>瞭解部署類型可幫助集中該類型固有的特定設計問題。</p>
範圍	<p>清楚指定專案的範圍。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。</p> <p>不清楚的定義範圍會導致部署設計無法充分滿足商業需求。</p>
利害關係人	<p>確認在部署的成功當中具有既得利益的個人和組織。</p> <p>所有的利害關係人都應該主動參加商業目標和需求的定義。</p>
關鍵特性	<p>定義成功關鍵的領域。這考慮到關於最重要標準的設計。</p>
鎖定使用者	<p>識別部署鎖定的使用者類型。例如：</p> <ul style="list-style-type: none"> • 現任和離職員工 • 活躍的客戶 • 會員場所 • 公眾 • 管理員

表 2-1 分析商業需求的主題 (續)

主題	描述
使用者的利益	<p>指出部署的使用者的預期利益。例如：</p> <ul style="list-style-type: none"> 遠端存取公司資源 企業合作 減少回應時間 減少錯誤率 單純化日常工作 由遠端團隊分享資源 增加產能 <p>清楚指出預期利益可幫助促成設計決定。</p>
服務層級合約	<p>您必須定義客戶支援的層級和範圍，否則部署會無法符合特定系統需求。</p> <p>一般而言，根據技術需求分析期間定義的服務層級需求，服務層級合約在專案核准期間簽訂。</p>
安全性問題	<p>您之前確認的目標可能有您不需要列在問題敘述中的潛在安全性問題。然而，調出部署所需的特定安全性目標可能會有幫助。例如：</p> <ul style="list-style-type: none"> 存取給授權的使用者之專屬資訊 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 與協力企業安全交易
優先性	<p>確定您目標的優先性。</p> <p>大型複雜的部署可能需要階段性實施。資源有限時可能需要排除或修改某些目標。透過清楚地確定優先性，您可以提供決策（可能需要為您的部署設計建立以便獲得承認）的指導。</p>

商業限制

商業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商業限制中找到符合商業需求的最佳方式。

下表列出可能會影響部署設計的典型商業限制。個別部署專案可能有其特定情況的商業限制。

表 2-2 分析商業限制的主題

主題	描述
時間範圍或排程	<p>部署的排程會影響您建立的設計決定。積極的排程可以會導致目標的縮小、改變優先性或是採取遞增的解決方法。</p> <p>在排程中，可能也有重要的里程碑值得考慮。</p>
預算限制	<p>大部分的部署必須依附特定的預算。在設計過程中永遠都要考慮到此預算以避免超過成本。</p> <p>當考慮預算時，記得不只有專案完成的成本，還有在特定的使用期間內維護專案需要的資源。</p>
資源	<p>考慮一個成功部署需要的所有資源，而不只是資金支出。這包括以下各點：</p> <ul style="list-style-type: none"> 現有的硬體和網路架構 信賴現有架構會影響系統設計。 需要開發資源來實施部署設計 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。此處的有限資源可能會影響您建立的設計決定。
所有權成本	<p>除了維護、管理和支援，您應該分析其他可能會影響所有權成本的因素。</p> <p>例如，可能需要的硬體和軟體升級，電源供應網上的足跡，電信成本和其他影響暫墊費用的因素。</p>
公司標準和政策	<p>確定瞭解組織要求部署的標準和政策。</p> <p>這些標準和政策可能會影響設計、產品選項和部署方法的技術面。</p>
公司改變經營	<p>改變經營的公司程序可能會戲劇化的影響部署方法和時間表。</p>
投資利潤	<p>每個部署都應該提供客戶投資利潤。投資利潤的分析通常包含資金支出獲得的經濟利益。</p> <p>估計部署的經濟利益包含部署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。</p>
規管需求	<p>規管需求有很大的程度是取決於部署的特性。</p>

遞增方式的部署

通常您會將部署視為一個全體、廣泛的系統。然而，您經常以穩健的步驟逐步完成廣泛的系統。

遞增方式提供這些優點：

- 您可以配合商業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施前的過渡階段
- 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

當採用遞增方式時，您通常會設計藍圖來提供引導到最終、廣泛解決方案的里程碑。此外，您可能要考慮規劃後期會採行的階段的短期解決方案。

不論您採取哪一種方式，您都應該設計一個容納變更與成長空間的部署。

遞增方式的部署

技術需求

本章探討技術需求分析中的一些程序與步驟。

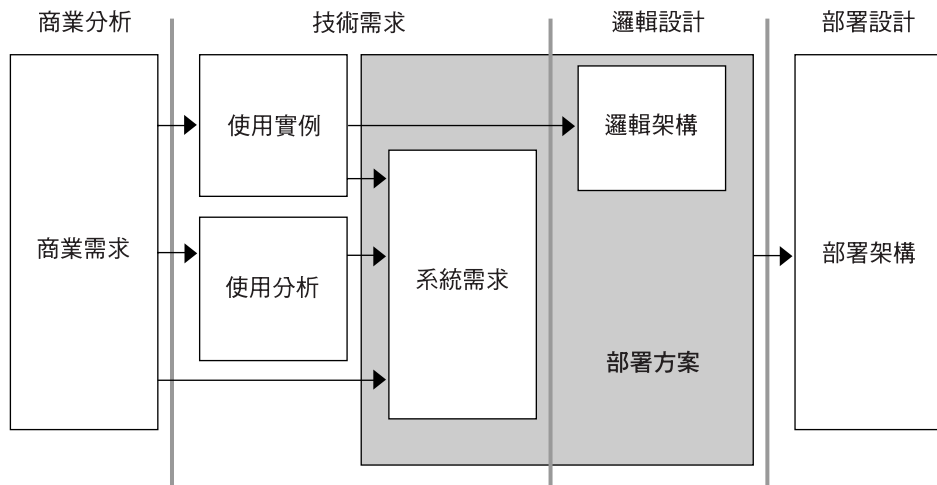
技術需求分析以商業分析階段期間所建立的商業需求文件開始著手進行。使用商業需求為基礎，執行以下步驟：

- 執行*使用分析*來協助決定部署的預期負載
- 建立一組塑造典型使用者與部署互動的*使用實例*
- 建立一組從商業需求、使用實例以及使用分析中得到的*系統需求*

使用實例也是在設計階段中設計*邏輯架構*的基礎。邏輯架構與系統需求一起形成稍後成為部署設計階段輸入資料的*部署方案*。

下圖顯示與商業分析、邏輯設計和部署設計階段相關的技術需求階段。

圖 3-1 技術需求階段以及其他部署規劃階段



在商業分析中，對於技術需求分析而言，並不存在產生使用分析、使用實例與系統需求的神奇公式。技術需求分析需要瞭解商業領域、商業目標以及系統科技的構成。

本章包含以下各節：

- 「[使用分析](#)」
- [第 30 頁的「使用實例」](#)
- [第 31 頁的「系統需求」](#)

使用分析

使用分析包含確認您設計的部署的各種使用者以及決定使用者的使用模式。您收集的資訊提供預期負載情況的概念並且稍後用於決定效能需求以及其他系統需求。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 30 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，也應該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主題。

表 3-1 使用分析主題

主題	描述
使用者的數量與類型	<p>決定您的部署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"> 「企業到消費者」部署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。 「企業到員工」部署通常必須要能容納每一個員工，不過有些可能需要從公司網路外部存取。 在「企業到員工」部署中，負責人可能需要得到一般員工不能存取的特定區域授權。
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元件互動的使用者。非作用中的使用者則可能是未登入系統的使用者或者是登入系統沒有與系統元件互動的使用者。</p>
管理使用者	<p>確認能進入部署系統監控、更新以及支援部署的使用者。</p> <p>決定任何可能會影響到系統需求的特定管理使用模式。例如：來自防火牆外部的部署管理。</p>
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"> 使用錯誤時是否有尖峰時段？ 正常上班時間為何？ 使用者是否分散全球？ 使用者連線的預期持續期間為何？
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"> 使用者會執行什麼工作？ 使用者何時登入，他們會維持登入狀態嗎？或者他們只是執行一些工作後就登出？ 使用者之間是否需要一般的行事曆、網路會議及網際網路網頁的部署等值得注意的協作？
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。現存應用的登入檔案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部署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。圍繞完整使用實例的優先設計確保持續集中焦點在預期功能的傳送上。

每個使用實例可包括關於使用者行為模式的數量估計，稍後您可以用來決定效能、可用性和其他服務品質的系統需求。使用實例也是設計邏輯架構的起始點，如第 4 章，第 41 頁的「設計邏輯架構」中所述。

您經常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工作。使用實例的權重可幫助決定系統需求。

使用實例可描述為兩種層級。

- 使用實例圖
行為者和使用實例之間關係的圖解說明。
- 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。

系統需求

系統需求敘述部署系統必須提供符合商業需求的服務品質，透過商業分析達成。您通常連同商業需求使用使用分析和使用實例以獲得系統需求。

下表列出經常用來指定系統需求的系統品質。

表 3-2 系統品質影響部署設計

系統品質	描述
可用性	對於一般使用者取得系統資源和服務的頻繁度測量，通常表現在系統的 正常執行時間 。
潛在容量	系統處理不尋常尖峰負載使用而無須額外資源的能力。
效能	回應時間以及與使用者負載狀況有關的潛在因素評量。
延展性	日後增加容量（以及使用者）到部署的系統上的能力。延展性通常包含增加資源到系統中，但不需要變更部署架構。
安全性	說明系統和其使用者健全性的因素複雜組合。安全性包括認證和授權使用者以及資訊的安全傳輸。
可服務性	部署系統可管理的舒適度，包括監控系統、修復產生的問題以及升級硬體和軟體元件的工作。

影響部署設計的系統品質互相之間息息相關。系統品質的需求可能會影響其他系統品質的需求和設計。例如，高階的安全性可能會影響效能，接著就影響可用性。增加額外的伺服器來應付可用性問題可能會影響維護成本（可服務性）。

瞭解系統品質如何產生關聯以及必須建立的交易，是設計能夠成功滿足商業需求和商業限制的系統最大的關鍵。

下節更進一步觀察影響部署設計的系統品質，提供公式化系統需求時的因素指導。也有關於服務層級需求的章節，其為一組特別用來建立服務層級需求的系統需求。

可用性

可用性是指定部署系統*正常執行時間*的一種方式。通常被測量為使用者存取系統的時間的百分比。系統無法存取的時間 (*當機時間*) 可能歸咎於硬體、軟體、網路錯誤或者任何其他造成系統當機的因素 (例如斷電)。大部分的狀況下，服務 (維護和升級) 的排程時間不考慮當機時間。

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」會明顯地影響可用性的部署設計。下表顯示增加可用性額外的「九」到系統的結果，系統為一整年 24x7 小時執行，合計為 8,760 小時。

表 3-3 系統執行一整年 (8,760 小時) 的當機時間

九	可用百分比	當機時間
二	99%	88 小時
三	99.9%	9 小時
四	99.99%	45 分鐘
五	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) 和軟體兩者的冗餘提供關鍵服務。

錯誤單點是一個沒有透過冗餘元件備份的硬體或軟體元件。此元件的錯誤會導致系統服務的損失。在設計容錯系統時，您必須確認潛在錯誤單點並且進行排除。

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

Sun Cluster 3.1 4/04

Sun Cluster 3.1 4/04 軟體為需要高可用性、容錯系統的部署提供了一個高可用性的解決方案。Sun Cluster 3.1 4/04 結合伺服器、儲存裝置和其他網路資源來提供故障轉移過程，完成快速並且只對系統的使用者造成極小的服務中斷。

服務的優先可用性

從使用者觀點來看，比起整體系統的可用性，可用性更常應用在部署系統提供的每個服務上。例如，如果即時訊息服務變成無法使用，對其他服務的可用性影響通常很小或根本沒有影響。然而，在其他許多服務所依賴的服務可用性之上（例如 Directory Server）則對系統有較大的影響。

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 服務的優先可用性

優先性	服務類型	描述
1	策略	作業所需的服務。例如，許多服務依賴 Directory Server。
2	關鍵任務	尖峰負載時必須能用的服務。例如，對應用程式的資料庫服務定義為關鍵任務。
3	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，Messaging Server 可用性在某些商業環境中可能不是關鍵。
4	可以延遲	在給定期間內可以使用的服務。例如，Instant Messaging 可用性在某些商業環境中可能不是必要。
5	選擇性	可以無限延遲的服務。

如需關於實施可用性需求的各種設計策略資訊，請參閱第 61 頁的「調整可用性大小」。

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，您不會直接圍繞著潛在容量來指定系統需求，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。

效能

決定效能需求是轉換效能上的商業需求預期為系統需求的過程。商業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 架構存取的商業需求可能強調以下：

使用者在登入時應該可預期合理的回應時間，通常不會大於四秒。

以商業需求開始，檢查所有使用實例來決定如何在系統層級表達此需求。考慮使用者負載條件，如同使用分析期間的決定。表示在*特定負載條件下的回應時間*或是*回應時間加上處理能力*方面每個使用實例的效能需求。您可能也指定允許的錯誤數量。

這是一個如何指定用於效能的系統需求範例。

在當日使用者登入的回應不得大於四秒，每隔 15 分鐘評量一次，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。

延展性

延展性描述日後增加容量以及使用者到系統上的能力。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**），延展性可能會有全系統的影響。

您不需要以系統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商業需求中。在部署設計階段期間，部署架構應該考慮調整系統比例，即使您未指定延展性需求。

決定延展性需求不是一項精密科學。評估系統的成長包含設計、評估和可能無法滿足的假設。這裡有建立延展性系統的三個關鍵。

- 採納高效能設計的策略。
在效能需求的規格和設計期間，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里程碑。
- 分階段實施部署。
遞增式的實施可協助排程額外的資源。
- 實施大規模的效能監控。
監控部署的效能可協助決定何時要新增資源到部署中。

下表列出某些要為延展性考慮的主題。

表 3-5 延展性考慮

主題	描述
使用分析	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。

表 3-5 延展性考慮 (續)

主題	描述
設定適當的里程碑	遞增式實施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里程碑。 例如： <ul style="list-style-type: none"> • 資金取得 例如季度或年度 • 硬體前置時間 例如，一到六週 • 緩衝 (10% 到 100%，根據成長預期)
吸收新興技術	瞭解新興技術，例如較快的 CPU 和 Web 伺服器，以及會如何影響基礎架構的效能。

安全需求

安全性是影響系統和使用者健全性的系統品質，包括使用者交易和相關資料的健全性。如同其他系統需求，商業需求、使用分析和使用實例驅動安全需求的分析。

安全需求的分析歸於以下類別：

- 認證
- 授權
- 識別管理

認證、授權和識別管理，連同屬於健全安全性實施的企業範圍策略，提供交易安全的信心並且儲存在網站上的資料不會被危及。

注意 影響架構 (例如防火牆軟體和網路設計) 健全的安全需求通常不會在系統需求分析期間考慮。反之，這些安全性問題在部署設計期間開始作用。

認證

認證是使用者如何向系統確認自己，同時系統向使用者確認自己的方式。認證是系統健全的關鍵部分，保護系統免於未認證的存取。

您應該瞭解使用者需求來選擇屬於部署的最佳認證方案。例如，「企業到客戶」部署可能允許使用者使用使用者名稱 / 密碼組合來註冊。這些使用者依賴可靠的認證機構發出的伺服器憑證，例如 VeriSign，驗證銷售系統經由安全傳輸。

「企業到員工」部署反而可能從現有的使用者基礎提供給員工。若來自公司防火牆內部，則允許存取到已知的安全位置。若來自公司防火牆外部，存取安全位置是經由執行認證的代理伺服器，並重新導入公司防火牆內部。

授權

授權是給認證使用者的特定權限識別。例如，具有管理員授權的使用者有部署系統的部份存取權，一般使用者則無法存取。

授權在部署實施單次登入 (SSO) 時也扮演重要的角色。部署認證的使用者具有多重服務的存取權，而不需要重複登入。

識別管理

部署系統必須有一套方法新增、修改或刪除存取系統服務的使用者。根據您的需要，識別管理可以透過授權的管理員完成，或是藉由授權管理介面的工具讓使用者自己達成。中型或大型企業的部署應該考慮授權管理設計。授權管理可增加顧客的滿意度並減少系統管理的成本。

可服務性需求

可服務性是部署系統可管理的舒適度，包括監控系統、修復產生的問題以及升級硬體和軟體元件的工作。

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工作。</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>確認部署的使用模式來決定維護機會時的視窗。</p> <p>例如，在尖峰使用通常為一般企業時間的系統上，機會的視窗出現在晚上或週末。對於地理分散的系統來說，確認這些時間會更具挑戰性。</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具有較小的維護、升級和修復的視窗。</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>
診斷與監控	<p>透過定期執行診斷和監控工具來確認問題區域，您可以改善系統的穩定性。</p> <p>這樣可以在發生前避免問題，根據可用性策略幫助平衡工作量並為維護和當機時間改善規劃。</p>

服務層級需求

*服務層級需求*是一組系統需求，指定在必須提供的客戶支援下的條件。服務層級需求是服務層級協定的基礎，通常在專案核准時簽訂。

如同系統需求，服務層級需求來自商業需求，並代表一種關於部署必須達到的整體系統品質的客戶擔保。因為服務層級協定是您和客戶之間的約束，服務層級需求的規格不能模稜兩可。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。

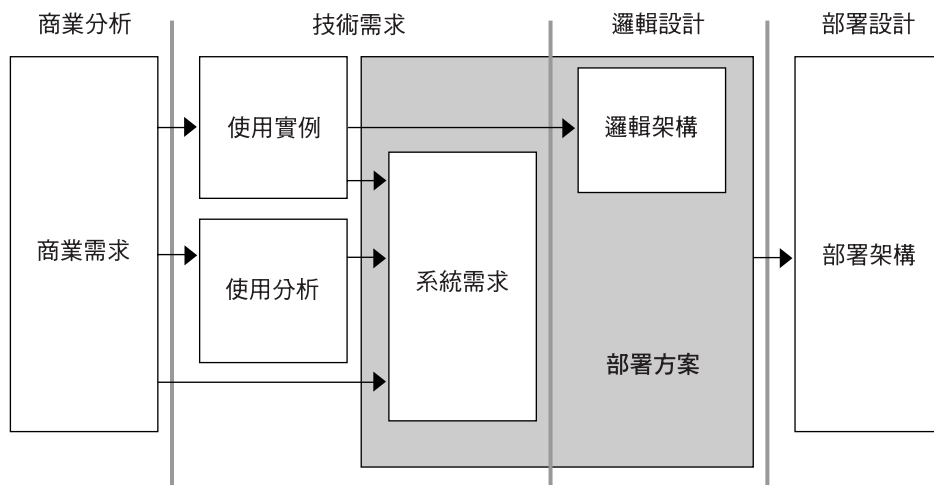
設計邏輯架構

本章討論建立邏輯架構的程序，並使用中型企業通訊部署的典型實例，提供您一個程序的範例。

邏輯架構確認 **Java Enterprise System** 元件（及其依賴性），以提供軟體服務符合部署的商業目標。一般而言會使用技術需求階段開發的實例指示需要哪一種軟體服務。然而，軟體服務上的資訊通常可直接從商業分析階段取得的商業需求獲得。

邏輯架構連同需求分析期間決定的系統需求，代表一個部署方案。部署方案是設計部署架構的基礎。下圖顯示邏輯設計階段到商業分析、技術需求和部署設計階段的關係。

圖 4-1 與其他部署規劃階段相關的邏輯設計



本章包含以下各節：

- 「[部署規劃範例](#)」
- [第 43 頁的「Java Enterprise System 服務」](#)
- [第 47 頁的「範例部署的邏輯架構」](#)

部署規劃範例

爲了協助圖解說明部署規劃程序，本節根據典型中型企業通訊需求介紹範例部署的使用實例。範例部署會持續出現在本白皮書後面的章節，圖解部署規劃的各種步驟。

警告 範例部署的使用實例、邏輯架構、部署架構和設計規格在部署規劃程序中爲簡化版的步驟。

範例爲了圖解說明的用途已經簡化過。範例的的設計不完整而且從未建立或測試過。請勿將範例當作您任何規劃部署的藍圖。

範例部署始於一組來自通訊部署的典型商業需求使用實例。下表摘要這些使用實例。

表 4-1 範例部署的使用實例

使用實例	描述
#1 單次登入	從 Web 瀏覽器，使用者登入系統（使用者名稱 / 密碼）以存取企業服務，服務可能為下任何之一： <ul style="list-style-type: none"> • 自訂入口網站網頁 • Web 架構電子郵件網頁 • 行事曆介面 • 安全性網頁
#2 開啓個人化入口網站畫面	從 Web 瀏覽器，使用者導覽至個人化入口網站畫面。
#3 透過入口網站，使用者可檢查電子郵件	從入口網站介面，使用者可檢查新的電子郵件訊息。
#4 透過入口網站，使用者可檢查安全性網頁	從個人化入口網站介面，使用者可檢查安全專案狀態網頁。
#5 透過入口網站，使用者可檢查行事曆	透過入口網站介面，使用者可檢查日常約會
#6 管理行事曆	從 Web 架構行事曆用戶端，使用者可排程約會。
#7 管理電子郵件	從電子郵件用戶端，使用者可讀取並傳送電子郵件。

從這些使用實例，您可以獲得邏輯架構需要的服務，如下列章節所述。

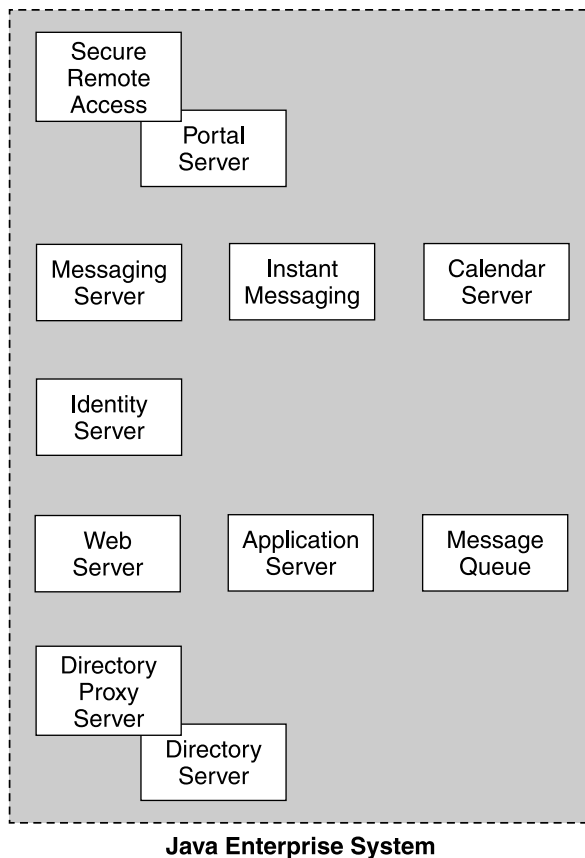
Java Enterprise System 服務

邏輯架構的設計從使用實例的分析開始，可以幫助您決定部署需要的服務。使用您的 Java Enterprise System 知識和從前的設計經驗，安排 Java Enterprise System 元件的初始邏輯設計，其可提供使用實例確認的服務。

當安排元件時，考慮系統內資料的邏輯流程以及提供服務的元件之間依賴性。您的邏輯設計應該反映這些會影響設計元件間資料流程的依賴性。

下圖顯示隨附於 Java Enterprise System 的元件。使用第 45 頁的表 4-2 的圖例瞭解 Java Enterprise System 元件的相互依賴性。一般而言，圖例底部的元件提供其上的元件支援。

圖 4-2 Java Enterprise System 元件



下表列出 Java Enterprise System 元件之間的實際相互依賴性。

表 4-2 Java Enterprise System 元件相互依賴性

Java Enterprise System 元件	提供支援到	依靠
Application Server	Identity Server Portal Server	Message Queue
Calendar Server	Portal Server (用於行事曆通道)	Directory Server Identity Server (用於單次登入) Messaging Server (用於 Calendar Server 電子郵件通知服務)
Directory Proxy Server	無	Directory Server
Directory Server	Administration Server Calendar Server Directory Proxy Server Identity Server Instant Messaging Messaging Server Portal Server	無
Identity Server	Portal Server 如果為單次登入配置： Calendar Server Instant Messaging Messaging Server	Directory Server Application Server 或 Web Server
Instant Messaging	Portal Server	Directory Server
Message Queue	Application Server	Directory Server (選擇性)
Messaging Server	Calendar Server Portal Server (用於訊息通道)	Directory Server Web Server Identity Server
Portal Server	Secure Remote Access	Directory Server Application Server 或 Web Server 如果配置為使用 Portal Server 通道： Calendar Server Messaging Server Instant Messaging
Secure Remote Access	無	Portal Server
Web Server	Identity Server Portal Server	無

例如，若要為範例通訊部署安排 Java Enterprise System 元件，則分析第 43 頁的表 4-1 中列出的使用實例。下表列出部署直接需要的元件，如使用實例所指出的一樣。

表 4-3 Java Enterprise System 元件支援範例使用實例

Java Enterprise System 元件	使用實例
Portal Server	#1 單次登入 #2 開啓個人化入口網站畫面 #3 透過入口網站，使用者可檢查電子郵件 #4 透過入口網站，使用者可檢查安全性網頁 #5 透過入口網站，使用者可檢查行事曆
Calendar Server	#1 單次登入 #5 透過入口網站，使用者可檢查行事曆 #6 管理行事曆
Messaging Server	#1 單次登入 #3 透過入口網站，使用者可檢查電子郵件 #7 管理電子郵件

您也需要決定哪一個 Java Enterprise System 元件需用來支援上述表 4-3 中列出的元件。下表列出這些額外元件。

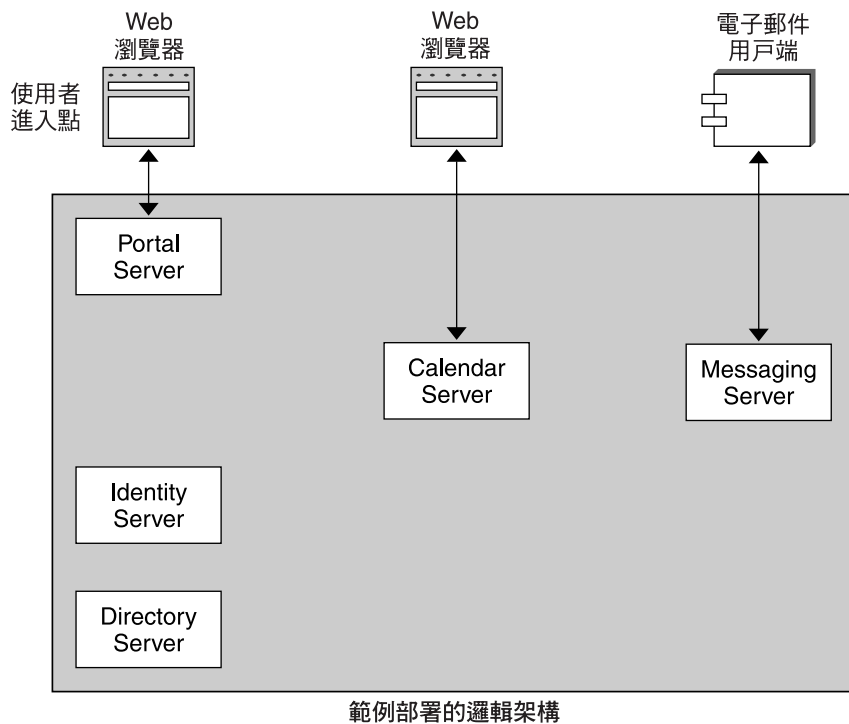
表 4-4 額外元件支援範例使用實例

Java Enterprise System 元件	提供的支援
Identity Server	提供支援到 Portal Server。 提供單次登入支援到 Calendar Server 和 Messaging Server。
Directory Server	提供支援到 Identity Server 和 Portal Server。
Application Server 或 Web Server	提供支援到 Identity Server 和 Portal Server。(Identity Server 和 Portal Server 必須在 Web 容器內部執行。)

範例部署的邏輯架構

以下的圖 4-3 顯示範例部署的元件安排，指出指向部署的使用者項目。此圖例將最需要支援的服務 (Portal Server) 放置在頂端並在其下列出支援元件，大致上反映元件之間的依賴性 (如第 45 頁的表 4-2 中所述)。此圖例未描述元件提供 Web 容器來支援 Portal Server 和 Identity Server，因為此依賴性不反映部署中的資料流程。

圖 4-3 Java Enterprise System 邏輯架構中的元件



未顯示支援 Portal Server 和 Identity Server 的 Web 容器

範例部署的資料流程

研究使用實例來決定邏輯架構中服務之間資料的邏輯流程，並在佈局中指出此流程。在調整效能和可用性時，系統中服務之間的資料流程扮演重要的角色，如第 52 頁的「調整規劃部署大小」中所述。

下圖描述範例部署的資料流程。資料流程從部署的使用實例及 Java Enterprise System 服務依賴性來決定。

圖 4-4 範例部署的資料邏輯流程

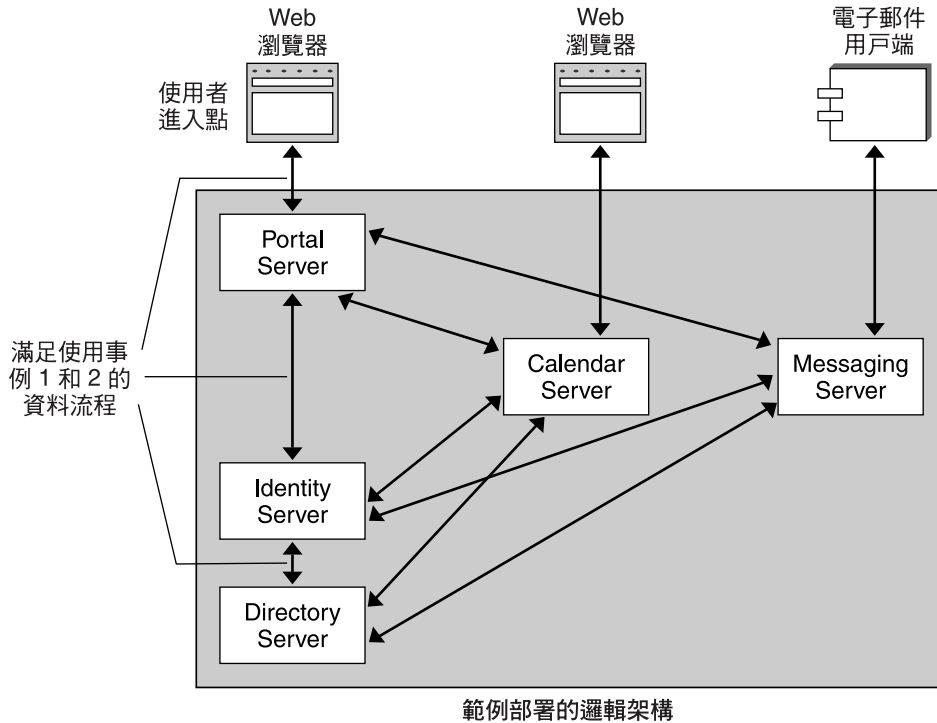


圖 4-4 叫出滿足實例 1 和 2 的資料流程。此資料流程表示以下各項：

- 來自 Web 架構用戶端的使用者登入請求
- 提供認證服務的 Identity Server 上 Portal Server 的依賴性
- 由 Directory Server 到 Identity Server 提供的 LDAP 資訊

圖 4-4 中剩餘的資料流程同樣從使用實例和伺服器依賴性中獲得。

部署方案

完整的邏輯架構設計和需求分析期間獲得的的系統需求，構成一個部署方案。部署方案是設計部署架構的起始點，如第 5 章，「設計部署架構」中所解釋。

設計部署架構

本章提供如何設計一個具有效能、安全性、可用性和其他系統品質的部署的資訊。本章也提供最佳化部署設計的資訊。

一個部署架構描述邏輯架構到實體環境的對映。實體環境包括企業內部網路或是網際網路環境中的計算機節點、CPU、記憶體、儲存裝置和其他硬體及網路裝置。

設計部署架構包含調整部署大小來決定必要的實體資源以符合技術需求階段中指定的系統需求。您也要透過分析調整部署大小的結果來最佳化資源，以便在商業限制內建立可提供最佳資源使用的設計。

在完成部署架構設計後，實際的部署成本會在專案核准期間評估。一旦核准專案後，需簽訂部署完成的合約並取得實施專案的資源。

詳細設計規格會在專案核准之前或之後產生。詳細設計規格用在實施階段建立設計時。

本章繼續使用第 4 章中的範例部署，圖解設計部署架構的過程中各種步驟。

本章包含以下各節：

- 第 52 頁的「調整規劃部署大小」
- 第 68 頁的「最佳化資源」
- 第 70 頁的「範例部署架構」
- 第 72 頁的「詳細設計規格」

調整規劃部署大小

調整規劃部署的大小是決定硬體資源（滿足系統需求所需要的，最終滿足商業目標）組合的過程。由於要配合其他規劃和設計部署的層面，調整大小不是一項精密科學，也無法以公式或處方來規定。成功的調整大小是過去設計經驗、系統架構的知識、領域知識，和應用創造性思维的組合結果。

調整大小以您之前為以下系統品質所決定的系統需求為中心，如第 31 頁的「系統需求」中所述。來自部署設計早期階段的商業需求、使用分析以及使用實例也在調整系統大小中扮演重要的角色。

當執行調整大小練習時，使用實例和使用分析可幫助決定支援使用實例需要的資源。您通常會從加權最重的使用實例開始（代表最常見的交易），然後繼續下去到加權最輕的。加權使用實例的使用可根據預期的系統重點來幫助分配資源。

以下章節提供如何為以下系統品質調整部署大小的一般指南：

- 效能
- 安全性
- 可用性
- 可服務性

調整效能大小

調整效能和負載需求大小是一項反覆的過程，其評估支援部署系統中服務需要的 CPU 的數目和對應記憶體。當評估支援服務需要的 CPU 數目時，考慮以下幾點：

- 使用實例和應用在服務的對應使用分析
- 分析技術需求期間決定的系統需求
- 過去使用 Java Enterprise System 元件提供服務的經驗
- 向具有調整各種類型部署方案經驗的 Sun 專業服務部門諮詢

調整效能的過程通常包含下列步驟。這些步驟的順序並不是關鍵 — 這只是提供一個考慮會影響最後結果的因素的方法。

1. 決定作為系統的使用者進入點元件的基線 CPU 估計。
2. 調整 CPU 估計來說明元件之間的依賴性。
3. 調整 CPU 估計來反映安全性、可用性、延展性和潛在容量需求。

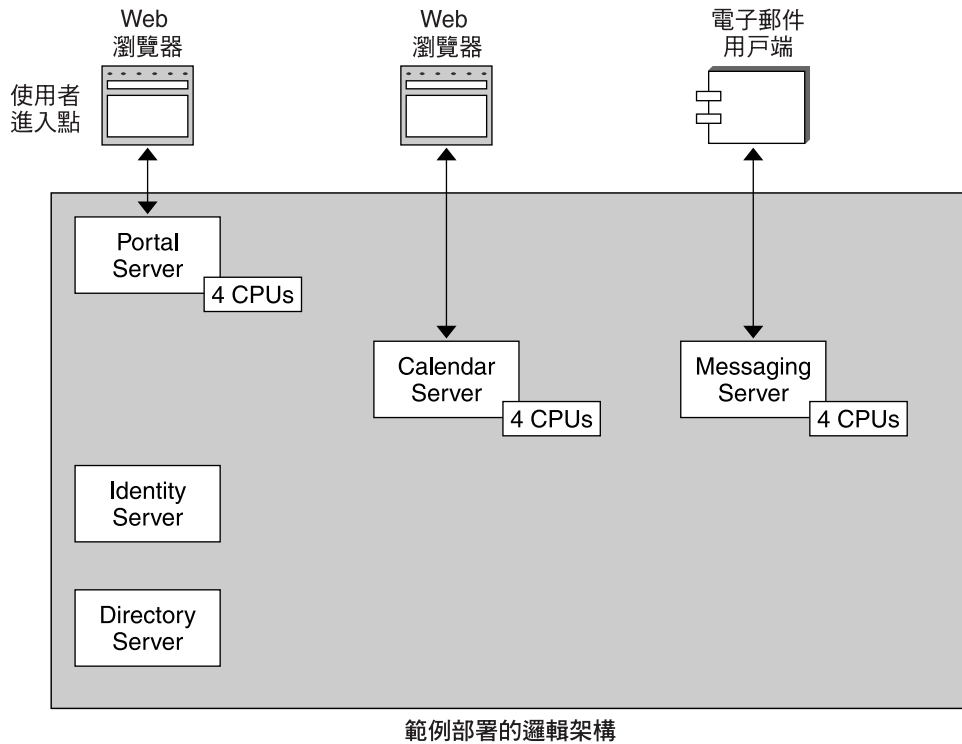
決定使用者進入點的基線 CPU 估計

從估計需要的 CPU 數量開始來處理每個作為使用者進入點的元件上預期的負載。注意您邏輯架構的佈局設計上的估計。

下圖使用第 42 頁的「部署規劃範例」中介紹的範例部署，描述作為使用者進入點的元件之初步 CPU 估計。這些估計代表可能從系統需求、使用實例和使用分析產生的數量。

警告 本白皮書不會指導您調整效能的細節。本手冊中使用的 CPU 和記憶體為隨意估計，僅限為圖解使用。除了圖解您可能在設計系統時會使用到的過程外，它們不代表任何特定的實施建議。

圖 5-1 提供使用者進入點的元件的基線 CPU 估計



調整服務依賴性的 CPU 估計

提供使用者進入點的元件需要 Java Enterprise System 服務的支援。若要繼續指定效能需求，調整效能估計以考慮其他元件需要的支援。

在範例中，檢查資料的邏輯流程，如第 48 頁的圖 4-4 中的圖解，並調整提供支援到其他元件的元件。下表總結對 CPU 估計的調整。在您的估計中，可以指定部分的 CPU。當完成效能估計時，將合計並集中 CPU 計數。

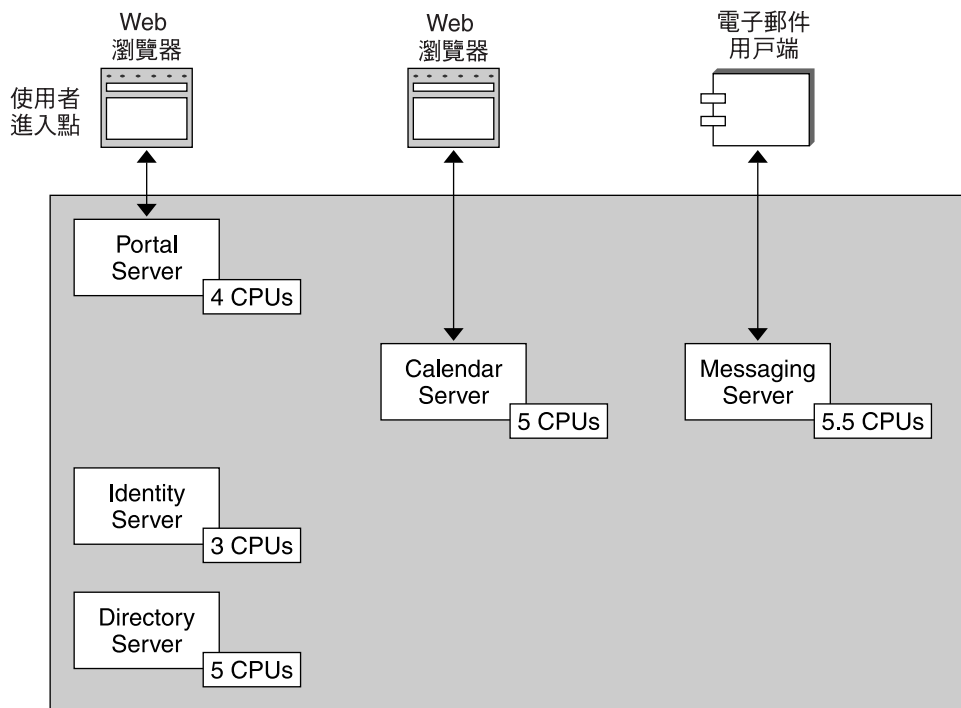
如同前一節使用的估計，下表中的效能估計為任意值，僅限於圖解使用。

表 5-1 支援服務的 CPU 估計

服務	估計	描述
Portal Server	無	未提供支援到其他服務。
Calendar Server	1 CPU	提供支援到： <ul style="list-style-type: none"> Portal Server 的行事曆通道
Messaging Server	1.5 CPU	提供支援到： <ul style="list-style-type: none"> Portal Server 的訊息傳送通道 Calendar Server 的電子郵件通知服務
Identity Server	3 CPU	提供支援到： <ul style="list-style-type: none"> Portal Server Calendar Server Messaging Server
Directory Server	5 CPU	提供支援到： <ul style="list-style-type: none"> Identity Server Calendar Server Messaging Server

下圖根據表 5-1 中的資訊更新效能的估計。

圖 5-2 為支援服務調整的 CPU 估計



範例部署的邏輯架構

調整潛在容量、延展性和可用性的 CPU 估計

一旦您完成效能的估計，集中 CPU 的數量。通常您要將 CPU 集中到下一個偶數。當集中 CPU 估計時，考慮以下的因素：

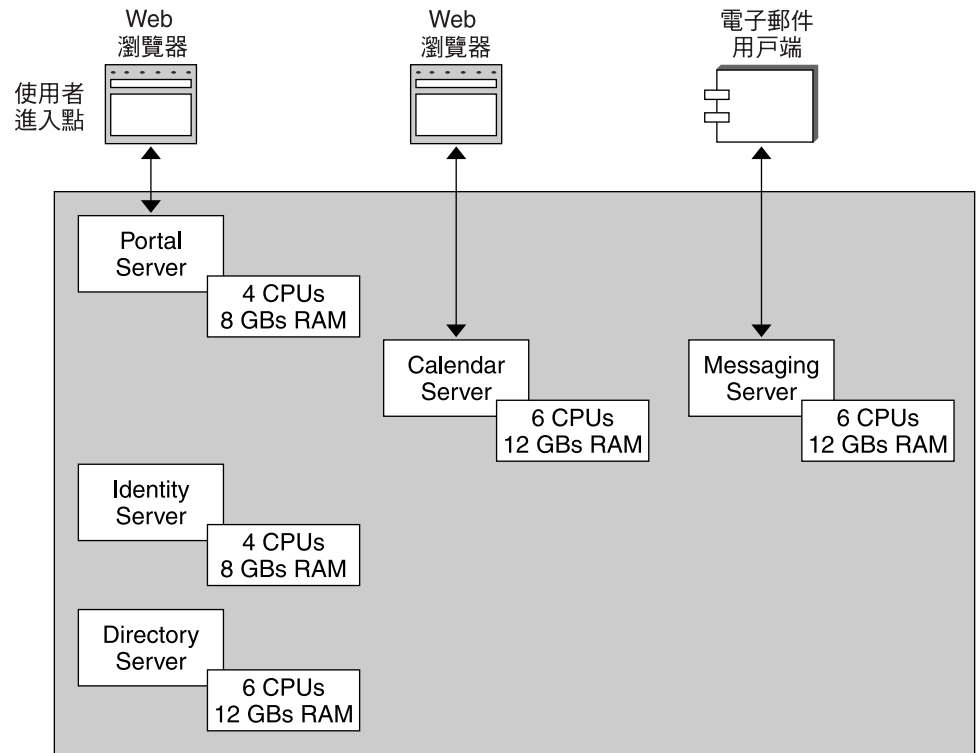
- 潛在容量
增加 CPU 估計以改善處理尖峰負載的能力。
- 延展性
增加 CPU 估計以確保您的部署不需要過早調整。注意調整的預期里程碑以及超時的計畫性負載增加，確定您允許足夠的潛在容量可達到里程碑。

- 可用性

調整 CPU 估計以承擔可用性或故障轉移需要的服務複製

下圖調整範例部署的 CPU 估計。圖例也指出每個 CPU 的記憶體需求。範例假定每個 CPU 需要 2GB 記憶體。這些範例的記憶體規格為任意值，只限於圖解使用。計算每個 CPU 的記憶體需求已經超過本白皮書的範圍。

圖 5-3 效能圖例包括記憶體需求



範例部署的邏輯架構

調整安全性大小

當調整部署大小時，安全性問題變成下列方式中的一項因素。

- 資料的安全傳輸
- 使用者的認證

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)**。使用者的認證也需要處理通過安全傳輸的交易。

通過安全傳輸處理的交易通常需要額外的計算能力，首先要建立安全作業階段（稱為 *訊號交換*），其次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 **Java Enterprise System** 服務而定，安全交易可能需要四倍（或更高）的計算能力。

當評估處理安全交易的效能需求時，首先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，修改 CPU 估計為安全交易需要的額外計算能力。

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。在許多實例中合理估計約有百分之五到十的交易需要安全傳輸。

例如，在線上電子商務網站瀏覽產品目錄時，直到客戶完成選擇並準備「結帳」前，所有的交易可能都是不安全的。同時許多這類的電子商務網站放鬆安全交易的潛在反應需求。然而有些使用方案，例如銀行或證券經濟商的部署則非常需要安全交易，並非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。

計算安全交易的效能

本節繼續以範例部署圖解使用實例的計算 CPU 需求，實例包括安全和非安全交易。

若要計算 CPU 需求，在試算表中建立以下計算：

1. 從 CPU 需求的基礎計算開始，如同您在前一節（第 52 頁的「調整效能大小」）所計算的。
2. 計算需要 SSL 的交易百分比，並計算 SSL 交易的 CPU 需求。
3. 調整非安全交易的 CPU 計算。
4. 清點安全和非安全需求以計算總 CPU 需求。

圖 5-4 中的試算表是根據 Portal Server 的額外使用實例和使用分析。額外使用實例和使用分析假定如下：

- 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等於非安全交易的效能需求。

此範例中，考量到計算處理 SSL 作業事件所需要的額外計算能力，處理作業事件的 CPU 數量將增加五倍。如同範例中其他的 CPU 計算，這是一個只限於圖例說明使用的隨意計算。

圖 5-4 計算安全交易的 CPU 估計的試算表

所有 Portal Server 交易的基線估計：4 CPUs

<p>1. 計算 SSL 交易的 CPU 估計：</p> <p>十分之一需要 SSL</p> $.10 \times 4 = 4$ <p>SSL 交易需要 5x CPU power</p> $5 \times .4 = 2 \text{ CPUs}$	2 CPUs						
<p>2. 調整非 SSL 交易的 CPU 估計：</p> <p>九成爲非安全</p> $.9 \times 4 = 3.6 \text{ CPUs (需要)}$	3.6 CPUs						
<p>3. 總 CPU 估計：</p> <table style="margin-left: auto; margin-right: auto;"> <tbody> <tr> <td>SSL</td> <td style="text-align: right;">2</td> </tr> <tr> <td>非 SSL</td> <td style="text-align: right;"><u>3.6</u></td> </tr> <tr> <td></td> <td style="text-align: right;">5.6 CPUs</td> </tr> </tbody> </table>	SSL	2	非 SSL	<u>3.6</u>		5.6 CPUs	5.6 CPUs
SSL	2						
非 SSL	<u>3.6</u>						
	5.6 CPUs						

處理 SSL 交易的專門硬體

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作業階段建立的計算能力，和 / 或資料的加密或解密。在使用專門的 SSL 作業硬體時，計算能力專用於 SSL 計算的某一部份，這部份通常是建立安全作業階段的「訊號交換」作業。

此硬體可能對您最後的部署架構有益。然而，因爲硬體的專門特性，最好先評估 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「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。

調整可用性大小

在您完成調整效能的大小後，您可以開始調整您系統的可用性大小。這是您指定特定伺服器主控邏輯架構中的元件之處，以及設計各種 Java Enterprise System 元件的負載平衡、冗位和故障轉移策略之處。

研究使用實例和使用分析來決定應該考慮何種可用性解決方案。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」？
- 有無指定故障轉移狀況的效能規格（例如在故障轉移期間最少有效能的 50%）？
- 使用實例和使用分析是否確認尖峰和非尖峰使用的時間？
- 潛在的效能需求為何？
- 是否有地理考量？

對每個元件，分析使用實例來決定故障轉移和負載平衡需求的最佳解決方案。同時，考慮使用實例和使用分析來決定負載平衡服務的最佳方式。

您選擇的可用性策略也要考慮服務性需求，如第 66 頁的「可服務性問題」中所討論。試著避免需要大量管理和維護的複雜解決方案，這樣有利於簡化管理系統。

複雜系統的目錄設計

大量使用者的複雜部署可能需要可以影響可用性策略的 Directory Server 目錄設計。這是因為 LDAP 目錄設計可能影響 Identity Server 和 Messaging Server 的可用性策略，依次影響到其他系統品質。

如果您設計的是複雜的部署，請考慮建立一個預備的目錄設計來協助可用性設計。之後，在詳細設計規格或開發階段期間，提供完整目錄設計。

硬體和軟體錯誤

您的可用性設計應該提供硬體和軟體錯誤的保護。軟體錯誤通常比硬體錯誤具有較高的成本。軟體錯誤之間比起硬體錯誤之間有較高的平均時間。此外，軟體錯誤較難診斷和修復並且需要較高的管理和維護成本來預防。

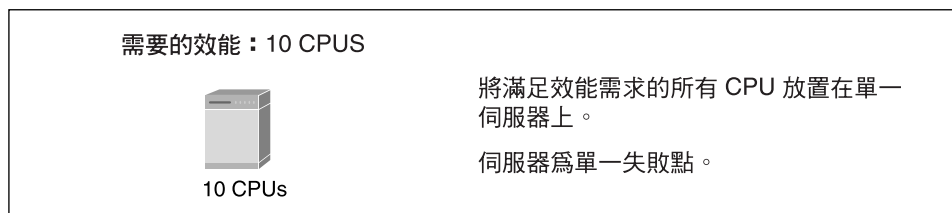
可用性的一般方式

本節提供您可以為可用性需求設計的某些一般方式。特定的可用性設計超過本白書的範圍。

單一伺服器系統

將您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 5-5 單一伺服器



Sun 提供高階伺服器，其提供下列效益：

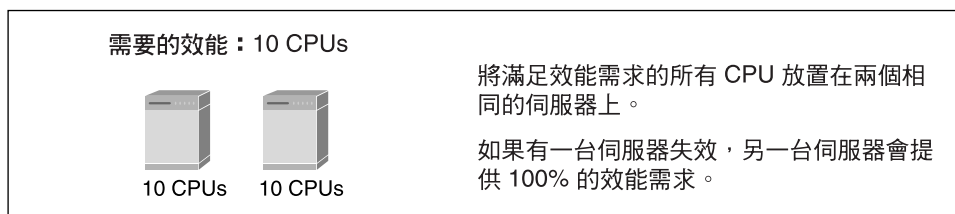
- 在系統執行時硬體元件可取代和重新裝配
- 在伺服器上的錯誤隔離網域中執行多個應用程式
- 無須重新啟動系統即可升級容量、效能速度和 I/O 組態

高階伺服器通常比較的多伺服器系統花費更大。然而，單一伺服器可以節約管理，監控和控管資料中心內伺服器的成本。然而，負載平衡、故障轉移和單一失敗點的移除比多伺服器系統更有彈性。

水平冗餘系統

有數種方式可使用平行冗餘伺服器（同時提供負載平衡和故障轉移）增加可用性。下圖說明兩個重複伺服器提供 $N+1$ 可用性系統。 $N+1$ 系統在一個伺服器錯誤時有額外的元件可提供 100% 的容量。

圖 5-6 兩個重複伺服器

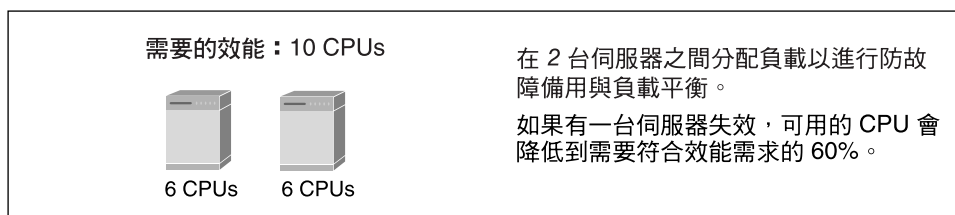


在上述圖 5-6 當中每個伺服器的計算能力是相同的。一個伺服器單獨處理效能需求。其他伺服器在呼叫服務做為備份時提供 100% 的效能。

重複伺服器設計的優點在於故障轉移狀況期間為 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。

下圖說明在兩個負載平衡和故障轉移的伺服器之間分配效能的方案。

圖 5-7 在兩個伺服器之間分配負載

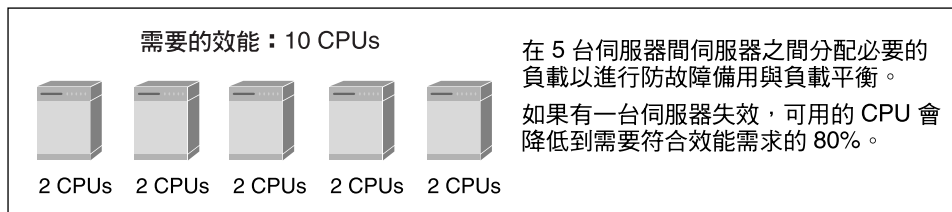


在上述圖 5-7 中，如果一個伺服器錯誤，所有服務仍然可使用，儘管只是全部容量的一部分百分比。剩餘的伺服器提供計算能力的 6 CPU，其為 10 CPU 需求的 60%。

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。同時如果一個伺服器錯誤，所有的服務都可使用，但是可能會減低效能。

下圖說明在一些伺服器之間的效能和負載平衡的分配。

圖 5-8 在 n 個伺服器之間分配負載



因為在圖 5-8 中的設計圖解說明有五個伺服器，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。資料中心內的伺服器也有主控成本。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

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 作為您的可用性設計的一部份。叢集系統是伺服器、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。

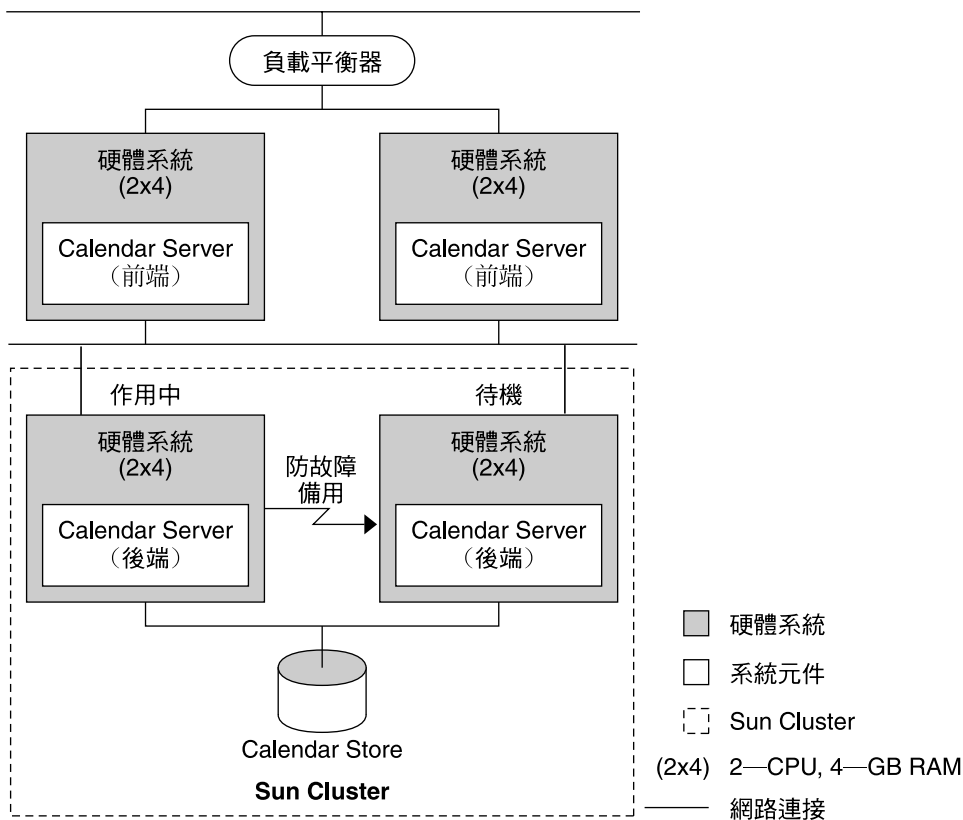
Sun Cluster 需要額外的專用硬體和專門技術來配置、管理和維護。

範例部署的可用性設計

下圖顯示範例部署的行事曆服務部分的可用性設計，說明在[第 4 章](#)，「[設計邏輯架構](#)」。圖例描述範例部署的邏輯架構的 Calendar Server 部分的可用性解決方案。範例部署的完整可用性解決方案分析已經超過本白皮書的範圍。

本章前面的調整大小練習決定 Calendar Server 需要 6 CPU 和 12 GB 的記憶體，如[第 57 頁的圖 5-3](#) 中所述。下圖顯示部署在兩個伺服器上的 Calendar Server 的前端用於負載平衡進出的要求。Calendar Server 的後端部署在獨立的伺服器上，並複製在 Sun Cluster 3.1 4/04 當中作為故障轉移用。為了故障轉移使用，Calendar Server 後端需要的 CPU 和記憶體複製在 Sun Cluster 3.1 4/04 當中。

圖 5-9 範例部署中的 Calendar Server 的可用性設計



可服務性問題

在設計可用性時，您也必須考慮解決方案的管理和維護成本。這些成本在設計中經常被忽略，因為並不會明確地與硬體採購綁在一起。當然，這些可能會被隱藏，花費的成長反映出您設計的複雜性。

例如，您的設計可能包括大量的可提供高度可用性的水平冗餘伺服器。但是如果您不將安裝和配置伺服器、持續升級軟體和監控系統運作狀態的成本列入重要因素，可能會危及可用性的取得。

在設計可服務性時，考慮下列管理和維護成本：

- 安裝和配置

- 監控
- 升級伺服器硬體
- 升級伺服器軟體
- 故障轉移自動化

調整延展性大小

延展性形容增加容量到您系統的能力，通常是系統資源的附加，但是不變更部署架構。本節討論設計延展性時考量的主題。

在需求分析期間，您通常會根據商業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。

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，以便於處理不尋常的尖峰負載。潛在容量是將安全性帶入您設計中的一種方式。

仔細分析使用實例能夠幫助確認可建立不尋常尖峰負載的方案（例如，一個可排程強制網路廣播的企業對員工的部署）。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。

您也可以監控在部署的系統中如何使用潛在容量，協助決定何時需要增加資源來調整系統。

升級系統容量

您的系統設計應該能夠處理計畫容量的前 6 到 12 個月的作業。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。依靠資源的仔細監控以及業務設計來決定何時要升級系統。

如果您執行的是增量式的部署，其中因為業務或技術原因而延遲的系統部分部署，可以將系統容量的升級排程與其他包含系統新功能的升級同時發生。

最佳化資源

調整部署大小不只是符合系統需求的資源評估。調整規模也是風險管理和資源管理的練習。一項設計如何處理風險管理和資源管理經常是達成商業目標的關鍵。

風險管理

調整大小所根據的大部分資訊，例如商業需求和使用分析，並非經驗資料而是基於評估和規劃。在完成規劃部署的調整前，重新檢視資料並確定您的調整設計考慮到任何來自評估或規劃的合理誤差。

例如，如果商業需求的設計低估系統的實際使用，您會擔負建立起一個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成功關鍵的領域。此外，浪費的解決方案可能會被視為不良的合約履行。

管理資源

管理資源是分析所有可用調整選項的過程，以及選擇成本最小化但仍滿足系統需求的最佳解決方案。這包括了瞭解每個設計決定的交易，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。

在完成調整策略之前，檢查您的決定確認您已經從設計的整體利益平衡資源的使用。這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。下表列出某些您可能要為資源管理考慮的主題。

表 5-2 資源管理主題

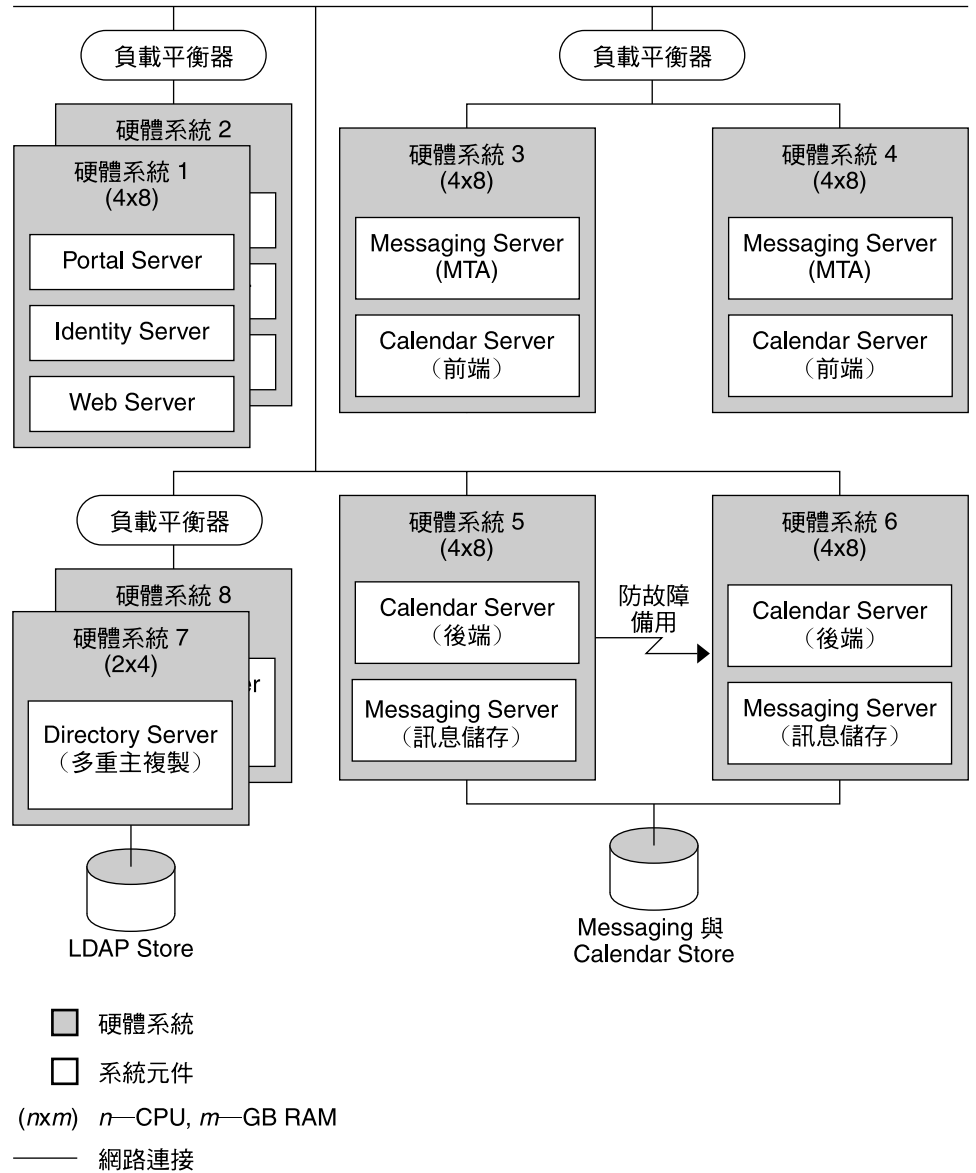
主題	描述
效能	就集中 CPU 在個別伺服器的效能解決方案而言，服務是否能有效地利用計算能力。(例如，某些裝置對能夠有效使用的 CPU 具有數目限制。)
潛在容量	<p>您有策略用來處理超出效能估計的負載嗎？</p> <p>在伺服器上使用垂直調整處理過度的負載，平分負載到其他伺服器，或是兩者同時使用？</p> <p>潛在容量足夠處理不尋常的尖峰負載，一直到調整部署的下一個里程碑嗎？</p>
安全性	您曾經充分地說明處理安全交易需要的效能費用嗎？
可用性	<p>就水平冗餘解決方案而言，您曾經充分地估計長期維護的費用嗎？</p> <p>您曾經考慮過維護系統必要的排程停工嗎？</p> <p>您是否平衡過高階伺服器和低階伺服器的成本。</p>
延展性	<p>您曾經評估過調整部署的里程碑嗎？</p> <p>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，一直到調整部署的里程碑出現之前？</p>
可服務性	<p>您有將管理、監控和維護成本考慮到您的可用性設計中嗎？</p> <p>您曾經考慮過委託管理解決方案（讓使用者自己執行某些管理工作）來減少管理成本嗎？</p>

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 5-10 範例部署架構



詳細設計規格

在部署架構完成後，有一段客戶審核的期間，接著是期待專案的核准。有時候客戶可能會在同意核准之前再要求您修改部署架構。

在專案核准後，您建立一個做為部署實施起始點的詳細設計規格。設計規格包括特定硬體資源和網路裝置的細節，以及詳細的 **LDAP** 目錄規格。

實施部署設計

本章提供實施部署設計所需步驟的概觀。

在部署架構核准後，並且完成了詳細的設計規格，您進入部署規劃的實施階段。在實施階段期間，您會建立部署架構。根據您部署專案的特性，實施部署設計包括以下一部分或是全部的步驟：

- 在測試環境中建立和部署引導或原型
- 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
- 建立產品部署、將成品逐步執行

本章包含以下各節：

- [第 74 頁的「開發引導和原型」](#)
- [第 74 頁的「測試引導和原型部署」](#)
- [第 75 頁的「轉出產品部署」](#)

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，後者為 *20:80 的部署*。

就 80:20 部署而言，在實施階段期間，您通常會開發一個 *引導部署* 用來測試。因為 80:20 部署使用成熟的 Java Enterprise System 服務，其提供「超出方塊」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。

20:80 部署，在另一方面，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 *原型部署*，其為概念驗證部署，在成為產品前通常需要一個更嚴格的部署、測試、修改循環。

注意 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。您是否使用引導或原型部署來測試根據您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商業目標。

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。

注意 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，也沒有舊的實施可供基本估計，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，並且在您將部署轉出到生產環境前可能包含數個設計、建立、測試重複。然而，測試不應該是您發現部署架構問題的地方。如果您在測試階段發現部署架構的設計問題，則您的分析、規劃及設計會被視為失敗。

轉出產品部署

一旦導引或概念驗證部署通過測試標準，您已準備就緒可將部署轉出至生產環境。一般而言，您會分階段轉出至生產環境。分階段轉出對影響大量使用者的大型部署特別重要。

分階段轉出部署可從小組的使用者開始，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，最後引入剩餘的服務中。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。

因為測試永遠無法完整模擬生產環境，您應該持續監控部署的系統以確認任何需要調整、維護或服務的區域。

