



# Sun Java System Application Server 9.1 部署規劃指南



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

文件號碼：820-4904

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 版權所有。

本產品或文件受版權保護，且按照限制其使用、複製、發行和反編譯的授權進行發行。未經 Sun 及其授權人(如果有)事先的書面許可，不得使用任何方法、任何形式來複製本產品或文件的任何部分。協力廠商軟體，包含字型技術，其版權歸 Sun 供應商所有，經授權後使用。

本產品中的某些部分可能源自加州大學授權的 Berkeley BSD 系統的開發成果。UNIX 是在美國及其他國家/地區的註冊商標，已獲得 X/Open Company, Ltd. 專屬授權。

Sun、Sun Microsystems、Sun 標誌、docs.sun.com、AnswerBook、AnswerBook2 與 Solaris 是 Sun Microsystems, Inc. 在美國及其他國家/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。凡具有 SPARC 商標的產品都是採用 Sun Microsystems, Inc. 所開發的架構。

OPEN LOOK 與 Sun<sup>TM</sup> Graphical User Interface (Sun 圖形化使用者介面) 都是由 Sun Microsystems, Inc. 為其使用者與授權者所開發的技術。Sun 感謝 Xerox 公司在研究和開發視覺化或圖形化使用者介面之概念上，為電腦工業所做的開拓性貢獻。Sun 已向 Xerox 公司取得 Xerox 圖形化使用者介面之非獨占性授權，該授權亦適用於使用 OPEN LOOK GUI 並遵守 Sun 書面授權合約的 Sun 公司授權者。

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

本文件以其「原狀」提供，對任何明示或暗示的條件、陳述或擔保，包括對適銷性、特殊用途的適用性或非侵權性的暗示保證，均不承擔任何責任，除非此免責聲明的適用範圍在法律上無效。

# 目錄

---

前言 .....	13
<b>1 產品概念 .....</b>	<b>19</b>
J2EE 平台簡介 .....	19
J2EE 應用程式 .....	19
容器 .....	20
J2EE 服務 .....	20
Web 服務 .....	20
用戶端存取 .....	20
外部系統與資源 .....	21
Application Server 元件 .....	22
伺服器實例 .....	22
管理網域 .....	22
叢集 .....	23
節點代理程式 .....	24
已命名的配置 .....	24
HTTP 負載平衡器外掛程式 .....	25
階段作業持續性 .....	25
叢集中的 IIOP 負載平衡 .....	26
Message Queue 與 JMS 資源 .....	27
高可用性資料庫 .....	27
簡介 .....	27
系統需求 .....	28
HADB 架構 .....	28
減少雙重故障的機會 .....	31
HADB 管理系統 .....	31
設定及配置資訊指南 .....	33
▼ 設定及配置高可用性的 Application Server .....	33

<b>2 規劃部署</b>	35
建立效能目標	35
估計流量	36
估計應用程式伺服器實例的負載	36
估計 HADB 的負載	39
規劃網路配置	41
估計頻寬需求	41
計算所需的頻寬	41
估計尖峰負載	42
配置子網路	42
選擇網路卡	43
HADB 的網路設定	43
規劃可用性	43
最佳化可用性	43
使用叢集改善可用性	44
增加系統的備援	44
設計決策	45
為尖峰負載或持續狀態負載設計	45
調整系統大小	46
規劃 Message Queue 代理程式部署	48
多個代理程式叢集	49
配置 Application Server 以使用 Message Queue 代理程式	50
部署方案範例	52
<b>3 選取拓樸</b>	53
共同需求	53
一般需求	53
HADB 節點與機器	54
負載平衡器配置	54
並置拓樸	55
配置範例	55
並置拓樸的變體	57
個別層級拓樸	59
配置範例	59
個別層級拓樸的變體	61

決定要使用的拓樸 .....	63
拓樸的比較 .....	63
<b>4 部署的檢核清單 .....</b>	<b>65</b>
部署的檢核清單 .....	65
<b>索引 .....</b>	<b>71</b>



# 圖清單

---

圖 3-1	並置拓樸範例 .....	56
圖 3-2	並置拓樸的變體 .....	58
圖 3-3	個別層級拓樸範例 .....	60
圖 3-4	個別層級拓樸的變體 .....	62





# 表清單

---

表 2-1	持續性頻率選項的比較 .....	39
表 2-2	持續性範圍選項的比較 .....	40
表 2-3	階段作業大小為 X MB 的 HADB 儲存空間需求 .....	48
表 3-1	拓樸的比較 .....	64
表 4-1	檢核清單 .....	65



# 範例清單

---

範例 2-1	回應時間的計算 .....	38
範例 2-2	每秒請求數的計算 .....	38
範例 2-3	所需頻寬的計算 .....	42
範例 2-4	尖峰負載的計算 .....	42



# 前言

「部署規劃指南」說明如何建立生產部署。

此前言說明整個 Sun Java™ 系統 Application Server 文件集的相關資訊與慣例。

## Application Server 文件集

Application Server 文件集說明部署規劃和系統安裝。Application Server 文件的單一資源定址器 (URL) 為 <http://docs.sun.com/coll/1343.4>。如需 Application Server 的簡介，請依下表所列順序參閱相關叢書。

表 P-1 Application Server 文件集中的書籍

書名	說明
「Documentation Center」	按作業與主旨分類組織的 Application Server 文件主題。
版本說明	軟體與文件的最新資訊。包含支援硬體、作業系統、Java 開發工具組 (JDK™) 與資料庫驅動程式的完整表格式摘要。
「Quick Start Guide」	如何開始使用 Application Server 產品。
Installation Guide	安裝軟體及其元件。
「Deployment Planning Guide」	評估系統需求和企業狀況，確保以最適合您的站點的方式部署 Application Server。此外還說明了部署伺服器時應該注意的常見問題及注意事項。
「Application Deployment Guide」	將應用程式與應用程式元件部署至 Application Server。其中包括有關部署描述元的資訊。
「Developer's Guide」	建立及實作計畫在 Application Server 上執行的 Java Platform, Enterprise Edition (Java EE 平台) 應用程式，這些應用程式採用 Java EE 元件與 API 的開放式 Java 標準模型。其中包括有關開發者工具、安全性、除錯及建立生命週期模組的資訊。
「Java EE 5 Tutorial」	使用 Java EE 5 平台技術與 API 來開發 Java EE 應用程式。
「Java WSIT Tutorial」	使用 Web 服務互通功能技術 (Web Service Interoperability Technologies, WSIT) 來開發 Web 應用程式。其中說明使用 WSIT 技術的方式、時機和原因，以及各種技術所支援的功能與選項。

表 P-1 Application Server 文件集中的書籍 (續)

書名	說明
「Administration Guide」	Application Server 的系統管理，包括配置、監視、安全性、資源管理及 Web 服務管理。
「High Availability Administration Guide」	高可用性資料庫安裝後的配置和管理說明。
「Administration Reference」	編輯 Application Server 配置檔案 domain.xml。
「Upgrade and Migration Guide」	從舊版 Application Server 升級或從競爭對手的應用程式伺服器遷移 Java EE 應用程式。此指南也說明鄰近產品發行版本與配置選項之間的差異，這些可能導致與產品規格不相容。
「Performance Tuning Guide」	調校 Application Server 以提昇效能。
「Troubleshooting Guide」	解決 Application Server 問題。
「Error Message Reference」	解決 Application Server 錯誤訊息。
「Reference Manual」	與 Application Server 一起提供的公程式指令；以線上手冊樣式編寫。其中包含 asadmin 指令行介面。

# 相關文件

Application Server 可單獨購買或隨 Sun Java Enterprise System (Java ES) 合購。這是一種軟體基礎架構，支援分散在網路或網際網路環境中的企業應用程式。若隨 Java ES 合購 Application Server，則應熟悉位於 <http://docs.sun.com/coll/1286.2> 上的系統文件。Java ES 及其元件的所有相關文件 URL 為 <http://docs.sun.com/prod/entsys.5>。

如需有關其他獨立 Sun Java System 伺服器產品的文件，請移至以下位址：

- Message Queue 文件 (<http://docs.sun.com/coll/1343.4>)
- Directory Server 文件 (<http://docs.sun.com/coll/1224.1>) 和 (<http://docs.sun.com/coll/1632.1>)
- Web Server 文件 (<http://docs.sun.com/coll/1308.3>) 和 (<http://docs.sun.com/coll/1425.2>)

Application Server 隨附之套裝軟體的 Javadoc 工具參考資料位於 <http://glassfish.dev.java.net/nonav/javaee5/api/index.html>。此外，以下資源可能會有用：

- Java EE 5 Specifications (<http://java.sun.com/javaee/5/javatech.html>)
- Java EE 5 Tutorial (<http://java.sun.com/javaee/5/docs/tutorial/doc/index.html>)
- Java EE Blueprints (<http://java.sun.com/reference/blueprints/index.html>)

# 預設路徑和檔案名稱

下表描述本書中使用的預設路徑和檔案名稱。

表 P-2 預設路徑和檔案名稱

預留位置	說明	預設值
<i>install-dir</i>	表示 Application Server 的基底安裝目錄。	Solaris™ 作業系統的 Java ES 安裝：  /opt/SUNWappserver/appserver  Linux 作業系統的 Java ES 安裝：  /opt/sun/appserver/  其他 Solaris 和 Linux 的安裝 (非超級使用者)：  user's-home-directory/SUNWappserver  其他 Solaris 和 Linux 的安裝 (root 使用者)：  /opt/SUNWappserver  所有 Windows 的安裝：  SystemDrive:\Sun\AppServer
<i>domain-root-dir</i>	表示包含所有網域的目錄。	Java ES Solaris 的安裝：  /var/opt/SUNWappserver/domains/  Java ES Linux 的安裝：  /var/opt/sun/appserver/domains/  所有其他的安裝：  install-dir/domains/
<i>domain-dir</i>	表示網域目錄。  在配置檔案中，您可能會看到 <i>domain-dir</i> 表示如下：  \${com.sun.aas.instanceRoot}	<i>domain-root-dir/domain-dir</i>
<i>instance-dir</i>	表示伺服器實例的目錄。	<i>domain-dir/instance-dir</i>

# 印刷排版慣例

下表描述本書在印刷排版上所做的變更。

表 P-3 印刷排版慣例

字體	意義	範例
AaBbCc123	指令、檔案及目錄的名稱；螢幕畫面輸出。	請編輯您的 <code>.login</code> 檔案。 請使用 <code>ls -a</code> 列出所有檔案。 <code>machine_name% you have mail.</code>
<b>AaBbCc123</b>	您所鍵入的資訊，與螢幕畫面輸出相對應	<code>machine_name% su</code> Password:
<i>AaBbCc123</i>	要使用實際名稱或值替代的預留位置	用於移除檔案的指令為 <code>rm filename</code> 。
<b><i>AaBbCc123</i></b>	書名、新術語和要強調的文字 (請注意，某些強調的項目線上顯示為粗體)	請閱讀「User's Guide」中的第 6 章。 <b>快取</b> 是本機儲存的副本。 請勿儲存該檔案。

# 符號慣例

下表說明本書中可能使用的符號。

表 P-4 符號慣例

符號	說明	範例	意義
[ ]	包含選擇性引數和指令選項。	<code>ls [-l]</code>	無需 <code>-l</code> 選項。
{   }	包含為所需指令選項提供的一組選擇。	<code>-d {y n}</code>	<code>-d</code> 選項要求您使用 <code>y</code> 引數或 <code>n</code> 引數。
\${ }	表示變數參照。	<code>\${com.sun.javaRoot}</code>	參照 <code>com.sun.javaRoot</code> 變數的值。
-	連接需同時按下的多個按鍵。	Ctrl-A	同時按 Ctrl 鍵和 A 鍵。
+	連接需連續按下的多個按鍵。	Ctrl+A+N	按 Ctrl 按鍵，然後鬆開，再依次按後面的按鍵。



表 P-4 符號慣例 (續)

符號	說明	範例	意義
→	表示圖形使用者介面中的功能表項目選取。	[檔案] → [新建] → [範本]	從 [檔案] 功能表中，選擇 [新建]。從 [新建] 子功能表中，選擇 [範本]。

## 文件、支援和訓練

Sun 網站提供以下其他資源的相關資訊：

- 文件 (<http://www.sun.com/documentation/>)
- 支援 (<http://www.sun.com/support/>)
- 訓練 (<http://www.sun.com/training/>)

## 搜尋 Sun 產品文件

除了從 docs.sun.com<sup>SM</sup> 網站搜尋 Sun 產品文件之外，還可以在搜尋欄位中鍵入下列語法，使用搜尋引擎：

```
search-term site:docs.sun.com
```

例如，若要搜尋「broker」，請輸入下列語法：

```
broker site:docs.sun.com
```

若要在搜尋中包含其他 Sun 網站 (例如 [java.sun.com](http://java.sun.com)、[www.sun.com](http://www.sun.com) 與 [developers.sun.com](http://developers.sun.com))，請在搜尋欄位中使用 sun.com 取代 docs.sun.com。

## 協力廠商網站參照

本文件中提供了協力廠商 URL 以供參考，另亦提供其他相關的資訊。

**備註** – Sun 對本文件中提到的協力廠商網站的可用性不承擔任何責任。對於此類網站或資源中的 (或透過它們所取得的) 任何內容、廣告、產品或其他材料，Sun 並不表示認可，也不承擔任何責任。對於因使用或依靠此類網站或資源中的 (或透過它們所取得的) 任何內容、產品或服務而造成的、名義上造成的或連帶產生的任何實際或名義上之損壞或損失，Sun 概不負責，也不承擔任何責任。

## Sun 歡迎您提出寶貴意見

Sun 致力於提高文件品質，因此誠心歡迎您提出意見與建議。若要分享您的意見，請至 <http://docs.sun.com>，並按一下 [Send Comments (傳送您的意見)]。在線上表單中，提供完整的文件標題與文件號碼。文件號碼是一個七位或九位的數字，可以在書的標題頁面或文件的 URL 中找到。例如，本書的文件號碼為 820-4904。

在您提出意見時，可能需要在表單中輸入英文版書名和文件號碼，本書的英文版文件號碼和書名為：819-3680 和「Sun Java System Application Server 9.1 Deployment Planning Guide」。

# 產品概念

---

Sun Java System Application Server 提供強大的平台，可供開發、部署及管理 J2EE 應用程式之用。主要功能包括具延伸性的作業事件管理、容器管理式的持續性執行階段、Web 服務效能、叢集、高可用性階段作業狀態、安全性以及整合等功能。

本小節包含以下主題：

- 第 19 頁的「J2EE 平台簡介」
- 第 22 頁的「Application Server 元件」
- 第 27 頁的「高可用性資料庫」
- 第 33 頁的「設定及配置資訊指南」

## J2EE 平台簡介

Application Server 實作 Java 2 Enterprise Edition (J2EE) 1.4 技術。J2EE 平台是一組標準規格，旨在說明應用程式伺服器的應用程式元件、API，以及執行階段容器與服務。

## J2EE 應用程式

J2EE 應用程式由 JavaServer Pages (JSP)、Java servlet 與 Enterprise JavaBeans (EJB) 模組等元件所組成。這些元件可供軟體開發人員建立大型之分散式應用程式。開發人員可將 J2EE 應用程式封裝為 Java 歸檔 (JAR) 檔案 (類似壓縮檔案)，然後發行至各生產站點。管理員透過將 J2EE JAR 檔案部署在一或多個伺服器實例 (或實例叢集)，可將 J2EE 應用程式安裝在 Application Server 上。

下圖說明將於以下章節討論的 J2EE 平台元件。

抱歉：目前尚未提供此圖。

## 容器

每個伺服器實例包含兩個容器：Web 和 EJB。容器是指提供 J2EE 元件安全性與作業事件管理等服務的執行階段環境。Web 元件 (例如 JSP 和 servlet) 執行於 Web 容器內。Enterprise JavaBeans 執行於 EJB 容器內。

## J2EE 服務

J2EE 平台提供應用程式服務，包括：

- **命名** - 命名與目錄服務可將物件連結至名稱。J2EE 應用程式可查找物件的 Java Naming and Directory Interface (JNDI) 名稱，以尋找物件。
- **安全性** - Java Authorization Contract for Containers (JACC) 是為 J2EE 容器所定義的一組安全性合約。依照用戶端的身分，容器可限制其對容器資源與服務的存取。
- **作業事件管理** - 作業事件是不可分割的工作單元。例如，在銀行帳戶之間轉帳是一個作業事件。作業事件管理服務用於確定作業事件是已完成或是已回復。
- **訊息服務** - 在個別系統上代管的應用程式可透過使用 Java™ Message Service (JMS) 交換訊息，以彼此通訊。JMS 是 J2EE 平台不可或缺的一部分，用以簡化整合異質性企業應用程式的作業。

## Web 服務

用戶端除了可透過 HTTP、RMI/IIOP 與 JMS 存取 J2EE 1.4 應用程式外，還可以遠端 Web 服務的方式加以存取。Web 服務的實作是使用 Java API for XML-based RPC (JAX-RPC)。J2EE 應用程式也可當作 Web 服務的用戶端，這在網路應用裡極為常見。

Web 服務描述語言 (WSDL) 是說明 Web 服務介面的 XML 格式。Web 服務用戶可動態剖析 WSDL 文件，以判斷 Web 服務所提供的作業以及執行方式。Application Server 使用其他應用程式可透過 Java API for XML Registries (JAXR) 存取的登錄，以發行 Web 服務介面說明。

## 用戶端存取

用戶端存取 J2EE 應用程式的方式有幾種。瀏覽器用戶端使用超文字傳輸協定 (HTTP) 存取 Web 應用程式。如需安全通訊，瀏覽器會採用使用安全通訊端層 (SSL) 的 HTTP 安全 (HTTPS) 通訊協定。

在應用程式用戶端容器中執行的豐富型用戶端應用程式，可使用 Object Request Broker (ORB)、遠端方法呼叫 (RMI) 與網際網路 ORB 交換協定 (IIOP)，或 IIOP/SSL (安全 IIOP)，直接查找及存取 Enterprise JavaBeans。它們可使用 HTTP/HTTPS、JMS 與 JAX-RPC 存取應用程式與 Web 服務，還可使用 JMS 將訊息傳送至應用程式與訊息驅動 Bean，並從之接收訊息。

符合 Web 服務互通性 (WS-I) 基本設定檔的用戶端，可存取 J2EE Web 服務。WS-I 是 J2EE 標準不可或缺的一部分，用於定義可互通的 Web 服務。它可讓使用任何支援語言編寫的用戶端，存取部署到 Application Server 上的 Web 服務。

最佳的存取機制視特定應用程式與預期流量而定。Application Server 支援可個別配置的 HTTP、HTTPS、JMS、IIOP 與 IIOP/SSL 偵聽程式。您可以為每個通訊協定設定多個偵聽程式，以增加延展性與穩定性。

J2EE 應用程式也可視作 J2EE 元件的用戶端 (例如部署在其他伺服器上的 Enterprise JavaBeans 模組)，且可使用任何一種存取機制。

## 外部系統與資源

在 J2EE 平台上，外部系統稱為**資源**。例如，資料庫管理系統是 JDBC 資源。每個資源是獨一無二的，並由 Java Naming and Directory Interface (JNDI) 名稱識別。應用程式透過下列 API 與元件，存取外部系統：

- *Java Database Connectivity (JDBC)* - 資料庫管理系統 (DBMS) 提供儲存、組織及擷取資料的能力。大多數的企業應用程式將資料儲存在關聯式資料庫中，而應用程式則透過 JDBC 存取關聯式資料庫。Application Server 包含 PointBase DBMS，可用於應用程式範例以及應用程式開發與原型設計，但是不適用於部署。Application Server 提供認證的 JDBC 驅動程式，用於連線至主要關聯式資料庫。這些驅動程式適用於部署。
- *Java Message Service* - 訊息傳送是軟體元件或應用程式之間的通訊方法。訊息傳送用戶端透過實作 Java Messaging Service (JMS) API 的訊息傳送提供者，可將訊息傳送至任何其他用戶端，及由其處接收訊息。Application Server 包含高效能 JMS 代理程式 Sun Java System Message Queue。Application Server Platform Edition 包含免費的 Message Queue Platform Edition。Application Server Enterprise Edition 包含支援叢集與容錯移轉的 Message Queue Enterprise Edition。
- **J2EE 連接器** - J2EE 連接器架構可整合 J2EE 應用程式與現有的企業資訊系統 (EIS)。應用程式透過稱為**連接器**或**資源配接卡**的可移植性 J2EE 元件存取 EIS，類似於使用 JDBC 驅動程式存取 RDBMS。資源配接卡的發行形式為獨立資源配接卡歸檔 (RAR) 模組，或包含在 J2EE 應用程式歸檔中。由於是 RAR，所以部署方式類似其他 J2EE 元件。Application Server 包含與常用 EIS 整合的評估資源配接卡。
- **JavaMail** - 透過 JavaMail API，應用程式可連線至簡易郵件傳輸協定 (SMTP) 伺服器，以傳送與接收電子郵件。

# Application Server 元件

本節說明 Sun Java System Application Server 中的元件：

- 第 22 頁的「伺服器實例」
- 第 22 頁的「管理網域」
- 第 23 頁的「叢集」
- 第 24 頁的「節點代理程式」
- 第 24 頁的「已命名的配置」
- 第 25 頁的「HTTP 負載平衡器外掛程式」
- 第 26 頁的「叢集中的 IIOP 負載平衡」
- 第 27 頁的「Message Queue 與 JMS 資源」

下圖描述這些 Application Server 元件如何使用提供高可用性的簡易拓模範例進行互動。在此拓模範例中，一位管理員管理兩部組織為叢集的機器。HADB 與 Application Server 程序位於相同的機器上。Domain Administration Server 可能在另一台機器上由本身代管，或在任何一部代管應用程式伺服器實例的機器上。圖中的線條表示通訊或控制。

瀏覽器型的管理主控台等管理工具會與 Domain Administration Server (DAS) 通訊，再由其與節點代理程式和伺服器實例通訊。

## 伺服器實例

伺服器實例是在單一 Java 虛擬機器 (JVM) 程序中執行的 Application Server。Application Server 經 Java 2 Standard Edition (J2SE) 5.0 與 1.4 認證。建議的 J2SE 發行軟體隨附於 Application Server 安裝中。

由於 Application Server 與伴隨的 JVM 皆設計為可擴充為多顆處理器，因此在一部機器上建立一個伺服器實例通常已足夠。但是，在一部機器上建立多個實例有助於應用程式隔離與輪替升級。在某些情況下，具有多項實例的大型伺服器可用在管理網域以外的地方。管理工具可簡化在多部機器之間建立、刪除及管理伺服器實例的作業。

## 管理網域

**管理網域** (或簡稱**網域**) 是一組共同管理的伺服器實例。伺服器實例屬於單一管理網域。網域中的實例可於不同實體主機上執行。

您可以從一個 Application Server 安裝建立多個網域。不同的組織和管理員可將所有伺服器實例分組為個別的網域，以共用單一的 Application Server 安裝。每個網域都有自己的獨立於其他網域的配置、記錄檔和應用程式部署區域。變更一個網域的配置並不會影響其他網域的配置。同理，在一個網域上部署應用程式，並不會將其部署至其他網域或使其顯示於其他網域中。管理員無論何時都只能向一個網域認證，因而僅能執行該網域的管理作業。

## Domain Administration Server (DAS)

網域具有一個 Domain Administration Server (DAS)，此為特別指定用於代管管理應用程式的應用程式伺服器實例。DAS 將認證管理員、接受來自管理工具的請求，並與網域中的伺服器實例進行通訊以執行請求。

管理工具包含 `asadmin` 指令行工具與瀏覽器型的管理主控台。Application Server 也提供 JMX 型的 API 以進行伺服器管理。管理員一次可檢視並管理一個網域，從而強制執行安全隔離。

DAS 有時也稱為**管理伺服器**或**預設伺服器**。稱為預設伺服器是因為它是某些管理作業的預設目標。

由於 DAS 是應用程式伺服器實例，因此也可因測試目的而代管 J2EE 應用程式。但是，請勿用於代管生產應用程式。例如，若尚未建立將代管生產應用程式的叢集與實例，最好將應用程式部署至 DAS。

DAS 會保留一個儲存庫，內含網域與所有已部署的應用程式配置。若 DAS 停用或關機，不會影響使用中的伺服器實例之效能或可用性，但會無法進行管理變更。在某些情況下，基於安全的考量，刻意停止 DAS 程序相當有助益；例如可用於固定生產環境的配置。

管理指令的功能是備份與復原網域配置與應用程式。透過標準備份與復原程序，您可以快速復原有效配置。若 DAS 主機故障，您必須建立新的 DAS 安裝以復原先前的網域配置。如需相關指示，請參閱「Sun Java System Application Server 9.1 管理指南」中的「重新建立網域管理伺服器」。

Sun Cluster Data Services 透過容錯移轉 DAS 主機 IP 位址及使用全域檔案系統，提供高可用性的 DAS。此解決方案在面對許多失敗類型時，幾乎可不間斷地使用 DAS 與儲存庫。Sun Cluster Data Services 隨附於 Sun Java Enterprise System 中，或隨 Sun Cluster 另外購買。如需更多資訊，請參閱 Sun Cluster Data Services 的文件。

## 叢集

叢集是共用相同的應用程式、資源與配置資訊之已命名的伺服器實例集合。可以將不同機器上的伺服器實例分組為一個邏輯叢集，然後將其做為一個單位來管理。可以使用 DAS 輕鬆控制多機器叢集的生命週期。

叢集可啓用水平可延伸性、負載平衡和容錯移轉保護。根據定義，叢集中的所有實例均具有相同的資源和應用程式配置。叢集中的伺服器實例或機器故障時，負載平衡器會偵測到該故障，並將通訊從出現故障的實例重新導向至叢集中的其他實例，然後回復使用者階段作業狀態。因為相同應用程式和資源均位於叢集中的所有實例上，因此實例可以容錯移轉至叢集中的任何其他實例。

叢集、網域和實例的相互關係如下：

- 管理網域可有零或多個叢集。



- 叢集有一或多個伺服器實例。
- 叢集屬於單一網域。

## 節點代理程式

節點代理程式是在每部代管伺服器實例的機器上執行的簡易程序，包括代管 DAS 的機器。節點代理程式：

- 聽從 DAS 指示啟動與停止伺服器實例。
- 重新啟動有故障的伺服器實例。
- 提供故障伺服器的記錄檔檢視，並協助進行遠端診斷。
- 可在啟動由其監視的伺服器實例時，將每個伺服器實例的本機配置儲存庫與 DAS 的中央儲存庫同步化。
- 最初建立實例時會建立實例所需的目錄，並將實例的配置與中央儲存庫同步化。
- 在刪除伺服器實例時，執行適當的清除作業。

每個實體主機必須為其所屬的每個網域設定至少一個節點代理程式。若實體主機有來自多個網域的實例，則該實例需要在每個網域各有一個節點代理程式。雖然可以在主機上為每個網域設定多個節點代理程式，但這樣做並沒有好處。

因為伺服器實例由節點代理程式啟動與停止，所以節點代理程式必須一直執行。因此，它會隨著作業系統啟動時啟動。在 Solaris 與其他 Unix 平台上，節點代理程式可由 `inetd` 程序啟動。在 Windows 上，節點代理程式可設定為 Windows 服務。

如需有關節點代理程式的更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 8 章「配置節點代理程式」。

## 已命名的配置

**已命名的配置**是封裝 Application Server 特性設定的抽象概念。叢集與獨立伺服器實例會參照已命名的配置取得其特性設定。透過已命名的配置，J2EE 容器的配置與其所在的實體機器無關，僅與 IP 位址、連接埠號碼與堆疊記憶體容量等特定項目相關。使用已命名的配置可使 Application Server 的管理更強大且更具彈性。

若要套用配置變更，只要變更已命名配置的特性設定，所有參照此配置的叢集與獨立實例便會獲得變更。當所有參照皆移除後，才能刪除已命名的配置。一個網域可包含多個已命名的配置。

Application Server 隨附預設配置，稱為 `default-config`。預設配置已針對 Application Server Platform Edition 中的開發人員生產力，以及 Application Server Enterprise Edition 中的安全性與高可用性進行最佳化。

您可以根據可因自己的用途進行自訂之預設配置，建立自己命名的配置。使用管理主控台與 `asadmin` 指令行公用程式，建立及管理已命名的配置。



## HTTP 負載平衡器外掛程式

負載平衡器會將工作負荷量分散至多部實體機器，因而增加系統的整體流量。

Application Server Enterprise Edition 包含 Sun Java System Web Server、Apache Web Server 與 Microsoft Internet Information Server 的負載平衡器外掛程式。

負載平衡器外掛程式會接受 HTTP 與 HTTPS 請求，然後將這些請求轉寄至叢集中的其中一個應用程式伺服器實例。若實例出現故障而無法使用 (因為網路故障) 或無回應，請求會重新導向至現有的可用機器。負載平衡器還可以識別出現故障的實例是否回復，並相應地重新分配負載。

對於簡單且無狀態的應用程式，負載平衡的叢集即已足夠。但是，對於具有階段作業狀態的任務關鍵性應用程式，請將負載平衡叢集與 HADB 配合使用。

若要設定具有負載平衡的系統，除了 Application Server 之外，還必須安裝 Web 伺服器及負載平衡器外掛程式。然後您必須：

- 建立要參與負載平衡的應用程式伺服器叢集。
- 將應用程式部署到這些負載平衡的叢集。

參與負載平衡的伺服器實例和叢集具有同質環境。這通常表示伺服器實例參照相同的伺服器配置，可以存取相同的實體資源，並部署相同的應用程式。同質環境確保故障前後，負載平衡器均始終在叢集使用中的實例間平均分配負載。

使用 `asadmin` 指令行工具建立負載平衡器配置、將叢集與伺服器實例的參照增加至配置、啟用負載平衡器所參照的叢集、啟用應用程式進行負載平衡、選擇性建立運作狀態檢查程式、產生負載平衡器配置檔案，最後將負載平衡器配置檔案複製到 Web 伺服器的 `config` 目錄。管理員可建立程序檔自動化此整個程序。

如需更詳細的資訊與完整的配置指示，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 5 章「配置 HTTP 負載平衡」。

## 階段作業持續性

J2EE 應用程式通常具有大量階段作業狀態資料。Web 購物車即為階段作業狀態的經典範例。而且，應用程式可以快取頻繁需要的階段作業物件資料。實際上，幾乎所有需要進行大量使用者互動活動的應用程式均需要維護階段作業狀態。HTTP 階段作業與有狀態的階段作業 Bean (SFSB) 皆具有階段作業狀態資料。

雖然階段作業狀態不如儲存在資料庫中的作業事件狀態重要，但是在伺服器故障期間能保留階段作業狀態，對一般使用者而言很重要。Application Server 提供在儲存庫中儲存 (或保留) 此階段作業狀態的功能。若代管使用者階段作業的應用程式伺服器實例發生故障，可回復階段作業狀態。階段作業可繼續而不會失去資訊。

Application Server 支援下列階段作業持續性存放區類型：

- 記憶體

- 高可用性 (HA)
- 檔案

若是記憶體持續性，狀態一律會保留在記憶體中，在失敗後不會存留。若是 HA 持續性，Application Server 會使用 HADB 作為 HTTP 與 SFSB 階段作業的持續性存放區。若是檔案持續性，Application Server 會序列化階段作業物件，並將其儲存在階段作業管理員特性所指定的檔案系統位置。若是 SFSB 且未指定 HA，Application Server 會將狀態資訊儲存在此位置的階段作業存放區子目錄中。

檢查 SFSB 的狀態是否有變更且需要儲存的動作，稱為**檢查點檢查**。啓用時，檢查點檢查一般會在完成 SFSB 的所有作業事件之後發生，即使作業事件回復亦然。如需有關開發有狀態的階段作業 Bean 之更多資訊，請參閱「Sun Java System Application Server 9.1 Developer's Guide」中的「Using Session Beans」。如需有關啓用 SFSB 容錯移轉的更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的「有狀態階段作業 Bean 容錯移轉」。

階段作業持續性配置設定除了影響 Application Server 處理的請求數目外，還會影響 HADB 每分鐘接收的請求數目，以及每個請求中的階段作業資訊。

如需有關配置階段作業持續性的更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 9 章「配置高可用性階段作業持續性和容錯移轉」。

## 叢集中的 IIOP 負載平衡

透過 IIOP 負載平衡，可將 IIOP 用戶端請求分散至不同的伺服器實例或名稱伺服器中。其目的為在整個叢集上平均分散負載，從而提供可延伸性。IIOP 負載平衡在 Sun Java System Application 中結合 EJB 叢集與可用性功能，不僅提供負載平衡，還提供 EJB 容錯移轉。

當用戶端執行 JNDI 物件查找時，命名服務會建立與特定伺服器實例相關聯的 InitialContext (IC) 物件。從那時起，使用該 IC 物件進行的所有查找請求均會傳送至相同的伺服器實例。使用該 InitialContext 查找的所有 EJBHome 物件均會在相同的目標伺服器上託管。之後取得的所有 Bean 參照也會建立在相同的目標主機上。由於建立 InitialContext 物件時，所有用戶端會隨機建立即時目標伺服器的清單，因此，此作業可有效地提供負載平衡。如果目標伺服器實例當機，則查找或 EJB 方法呼叫將容錯移轉至其他伺服器實例。

例如，如此圖所述，ic1、ic2 與 ic3 是使用 Client2 的程式碼所建立的三個不同的 InitialContext 實例。這些實例會發行至叢集中的三個伺服器實例。接著，此用戶端所建立的 Enterprise JavaBeans 會分配至此三個實例。Client1 僅建立一個 InitialContext 物件，且來自此用戶端的 Bean 參照僅會在伺服器實例 1 上。若伺服器實例 2 當機，ic2 上的查找請求會容錯移轉至其他伺服器實例（不一定是伺服器實例 3）。任何對伺服器實例 2 上先前代管之 Bean 的 Bean 方法呼叫，也會在安全無虞的前提下自動重新導向至其他實例。當查找容錯移轉自動化時，Enterprise JavaBeans 模組僅會在安全無虞的前提下重試方法呼叫。

IIOP 負載平衡與容錯移轉不需設定即可執行。在部署應用程式期間，無需特殊的步驟。對叢集增加新實例或從叢集中刪除實例，將不會更新該叢集的現有用戶端檢視。您必須在用戶端手動更新端點清單。

## Message Queue 與 JMS 資源

Sun Java System Message Queue (MQ) 提供可靠且非同步的分散式應用程式訊息傳送。MQ 是實作 Java Message Service (JMS) 標準的企業訊息傳送系統。MQ 提供 J2EE 應用程式元件的訊息傳送，例如訊息驅動 Bean (MDB)。

Application Server 將 Sun Java System Message Queue 整合至 Application Server，以實作 Java Message Service (JMS) API。Application Server Enterprise Edition 包含 MQ 的企業版，該版具有容錯移轉、叢集與負載平衡等功能。

對於基本 JMS 管理作業，請使用 Application Server 管理主控台與 `asadmin` 指令行公用程式。

對於進階作業 (包含管理 Message Queue 叢集)，請使用 `install_dir/imq/bin` 目錄中所提供的工具。如需有關管理 Message Queue 的詳細資訊，請參閱「[Sun Java System Message Queue Administration Guide](#)」。

如需有關部署 JMS 應用程式與 MQ 叢集以進行訊息容錯移轉的資訊，請參閱第 48 頁的「[規劃 Message Queue 代理程式部署](#)」。

## 高可用性資料庫

本節論述以下主題：

- 第 27 頁的「簡介」
- 第 28 頁的「系統需求」
- 第 28 頁的「HADB 架構」
- 第 31 頁的「減少雙重故障的機會」
- 第 31 頁的「HADB 管理系統」

### 簡介

J2EE 應用程式對階段作業持續性的需求先前於第 25 頁的「[階段作業持續性](#)」中已有說明。Application Server 使用高可用性資料庫 (HADB) 作為高可用性階段作業存放區。HADB 隨附於 Application Server Enterprise Edition 中，但可在不同的主機上執行部署。HADB 為 HTTP 階段作業與有狀態的階段作業 Bean 資料提供高可用性資料存放區。

此分離性架構的優點包含：

- 高可用性叢集中的伺服器實例鬆散結合，且可作為高效能的 J2EE 容器。

- 停止與啟動伺服器實例不會影響其他伺服器或其可用性。
- HADB 可在另外一組較便宜的機器上執行 (例如，具有單一處理器或雙處理器的機器)。數個叢集可共用這些機器。您可以在執行 Application Server 的機器 (並置) 上或不同的機器 (個別層級) 上執行 HADB，視部署需求而定。如需有關這兩個選項的更多資訊，請參閱第 55 頁的「並置拓樸」。
- 當狀態管理需求變更時，可以將資源增加至 HADB 系統，而不會影響現有的叢集或其應用程式。

---

**備註** – 我們已針對 Application Server 用途最佳化 HADB，HADB 不適合讓應用程式當作一般用途的資料庫。

---

如需 HADB 硬體與網路系統需求的資訊，請參閱「Sun Java System Application Server 9.1 版本說明」中的「硬體和軟體需求」。如需 HADB 所需的其他系統配置步驟，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 2 章「安裝和設定高可用性資料庫」。

## 系統需求

HADB 主機的系統需求包含：

- 每個 HADB 節點至少有一個 CPU。
- 每個節點至少有 512 MB 的記憶體。

如需網路配置需求的資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 2 章「安裝和設定高可用性資料庫」。如需非常高可用性的其他需求，請參閱第 31 頁的「減少雙重故障的機會」。

## HADB 架構

HADB 是包含成對節點的分散式系統。節點會分成兩個資料備援單元 (DRU)，每個節點來自每個 DRU 的每對節點，如第 29 頁的「資料備援單元」中所述。

每個節點包含：

- 作業事件狀態複製的一組程序
- 在程序間通訊時所使用的專屬共用記憶體區域。
- 一或多個輔助儲存裝置 (磁碟)。

可代管一或多個階段作業資料庫的一組 HADB 節點。每個階段作業資料庫與不同的應用程式伺服器叢集相關聯。刪除叢集也會刪除相關聯的階段作業資料庫。

如需 HADB 硬體需求的資訊，請參閱「Sun Java System Application Server 9.1 版本說明」中的「硬體和軟體需求」。

## 節點與節點程序

HADB 節點有兩種類型：

- 儲存資料的**使用中節點**。
- 一開始不包含任何資料的**備用節點**，但在使用中節點無法使用時，會當作使用中節點執行。備用節點為選用節點，但對達成高可用性很有用。

每個節點皆有一個父程序和數個子程序。父程序又稱為節點監督員 (NSUP)，由管理代理程式啟動。父程序負責建立子程序並將其保持在執行狀態。

子程序包含：

- 作業事件伺服器程序 (TRANS) 可協調分散式節點上的作業事件，並管理資料儲存。
- 關聯代數伺服器程序 (RELALG) 可協調並執行複合關聯代數查詢，例如排序和連結。
- SQL 共用記憶體伺服器程序 (SQLSHM) 可維護 SQL 字典快取。
- SQL 伺服器程序 (SQLC) 可接收用戶端查詢、將查詢編譯為本機 HADB 指示、將指示傳送至 TRANS、接收結果，然後將結果傳遞至用戶端。每個節點針對每個用戶端連線會有一個主要的 SQL 伺服器和一個輔助伺服器。
- 節點管理員伺服器程序 (NOMAN)，讓管理代理程式用來執行 hadbm 管理用戶端所發出的管理指令。

## 資料備援單元

如前所述，每個 HADB 實例包含一對 DRU。每個 DRU 與配對中另一個 DRU 有相同數目的使用中節點及備用節點。DRU 中的每個使用中節點在另一個 DRU 中有 **鏡像節點**。由於鏡像的原因，每個 DRU 會包含完整的資料庫複本。

下圖顯示內含六個節點的 HADB 架構範例：四個使用中節點與兩個備用節點。節點 0 與 1、2 與 3 是鏡像組。在此範例中，每個主機有一個節點。一般來說，若主機有足夠的系統資源，便可有多個節點 (請參閱第 28 頁的「系統需求」)。

---

**備註** – 您必須增加代管成對 HADB 節點的機器，每個 DRU 中有一部機器。

---

HADB 透過複製資料與服務可達成高可用性。鏡像節點上的資料複本會指定為**主要複本**與**緊急待命複本**。主要複本會執行插入、刪除、更新與讀取等作業。緊急待命複本會接收主要複本作業的記錄，並在作業事件存留時間內加以重做。僅有主要節點可執行讀取作業，因此不會記錄。每個節點皆包含主要複本與緊急待命複本，並扮演這兩個角色。資料庫會分段並分散至 DRU 中的使用中節點。鏡像組中的各節點包含相同的一組資料片段。複製鏡像節點中的資料稱為**複製**。複製可讓 HADB 提供高可用性：當節點故障時，鏡像節點幾乎會立即接管 (幾秒內)。複製可確保可用性並使人無法察覺節點故障或 DRU 故障，而不致失去資料或無法提供服務。



當鏡像節點接管故障節點的功能時，必須執行雙份工作：也就是自己的工作以及故障節點的工作。若鏡像節點的資源不足，超載情況會降低節點的效能並增加節點故障的可能性。當節點故障時，HADB 會嘗試重新啟動節點。若故障的節點無法重新啟動 (例如由於硬體故障)，系統會繼續運作但可用性會降低。

HADB 可容許一個節點、整個 DRU 或多個節點故障，但無法容許節點與鏡像節點同時故障的「雙重故障」。如需有關如何降低雙重故障可能性的資訊，請參閱第 31 頁的「減少雙重故障的機會」。

## 備用節點

當節點故障時，其鏡像節點會接管故障的節點。若故障的節點沒有備用節點，則此時故障的節點將無鏡像節點。備用節點會自動取代故障節點的鏡像節點。擁有備用節點可降低系統沒有鏡像節點的運作時間。

備用節點一般不含資料，但會不斷監視 DRU 中使用中節點是否故障。當節點故障且在指定逾時期間未回復時，備用節點會從鏡像節點複製資料並與之同步化。所需的時間視複製的資料量及系統與網路能力而定。進行同步化之後，備用節點會無須手動介入而自動取代鏡像節點，從而解除鏡像節點的超載情況，而平衡鏡像節點的負載。這又稱為故障回復或自我修復。

修復故障的主機 (透過更換硬體或升級軟體) 並重新啟動時，由於原始備用節點現在變成使用中，因此在此主機上執行的一或多個節點，會加入系統成為備用節點。

備用節點不是必要項目，但這些節點可讓系統即使在機器故障時，也可維持其服務整體水準。備用節點亦可簡化在代管使用中節點的機器上，執行規劃維護的作業。為每個 DRU 配置一部機器作為備用機器，如此一來，若其中一部機器故障，HADB 系統便可繼續，而不會對效能與可用性造成不良影響。

---

**備註** – 一般來說，請以具有足夠應用程式伺服器實例與 HADB 節點的備用機器，取代無法使用的任何機器。

---

## 備用節點配置範例

下列範例說明如何在 HADB 部署中使用備用節點。可能的部署拓樸有兩種：**並置**，在此拓樸中，HADB 與 Application Server 位於相同的主機上；以及**個別層級**，在此拓樸中，HADB 與 Application Server 位於不同的主機上。如需有關部署拓樸的更多資訊，請參閱第 3 章。

### 範例：並置配置

在此備用節點配置範例中，假設您有一個內含四部 Sun Fire™ V480 伺服器的並置拓樸，其中每部伺服器有一個應用程式伺服器實例與兩個 HADB 資料節點。

請為備用節點額外配置兩部伺服器 (每個 DRU 一部機器)。每個備用機器執行一個應用程式伺服器實例以及兩個備用 HADB 節點。

## 範例：個別層級配置

假設您有一個個別層級的拓樸，其中 HADB 層有兩部 Sun Fire™ 280R 伺服器，每部伺服器執行兩個 HADB 資料節點。若要在即使一部機器無法使用的情況下，維持此系統的完整能力，請為應用程式伺服器實例層與 HADB 層各配置一部備用機器。

應用程式伺服器實例層的備用機器，必須和應用程式伺服器實例層中其他機器有相同數目的實例。同理，HADB 層的備用機器，必須和 HADB 層中其他機器有相同數目的 HADB 節點。

## 減少雙重故障的機會

HADB 的內建資料複製可容許單一節點或整個 DRU 故障。依預設，當鏡像節點組或兩個 DRU 皆故障時，HADB 無法承受此**雙重故障**。在這類情況下，將無法使用 HADB。

除了如上節所述使用備用節點外，還可以採取下列步驟降低雙重故障的可能性：

- **提供獨立的電源供應**：為達最佳的容錯能力，支援一個 DRU 的伺服器必須有獨立的電源(透過不斷電系統)、處理單元及儲存裝置。若在一個 DRU 中發生電源故障，其他 DRU 中的節點會繼續處理請求，直到電源恢復為止。
- **提供雙重互連**：若要容許單一網路的故障，請複製 DRU 之間的線路與交換器。

這些步驟為選用步驟，但會增加 HADB 實例的整體可用性。

## HADB 管理系統

HADB 管理系統提供內建安全性並協助多平台的管理。如下圖所述，HADB 管理架構包含下列元件：

- 第 32 頁的「管理用戶端」
- 第 32 頁的「管理代理程式」
- 第 32 頁的「管理網域」
- 第 33 頁的「儲存庫」

如圖所示，執行 HADB 服務的每部機器上會執行一個 HADB 管理代理程式。每部機器一般會代管一或多個 HADB 節點。HADB 管理網域類似 Application Server 網域，包含許多機器。一個網域中至少需有兩部機器，資料庫才有容錯能力，且一般來說，機器的數目必須是偶數才可組成 DRU 組。因此，一個網域可包含許多管理代理程式。

如圖所示，一個網域可包含一或多個資料庫實例。一部機器可包含一或多個屬於一或多個資料庫實例的節點。

## 管理用戶端

HADB 管理用戶端是指令行公用程式 `hadbm`，用於管理 HADB 網域及其資料庫實例。HADB 服務可持續執行 (即使相關聯的應用程式伺服器叢集停止亦然)，但如果要刪除，則必須小心關閉。如需有關使用 `hadbm` 的更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 3 章「管理高可用性資料庫」。

您可以使用 `asadmin` 指令行公用程式建立及刪除與高可用性叢集相關聯的 HADB 實例。如需更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 9 章「配置高可用性階段作業持續性和容錯移轉」。

## 管理代理程式

管理代理程式是伺服器程序 (名為 `ma`)，可存取主機上的資源；例如，它可建立裝置並啟動資料庫程序。管理代理程式會協調及執行管理用戶端指令，例如啟動或停止資料庫實例。

管理用戶端透過指定代理程式的位址與連接埠號碼，連線至管理代理程式。連線之後，管理用戶端就會透過管理代理程式將指令傳送至 HADB。代理程式接收到請求後會加以執行。因此，管理代理程式必須執行於主機上，才可對該主機發出任何 `hadbm` 管理指令。您可將管理代理程式配置為自動啟動的系統服務。

## 確保管理代理程式的可用性

管理代理程式程序可在 HADB 節點監督員程序失敗時將其重新啟動，從而確保 HADB 節點監督員程序的可用性。因此，對於部署，您必須確保 `ma` 程序的可用性，才可維持 HADB 的整體可用性。重新啟動後，管理代理程式會從系統網域的其他代理程式，回復網域與資料庫配置資料。

使用主機作業系統 (OS) 以確保管理代理程式的可用性。在 Solaris 或 Linux 上，`init.d` 可確保 `ma` 程序在程序失敗並重新啟動作業系統之後的可用性。在 Windows 上，管理代理程式會以 Windows 服務執行。因此，若代理程式故障或作業系統重新啟動，作業系統便會重新啟動管理代理程式。

## 管理網域

HADB 管理網域是一組主機，每個主機有在相同連接埠號碼上執行的管理代理程式。網域中的主機可包含一或多個 HADB 資料庫實例。定義管理網域的方法，是依據代理程式所使用的共用連接埠號碼，以及建立網域或增加代理程式至網域時所產生的識別碼 (稱為 `domainkey`)。`domainkey` 提供唯一的網域識別碼，這點極為重要，因為管理代理程式使用多重播送進行通訊。您可以設定 HADB 管理網域對應 Application Server 網域。

在一個網域中有多個資料庫實例有助於開發環境，因為如此可讓不同的開發人員群組使用自己的資料庫實例。在某些情況下，也有助於生產環境。



屬於同一個網域的所有代理程式會協調其管理作業。當您透過 `hadbm` 指令變更資料庫配置時，所有代理程式會隨之變更配置。除非節點的主機上有執行的管理代理程式，否則您無法停止或重新啟動節點。但是，您可以在無法使用某些代理程式的情況下執行 `hadbm` 指令，以讀取 HADB 狀態或配置變數值。

使用下列管理用戶端指令處理管理網域：

- **hadbm createdomain**：使用特定主機建立管理網域。
- **hadbm extenddomain**：將主機增加至現有的管理網域。
- **hadbm deletedomain**：移除管理網域。
- **hadbm reducedomain**：從管理網域移除主機。
- **hadbm listdomain**：列出管理網域中所定義的所有主機。

## 儲存庫

管理代理程式會將資料庫配置儲存在**儲存庫**。由於儲存庫會在所有管理代理程式之間進行複製，因此容錯能力很高。將配置保留在伺服器上，可讓您從任何安裝管理用戶端的電腦上執行管理作業。

必須執行網域中的大部分管理代理程式，才可對儲存庫進行任何變更。因此，若網域中有  $M$  個代理程式，至少必須執行  $M/2 + 1$  個代理程式（無條件捨去到整數），才可對儲存庫進行變更。

若網域中有些主機無法使用（例如由於硬體故障），且您因為沒有配額而無法執行某些管理指令，請使用 `hadbm disablehost` 指令從網域移除故障的主機。

# 設定及配置資訊指南

## ▼ 設定及配置高可用性的 Application Server

- 1 如第1章中所述，決定效能及 QoS 需求與目標。
- 2 如第1章的第45頁的「設計決策」中所述，調整系統大小。
  - 應用程式伺服器實例的數目
  - HADB 節點與主機的數目
  - HADB 儲存容量
- 3 如第3章中所述，決定系統拓樸。  
如此會決定您要在與 Application Server 相同或不同的主機機器上安裝 HADB。
- 4 安裝應用程式伺服器實例以及相關子元件，例如 HADB 與 Web 伺服器。
- 5 建立網域與叢集。

- 6 配置 Web 伺服器軟體。
- 7 安裝負載平衡器外掛程式。
- 8 設定並配置負載平衡。
- 9 設定並配置 HADB 節點與 DRU。
- 10 為 HA 階段作業持續性配置 AS Web 容器與 EJB 容器。
- 11 為高可用性與階段作業容錯移轉部署並配置應用程式。
- 12 如果廣泛使用訊息傳送，請配置 JMS 叢集進行容錯移轉。  
如需更多資訊，請參閱「*Sun Java System Message Queue Administration Guide*」。

## 規劃部署

---

部署 Application Server 之前，首先要決定效能及可用性目標，然後據此決定硬體、網路及儲存需求。

本章包含以下各節：

- 第 35 頁的「建立效能目標」
- 第 41 頁的「規劃網路配置」
- 第 43 頁的「規劃可用性」
- 第 45 頁的「設計決策」
- 第 48 頁的「規劃 Message Queue 代理程式部署」

### 建立效能目標

簡單來說，高效能是指增加流量並縮短回應時間。除了這些基本目標，還可以透過決定下列項目以建立特定的目標：

- 要部署的應用程式與服務類型為何，以及用戶端要如何存取這些應用程式與服務？
- 哪些應用程式與服務需要高可用度？
- 應用程式具有階段作業狀態或無狀態？
- 系統必須支援的請求容量或流量為何？
- 系統必須支援多少同步運作的使用者？
- 可接受的使用者請求平均回應時間為何？
- 請求之間的平均考慮時間為何？

您可以使用模擬預期應用程式活動的遠端瀏覽器模擬器 (RBE) 工具或網站效能與基準軟體，計算部分度量。一般來說，RBE 與基準產品會產生同步運作的 HTTP 請求，然後報告指定的每分鐘請求數之回應時間。然後，您可以使用這些數據計算伺服器活動。

本章所述的計算結果並非絕對。由於您需要微調 Application Server 與應用程式的效能，因此請將結果當作工作時的參考點。

本節論述以下主題：

- 第 36 頁的「估計流量」
- 第 36 頁的「估計應用程式伺服器實例的負載」
- 第 39 頁的「估計 HADB 的負載」
- 第 41 頁的「估計頻寬需求」
- 第 42 頁的「估計尖峰負載」

## 估計流量

廣義來說，**流量**可用來測量 Application Server 執行的工作量。對於 Application Server 而言，流量可定義為每個伺服器實例每分鐘處理的請求數。高可用性應用程式也會使 HADB 有更高的流量需求，以定期儲存階段作業狀態資料。對於 HADB，流量可定義為每分鐘儲存的階段作業資料量，這是每分鐘 HADB 請求數與每個請求平均階段作業大小的乘積。

如下一節所述，Application Server 流量是由許多因素決定的函數，這些因素包含使用者請求的特性與大小、使用者數目，以及應用程式伺服器實例與後端資料庫的效能。您可以模擬的工作負荷量為基準，估計單一機器上的流量。

高可用性應用程式因為會定期將資料儲存至 HADB，因此會造成額外的經常性耗用時間。額外的經常性耗用時間量取決於資料量、資料量變更頻率及資料儲存頻率。前兩項因素與討論的應用程式相關，最後一個也會受到伺服器設定所影響。

HADB 流量可定義為每分鐘 HADB 請求數乘以每個請求的平均資料量。HADB 流量愈大表示需要愈多的 HADB 節點，以及愈大的存放區。

## 估計應用程式伺服器實例的負載

請考慮下列因素以估計應用程式伺服器實例的負載：

- 第 36 頁的「同步運作使用者人數上限」
- 第 37 頁的「考慮時間」
- 第 37 頁的「平均回應時間」
- 第 38 頁的「每分鐘請求數」

### 同步運作使用者人數上限

使用者透過 Web 瀏覽器或 Java 程式等用戶端，與應用程式互動。視使用者動作之不同，用戶端會定期將請求傳送至 Application Server。只要使用者的階段作業尚未到期或終止，該名使用者即會視為**目前的使用者**。估計同步運作使用者人數時，請納入所有目前的使用者。

下圖描述每分鐘處理的請求數(流量)與使用者人數的典型圖形。最初，當使用者人數增加時，流量會相對增加。但是，當同步運作請求數目增加時，伺服器效能會開始飽和，且流量會開始降低。

找出增加同步運作使用者會降低每分鐘可處理的請求數之時間點。此時間點表示達到最佳效能的時候，在此時間點之後流量會開始降低。一般會儘可能地以最佳流量來運作系統。您可能需要增加處理能力，以處理額外的負載並增加流量。

## 考慮時間

使用者不會無間斷地送出請求。使用者送出請求，伺服器接收並處理請求，然後傳回結果，此時使用者會花些時間才會再送出新的請求。一個請求與下一個請求之間的時間稱為**考慮時間**。

考慮時間取決於使用者的類型而定。例如，機器對機器的互動(如 Web 服務的互動)一般需要比人類使用者的互動更短的考慮時間。您在估計考慮時間時，必須將機器與人類的混合互動列入考量。

判斷平均考慮時間很重要。您可以使用此持續時間計算每分鐘必須完成的請求數，以及系統可支援的同步運作使用者人數。

## 平均回應時間

回應時間是指 Application Server 將請求結果傳回給使用者所需的時間。回應時間受到網路頻寬、使用者人數、送出的請求數目與類型，以及平均考慮時間等因素的影響。

在本節中，回應時間是指平均回應時間。每種請求類型各有其最短回應時間。但是，評估系統效能時，請根據所有請求的平均回應時間分析。

回應時間愈快，每分鐘處理的請求數愈多。但是，如下圖所述，當系統上的使用者人數增加時，回應時間也會開始增加，即使每分鐘請求數減少亦然：

抱歉：本版手冊尚未提供此圖。

類似此圖的系統效能圖會指出在過了特定時間點後，每分鐘請求數會與回應時間呈反比。每分鐘請求數減少幅度越大，回應時間的增加幅度就越大(以虛線箭頭表示)。

在圖中，尖峰負載點是每分鐘請求數開始減少的時間點。在此時間點之前，回應時間的計算不一定正確，因為未在公式中使用尖峰數目。在此時間點之後，由於每分鐘請求數與回應時間呈反比關係，所以管理員可以透過使用者人數上限及最大每分鐘請求數，更精準地計算回應時間。

使用下列公式判斷  $T_{\text{回應}}$ ，亦即尖峰負載時的回應時間(以秒為單位)：

$$T_{\text{回應}} = n/r - T_{\text{考慮}}$$

其中

- $n$  是同步運作使用者人數
- $r$  是伺服器每秒鐘接收的請求數
- $T_{\text{考慮}}$  是平均考慮時間(以秒為單位)

若要取得正確的回應時間結果，請一律將考慮時間列入方程式。

#### 範例 2-1 回應時間的計算

假設存在下列條件：

- 系統在尖峰負載時，可支援的同步運作使用者人數上限  $n$  為 5,000 人。
- 系統在尖峰負載時，可處理的最大請求數  $r$  為每秒 1,000 個。

平均考慮時間  $T_{\text{考慮}}$  為每個請求三秒。

回應時間的計算如下：

$$T_{\text{回應}} = n/r - T_{\text{考慮}} = (5000/1000) - 3 \text{ 秒} = 5 - 3 \text{ 秒}$$

因此，回應時間為兩秒。

計算系統的回應時間之後 (特別是在尖峰負載時)，請與應用程式可接受的回應時間做比較。回應時間及流量是決定 Application Server 效能的主要因素。

## 每分鐘請求數

若隨時知道同步運作使用者人數、這些使用者請求的回應時間，以及使用者平均考慮時間，即可計算每分鐘的請求數。一般會從估計系統上的同步運作使用者人數開始。

例如，執行網站效能軟體之後，管理員得知同時在網路銀行網站送出請求的使用者平均人數為 3,000 人。此人數取決於已登記成為網路銀行會員的使用者人數、使用者的銀行交易行為、使用者選擇送出請求的日期時間...等等。

因此，瞭解此資訊可讓您使用本節所述的每分鐘請求數公式，計算系統根據此使用者人數每分鐘可處理的請求數。由於每分鐘請求數與回應時間在尖峰負載時會形成反比，因此請決定是要接受較少的每分鐘請求數以換取較佳的回應時間，還是要接受較慢的回應時間以換取每分鐘更多的請求數。

實驗不同的可接受每分鐘請求數與回應時間臨界值，作為微調系統效能的起點。接著決定需要調整的系統區域。

求解上一節方程式中的  $r$  得出：

$$r = n / (T_{\text{回應}} + T_{\text{考慮}})$$

#### 範例 2-2 每秒請求數的計算

假設下列值：

- $n = 2,800$  名同步運作使用者
- $T_{\text{回應}} = 1$  (每個請求平均回應時間為一秒)
- $T_{\text{考慮}} = 3$  (平均考慮時間為三秒)

每秒請求數的計算如下：

範例 2-2 每秒請求數的計算 (續)

$$r = 2800 / (1+3) = 700$$

因此，每秒請求數為 700，而每分鐘請求數為 42000。

## 估計 HADB 的負載

若要計算 HADB 的負載，請考慮下列因素：

- 第 39 頁的「HTTP 階段作業持續性頻率」
- 第 39 頁的「HTTP 階段作業大小與範圍」
- 第 40 頁的「有狀態的階段作業 Bean 檢查點檢查」

如需有關配置階段作業持續性的指示，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 9 章「配置高可用性階段作業持續性和容錯移轉」。

### HTTP 階段作業持續性頻率

HADB 接收的每分鐘請求數取決於**持續性頻率**。持續性頻率會決定 Application Server 將 HTTP 階段作業資料儲存至 HADB 的頻率。

持續性頻率選項包括：

- **Web 方法** (預設值)：伺服器會在每次 HTTP 回應時儲存階段作業資料。此選項可保證儲存的階段作業資訊是最新的，但其會造成 HADB 的高流量。
- **以時間為基礎**：階段作業會依指定時間間隔儲存。此選項會降低 HADB 的流量，但不保證階段作業資訊是最新的。

下表是持續性頻率選項的優缺點摘要。

表 2-1 持續性頻率選項的比較

持續性頻率選項	優點	缺點
Web 方法	保證可提供最新的階段作業資訊。	可能增加回應時間並降低流量。
以時間為基礎	改善回應時間，並有可能增加流量。	較無法保證可在應用程式伺服器實例故障之後仍能提供最新的階段作業資訊。

### HTTP 階段作業大小與範圍

每個請求的階段作業大小取決於階段作業中所儲存的階段作業資訊量。

**提示** – 若要改善整體效能，請儘可能降低階段作業中的資訊量。



可透過 **持續性範圍** 設定，微調每個請求的階段作業大小。請從 HTTP 階段作業持續性範圍的下列選項中選擇：

- **session**：每次伺服器將階段作業資訊儲存至 HADB 時，便會序列化並儲存整個階段作業物件。
- **modified-session**：伺服器僅會在階段作業已修改時儲存階段作業。伺服器可截取對 Bean 的 `setAttribute()` 方法之呼叫，以偵測有無修改。此選項不會偵測對內部物件的直接修改，因此在這類情況下，必須編寫 SFSB 程式碼以明確呼叫 `setAttribute()`。
- **modified-attribute**：伺服器僅會儲存自上次儲存階段作業以來，已修改 (已插入、已更新或已刪除) 的屬性。這與 **modified-session** 有相同的缺點，但是若套用適當，則可大幅降低 HADB 的寫入流量需求。

若要使用此選項，應用程式必須：

- 在每次修改階段作業狀態時，呼叫 `setAttribute()` 或 `removeAttribute()`。
- 確定各屬性之間沒有交叉參照。
- 在多個屬性之間分布階段作業狀態，或者至少在唯讀屬性和可修改屬性之間分布階段作業狀態。

下表是持續性範圍選項的優缺點摘要。

表 2-2 持續性範圍選項的比較

持續性範圍選項	優點	缺點
modified-session	當請求不修改階段作業狀態時，提供較優良之請求回應時間。	執行 Web 方法期間 (一般是 <code>doGet()</code> 或 <code>doPost()</code> )，應用程式必須呼叫階段作業方法： <ul style="list-style-type: none"><li>■ <code>setAttribute()</code>，若已變更屬性。</li><li>■ <code>removeAttribute()</code>，若已移除屬性。</li></ul>
session	應用程式沒有限制。	與 <b>modified-session</b> 和 <b>modified-attribute</b> 選項相較下，可能使流量及回應時間更差。
modified-attribute	在請求的階段作業狀態修改比例很低的情況下，改善請求的流量與回應時間。	當指定請求的階段作業狀態修改比例接近 60% 時，流量會降低且回應時間會變慢。在這種情況下的效能是所有選項中最差的，因為將屬性分割成個別的記錄會造成額外的經常性耗用時間。

## 有狀態的階段作業 Bean 檢查點檢查

對於 SFSB 階段作業持續性，HADB 的負載與下列項目相關：

- 啟用檢查點檢查的 SFSB 數目。
- 已選取哪些 SFSB 方法進行檢查點檢查，以及這些方法使用的頻率。
- 階段作業物件的大小。
- 哪些方法會有所異動。



檢查點檢查一般會在完成 SFSB 的任何作業事件之後發生，即使作業事件回復亦然。

如需更佳的效能，請指定一小組方法進行檢查點檢查。進行檢查點檢查的資料大小與檢查點檢查的頻率，會決定指定用戶端互動之回應時間的額外經常性耗用時間。

## 規劃網路配置

規劃如何將 Application Server 整合到網路時，請估計頻寬需求，並將網路規劃成可符合使用者的效能需求。

本節包含以下主題：

- [第 41 頁的「估計頻寬需求」](#)
- [第 41 頁的「計算所需的頻寬」](#)
- [第 42 頁的「估計尖峰負載」](#)
- [第 42 頁的「配置子網路」](#)
- [第 43 頁的「選擇網路卡」](#)
- [第 43 頁的「HADB 的網路設定」](#)
- [第 44 頁的「找出故障類別」](#)

### 估計頻寬需求

若要決定所需的網路大小與頻寬，請先判斷網路流量並找出其尖峰時段。檢查整體流量是否只在特定時刻、一週或一個月內的特定日子達到尖峰，然後判斷該尖峰的持續時間。

在尖峰負載期間，網路中的封包數會達到最高。一般來說，若要進行尖峰負載設計，請依照處理 100% 尖峰量的目標來延伸系統規模。但是請記住，當網路運作異常時，即使您已經延伸規模，可能還是不一定能處理 100% 的尖峰量。

例如，假設在尖峰負載時，5% 的使用者在存取 Application Server 上部署的應用程式時，偶爾會無法立刻存取網路。在 5% 的使用者中，估計有多少使用者在第一次嘗試之後重試存取。重試之後，並非所有使用者皆能成功存取，在這些不成功的部分中，又會有另外比例的人數進行重試。因此，顯示的尖峰時間會比較長，因為使用者會不斷嘗試存取，而造成尖峰使用的時間拉長。

### 計算所需的頻寬

根據[第 35 頁的「建立效能目標」](#)中計算的結果，判斷在網站上部署 Application Server 時所需的額外頻寬。

根據存取方法 (T1 專線、ADSL、纜線數據機...等等)，計算處理估計的負載所需增加的頻寬量。例如，假設您的網站使用 T1 或更高速的 T3 專線。根據網站每秒所產生的平均請求數及最大尖峰負載，在指定頻寬下估計所需的網路線路數目。使用網站分析與監視工具計算這些數據。

**範例 2-3 所需頻寬的計算**

一條 T1 專線可處理 1.544 Mbps。因此，四條 T1 專線的網路可處理約 6 Mbps 的資料。假設傳送回用戶端的平均 HTML 網頁為 30 KB，此內含四條 T1 專線的網路每秒可處理下列流量：

$6,176,000 \text{ 位元} / 8 \text{ 位元} = \text{每秒 } 772,000 \text{ 個位元組}$

$\text{每秒 } 772,000 \text{ 位元組} / 30 \text{ KB} = \text{約每秒 } 25 \text{ 頁同步運作回應頁面。}$

假設一整天的負載很平均，在每秒 25 頁的流量下，此系統每小時可處理 90,000 頁 (25 x 60 秒 x 60 分鐘)，因此每天最多可處理 2,160,000 頁。若最大尖峰負載大於此負載，請隨之增加頻寬。

## 估計尖峰負載

在現實情況下，一整天的負載不可能很平均。您必須判斷何時會發生尖峰負載、尖峰負載持續的時間，以及尖峰負載佔總負載的比例。

**範例 2-4 尖峰負載的計算**

若尖峰負載持續兩小時，並佔 2,160,000 頁面總負載的 30%，表示必須在一天兩小時內在 T1 專線上傳送 648,000 個頁面。

因此，若要在這兩小時內容納尖峰負載，請根據下列計算增加 T1 專線的數目：

$648,000 \text{ 頁} / 120 \text{ 分鐘} = \text{每分鐘 } 5,400 \text{ 頁}$

$\text{每分鐘 } 5,400 \text{ 頁} / 60 \text{ 秒} = \text{每秒 } 90 \text{ 頁}$

如果四條網路線每秒可處理 25 頁，90 頁近似於 25 的四倍，因此需要四倍的網路線，亦即 16 條網路線。這 16 條網路線的用途，是處理實際上最多 30% 的尖峰負荷。其他 70% 的負載毫無疑問可在一天的其他時間由這些數目的網路線處理。

## 配置子網路

如果使用個別層級拓樸，其中應用程式伺服器實例與 HADB 節點位於不同的主機機器上，您可以透過使所有 HADB 節點位於不同的子網路，以改善效能。這是因為 HADB 使用使用者資料封協定 (UDP)。使用不同的子網路可降低子網路外部機器上的 UDP 流量。但請注意，所有 HADB 節點必須位於相同的子網路上。

只要所有節點與管理代理程式位於相同的子網路上，您仍可從不同的子網路執行管理用戶端。所有節點代理程式內應該皆可存取所有主機與連接埠，且防火牆、UDP 的封鎖功能...等等都不得封鎖此節點。

HADB 使用 UDP 多重播送，因此內含 HADB 節點的所有子網路皆須配置為可進行多重播送。

## 選擇網路卡

若要取得最佳的頻寬及理想的網路效能，請在代管 Application Server 與 HADB 節點的伺服器之間，至少使用 100 Mbps 的乙太網路卡，最好是 1 Gbps 的乙太網路卡。

## HADB 的網路設定

**備註** – HADB 使用 UDP 多重播送，因此您必須啟用系統路由器與主機網路介面卡的多重播送。若 HADB 跨多個子網路，則也必須啟用子網路之間路由器的多重播送。為取得最佳結果，請將所有 HADB 節點置於相同的網路。應用程式伺服器實例可在不同的子網路上。

下列建議可讓 HADB 在網路中的運作最佳。

- 使用交換路由器使每個網路介面皆有專屬 100 Mbps 或更高的乙太網路通道。
- 在代管四個或更多 HADB 節點的多重 CPU 機器上執行 HADB 時，請使用 1 Gbps 的乙太網路卡。若階段作業平均大小大於 50 KB，請使用 1 Gbps 的乙太網路卡，即使每部機器的 HADB 節點少於四個亦然。
- 若懷疑 HADB 節點內有網路瓶頸：
  - 請在 HADB 伺服器上執行網路監視軟體以診斷問題。
  - 請考慮以 1 Gbps 的乙太網路卡取代網路中所有的 100 Mbps 乙太網路卡。

## 規劃可用性

本小節包含下列主題：

- [第 43 頁的「最佳化可用性」](#)
- [第 44 頁的「使用叢集改善可用性」](#)
- [第 44 頁的「增加系統的備援」](#)

## 最佳化可用性

若要規劃系統與應用程式的可用性，請評估存取不同應用程式之使用者群組的可用性需求。例如，付費的外部使用者與企業夥伴常比內部使用者有更高的服務品質 (QoS) 期望。因此，內部使用者可能比付費的外部用戶較能接受應用程式功能、應用程式或伺服器無法使用。

下圖描述為了減少可能發生的事件，成本與複雜度會如何增加。在此連續曲線的一端，簡易負載平衡的叢集即可容許本土化應用程式、中介軟體與硬體故障。在圖表的另一端，位置不同的叢集可降低影響整個資料中心的重大災難。

若要實現理想的投資報酬率，通常最好找出應用程式內各功能的可用性需求。例如，保險報價系統絕對不能無法使用(可能會失去新業務)，但是帳號管理功能(現有用戶可由此檢視其目前的保險項目)暫時無法使用則不太可能會失去現有的用戶。

## 使用叢集改善可用性

在最基本的層級中，叢集是一組應用程式伺服器實例(通常代管於多個實體伺服器上)，並讓用戶端以為是單一實例。如此可提供水平延展性，以及比單一機器上的單一實例更高的可用性。此基本叢集層級結合 Application Server 的 HTTP 負載平衡器外掛程式使用，該外掛程式接受 HTTP 與 HTTPS 請求，並會轉寄給叢集中其中一個應用程式伺服器實例。ORB 與整合的 JMS 代理程式也會對應用程式伺服器叢集執行負載平衡。若實例出現故障而無法使用(因為網路故障)或無回應，請求僅會重新導向至現有的可用機器。負載平衡器還可以識別出現故障的實例是否已回復，並會隨之重新分配負載。

HTTP 負載平衡器也提供**運作狀態檢查程式**，該程式可監視伺服器與特定 URL，以判斷其是否可用。您必須謹慎管理運作狀態檢查的經常性耗用時間，使其本身不會成為龐大的處理負擔。

對於無狀態應用程式或價值不高、僅需簡單使用者作業事件的應用程式，通常只需要簡易負載平衡的叢集。對於有狀態的關鍵性應用程式，請考慮使用 HADB 取得階段作業持續性。如需 HADB 的簡介，請參閱「Application Server Administration Guide」第 1 章中第 27 頁的「高可用性資料庫」。

若要執行應用程式的線上升級，最好將一應用程式伺服器實例分組為多個叢集。Application Server 可同時**靜止**應用程式與實例。靜止是以控制的方式讓實例(或實例群組)或特定應用程式離線的功能，而且其不會影響實例或應用程式目前提供服務的使用者。當一個實例靜止時，新的使用者會由其他實例上已升級的應用程式提供服務。此應用程式升級類型稱為**輪替式升級**。如需有關升級即時應用程式的更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的「在維持可用性的情況下升級應用程式」。

## 增加系統的備援

達成高可用性的一個方式是增加系統的軟硬體備援。當一個單元故障時，備援單元會接管。這又稱為容錯。一般來說，若有最佳的高可用性，請判斷並移除系統中每個可能的故障點。

### 找出故障類別

備援層級取決於系統需要容許的故障類別(故障的類型)。以下是一些故障類別範例：

- 系統程序
- 機器

- 電源供應
- 磁碟
- 網路故障
- 建築物失火或其他可避免的災難
- 無法預期的天災

複製系統程序可容許單一系統程序故障，以及單一機器故障。將複製的鏡像 (成對) 機器接到不同的電源供應，即可容許單一電源故障。透過將鏡像機器保留在不同的建築物中，可容許單一建築物失火。透過將鏡像機器保留在不同的地理位置，可容許地震等天災。

## 使用 HADB 備援單元改善可用性

若要改善可用性，一律請如第 35 頁的「建立效能目標」中所述，在資料備援單元 (DRU) 中使用 HADB 節點。

## 使用 HADB 備用節點改善容錯

使用備用節點改善容錯。雖然備用節點不是必要項目，但可提供最佳的可用性。

## 規劃容錯移轉容量

容錯移轉容量規劃是指決定需要額外增加至 Application Server 部署的伺服器與程序數目，如此一來，當伺服器或程序故障時，系統可緊密地回復資料並繼續處理。若系統超載，可能會產生程序或伺服器故障，造成回應時間變慢或甚至失去整個服務。為這類事件做足準備對於成功部署非常重要。

若要維持容量，特別是在尖峰負載時，請將執行應用程式伺服器實例的備用機器增加至現有的部署中。

例如，假設某部系統有兩部機器，各執行一個應用程式伺服器實例。這些機器在尖峰負載時每秒總共可處理 300 個請求。如果一部機器無法使用，系統將只能處理 150 個請求 (假設機器之間平均分散負載)。因此將會無法處理尖峰負載期間的一半請求。

# 設計決策

設計決策包含您是為尖峰負載還是持續狀態負載設計系統、不同角色的機器數目及其大小。

## 為尖峰負載或持續狀態負載設計

在一般部署中，持續狀態工作負荷量與尖峰工作負荷量不同。

- 若系統的設計目的是處理尖峰負載，則可維持預期的最大使用者與請求負載，而不會使回應時間變慢。這表示系統可處理預期系統負載的極限情況。若尖峰負載與持續狀態負載之間的差異很大，則為尖峰負載設計表示會耗費許多金錢在經常閒置的資源上。
- 若系統的設計目的是處理持續狀態負載，則不會有處理預期尖峰負載所需的所有資源。因此，發生尖峰負載時，系統的回應時間會變慢。

系統預期處理尖峰負載的頻率，將決定您要為尖峰負載還是持續性狀態進行設計。

若尖峰負載經常發生 (例如每天多次)，便有必要擴充容量以處理負載。若系統 90% 的時間以持續狀態運作，10% 的時間在尖峰，則適合部署依照持續狀態負載所設計的系統。這表示系統的回應時間僅會在 10% 的時間內變慢。決定系統在尖峰運作時的頻率或持續時間是否有必要增加系統資源。

## 調整系統大小

根據應用程式伺服器實例的負載、HADB 的負載及容錯移轉需求，您可以判斷：

- 第 46 頁的「應用程式伺服器實例的數目」
- 第 46 頁的「HADB 節點的數目」
- 第 47 頁的「HADB 主機的數目」
- 第 48 頁的「HADB 儲存容量」

### 應用程式伺服器實例的數目

若要判斷所需的應用程式伺服器實例 (主機) 數目，請根據第 36 頁的「估計應用程式伺服器實例的負載」中所述的因素，針對每個應用程式伺服器實例評估您的環境，但是每個實例可使用多個中央處理器 (CPU)。

### HADB 節點的數目

一般來說，請為系統的每個 CPU 規劃一個 HADB 節點。例如，為具有兩個 CPU 的機器使用兩個 HADB 節點。

---

**備註** – 若每部機器有多個 HADB 節點 (例如，如果您使用較大型的機器)，則必須確定這些機器上有足夠的備援與延展性；例如，多個不斷電系統及獨立的磁碟控制器。

---

或者使用下列程序。

## ▼ 判斷所需的 HADB 節點數目

### 1 判斷下列參數：

- 同步運作使用者人數上限  $n_{\text{使用者}}$ 。
- 平均 BLOB 大小  $s$ 。



- 每位使用者的最大作業事件率，稱為 NTPS。
- 2 判斷主要資料量大小上限 (以 GB 為單位)  $V_{\text{資料}}$ 。  
使用下列公式：  
$$V_{\text{資料}} = n_{\text{使用者}} \cdot s$$
- 3 判斷最大 HADB 資料傳輸速率  $R_{\text{dt}}$ 。  
這會反映從應用程式端傳送至 HADB 中的資料量。使用下列公式：  
$$R_{\text{dt}} = n_{\text{使用者}} \cdot s \cdot \text{NTPS}$$
- 4 判斷節點數目  $N_{\text{節點}}$ 。  
使用下列公式：  
$$N_{\text{節點}} = V_{\text{資料}} / 5\text{GB}$$
  
由於節點是成對運作，因此請將此值無條件進入成偶數。

## HADB 主機的數目

根據資料傳輸需求判斷 HADB 主機的數目。此計算假設所有主機皆具有類似的硬體配置與作業系統，並有必要的資源可容納主機執行的節點。

### ▼ 計算主機的數目

- 1 判斷最大主機資料傳輸速率  $R_{\text{max}}$ 。  
由於此值與網路及主機硬體相關，因此請憑經驗判斷此值。請注意，此值與上一節中所判斷的最大 HADB 資料傳輸速率  $R_{\text{dt}}$  不同。
- 2 判斷容納此資料所需的主機數目。  
更新分散至數個主機  $N_{\text{主機}}$  的資料量  $V$ ，會造成每個主機接收約  $4V/N_{\text{主機}}$  的資料。使用下列公式判斷容納此資料量所需的主機數目：  
$$N_{\text{主機}} = 4 \cdot R_{\text{dt}} / R_{\text{max}}$$
  
將此值無條件進入成最接近的偶數，以針對每個 DRU 取得相同的主機數目。
- 3 在每個 DRU 上為備用節點增加一個主機。  
如果其他主機各執行  $N$  個資料節點，請讓此主機執行  $N$  個備用節點。如此可允許單一機器故障關閉  $N$  個資料節點。  
每個主機至少需要執行一個節點，因此如果節點的數目小於主機的數目 ( $N_{\text{節點}} < N_{\text{主機}}$ )，請調整  $N_{\text{節點}}$ ，使其等於  $N_{\text{主機}}$ 。如果節點的數目大於主機的數目 ( $N_{\text{節點}} > N_{\text{主機}}$ )，則相同主機上可執行數個節點。

## HADB 儲存容量

HADB 以近乎直線的比例遞增更多節點，直到超過網路容量為止。您必須在專屬磁碟上配置每個節點的儲存裝置。所有節點必須在儲存裝置上配置相等的空間。請務必於本機磁碟上配置儲存裝置。

假設預期的階段作業資料大小是  $x$  MB。HADB 會在鏡像節點上複製資料，因此需要  $2x$  MB 的儲存容量。此外，HADB 使用索引加速存取資料。兩個節點需要額外的  $2x$  MB 供索引使用，因此所需的總儲存容量為  $4x$ 。因此，HADB 的預期儲存容量需求是預期資料量的四倍。

爲了日後擴充而不失去 HADB 資料，您必須提供線上升級的額外儲存容量，因爲您可能需要在增加新節點後重新分段資料。在這種情況下，資料裝置上需要類似的額外空間量 ( $4x$ )。因此，預期儲存容量是預期資料量的八倍。

此外，HADB 會如下使用磁碟空間：

- 記錄緩衝區的暫存空間。此空間是記錄緩衝區大小的四倍。記錄緩衝區會追蹤與資料相關的作業。記錄緩衝區大小的預設值爲 48 MB。
- 用於內部管理的空間。此空間是儲存裝置大小的 1%。

下表是階段作業資料爲  $x$  MB 的 HADB 儲存空間需求摘要。

表 2-3 階段作業大小爲 X MB 的 HADB 儲存空間需求

條件	所需的 HADB 儲存空間
不需要在線上時，HADB 節點的增加或移除。	$4x$ MB + (4*記錄緩衝區大小) + 1% 的裝置大小
需要在線上時，HADB 節點的增加或移除。	$8x$ MB + (4*記錄緩衝區大小) + 1% 的裝置大小

若 HADB 的裝置空間不足，即不會接受插入或更新資料的用戶端請求。但會接受刪除作業。若 HADB 的裝置空間不足，則會傳回錯誤碼 4593 或 4592，並將對應的錯誤訊息寫入歷史檔案。如需有關這些訊息的更多資訊，請參閱「Sun Java System Application Server 9.1 Error Message Reference」中的第 14 章「HADB Error Messages」。

## 規劃 Message Queue 代理程式部署

Java Message Service (JMS) API 是一種允許 J2EE 應用程式和元件建立、傳送、接收以及讀取郵件的訊息傳送標準。它可啓用可靠的非同步鬆耦合分散式通訊。實作 JMS 的 Sun Java System Message Queue 與 Application Server 整合在一起，可讓您建立諸如訊息驅動 Bean (MDB) 之類的元件。

Sun Java System Message Queue (MQ) 使用由 J2EE 連接器架構規格 (JCA) 1.5 定義的**連接器模組** (亦稱爲資源配接卡) 與 Application Server 整合在一起。連接器模組是增加 Application Server 功能的標準化方式。部署至 Application Server 的 J2EE 元件使用透過連



接器模組整合的 JMS 提供者，交換 JMS 訊息。依預設，JMS 提供者是 Sun Java System Message Queue，但是如果需要使用不同的 JMS 提供者，則只要是實作 JCA 1.5 的提供者皆可。

在 Application Server 中建立 JMS 資源會在背景中建立連接器資源。因此，每個 JMS 作業均會在背景中呼叫連接器執行階段並使用 MQ 資源介面。

除了使用資源配接卡 API 之外，Application Server 還使用其他 MQ API 提供與 MQ 的更佳整合。此緊密的整合可啟用連接器容錯移轉、外送連線的負載平衡，以及內送訊息至 MDB 的負載平衡等功能。這些功能可讓您使訊息傳送流量容錯且高度可用。

## 多個代理程式叢集

MQ Enterprise Edition 支援使用多個互連的代理程式實例，稱為**代理程式叢集**。對於代理程式叢集，用戶端連線會分散至叢集中的所有代理程式。叢集可提供水平可延伸性並可提高可用性。

單一訊息代理程式可延伸至約八個 CPU，並為一般應用程式提供足夠的流量。若代理程式程序故障，則會自動重新啟動。但是，隨著連線至代理程式的用戶端數目增加，以及提供的訊息數目增加，代理程式終會超過檔案描述元的數目與記憶體等限制。

在叢集中有多個代理程式而非只有一個代理程式，可讓您：

- 即使單一機器發生硬體故障，仍可提供訊息傳送服務。
- 執行系統維護時可縮短停機時間。
- 容納具有不同使用者儲存庫的工作群組。
- 處理防火牆限制。

但是，具有多個代理程式並不保證代理程式故障時，進行中的作業事件會繼續在替代代理程式上執行。當 MQ 使用叢集中不同的代理程式重新建立故障的連線時，會失去作業事件訊息傳送並回復進行中的作業事件。除了無法完成作業事件之外，使用者應用程式將不會受到影響。由於可繼續使用連線，因此確認了服務容錯移轉功能。

因此，MQ 不支援在叢集中的高可用持續性訊息傳送。若代理程式在故障後重新啟動，則將自動回復並完成持續性訊息的傳遞。持續性訊息可儲存在資料庫中或儲存於檔案系統上。但是，若代管代理程式的機器未從硬體故障中回復，則可能遺失訊息。

內含適用於 Sun Message Queue 之 Sun Cluster Data Service 的 Solaris 平台，支援不需設定即可針對持續性訊息進行容錯移轉。此配置運用 Sun Cluster 的全域檔案系統及 IP 容錯移轉，提供實在的高可用性並隨附於 Java Enterprise System 中。

## 主代理程式與用戶端同步化

在多個代理程式配置中，每個目標會在叢集中所有代理程式上遭到複製。每個代理程式知道在所有其他代理程式上的目標註冊之訊息用戶。因此，每個代理程式可從其自己的直接連線訊息生產者，將訊息路由至遠端的訊息用戶，並從遠端生產者將訊息傳遞至自己的直接連線用戶。

在叢集配置中，每個訊息生產者直接連線的代理程式會為該生產者傳送的訊息執行路由。由此，持續性訊息會再透過訊息的**主代理程式**進行儲存與路由。

當管理員建立或銷毀代理程式上的目標時，此資訊會自動傳播至叢集中的所有其他代理程式。同理，當訊息用戶向主代理程式註冊，或從其主代理程式中斷連線時(明確指定、由於用戶端或網路故障，或由於其主代理程式關閉)，有關此用戶的資訊會在整個叢集中傳播。有關長期訂閱的資訊也會以類似的方式傳播至叢集中的所有代理程式。

## 配置 Application Server 以使用 Message Queue 代理程式

Application Server 的 Java 訊息服務表示 Message Queue 的連接器模組(資源配接卡)。您可以透過管理主控台或 `asadmin` 指令行公用程式來管理 Java 訊息服務。

MQ 代理程式(JMS 主機)會在與 Application Server 程序不同的 JVM 中執行。如此可讓多個應用程式伺服器實例或叢集共用同一組 MQ 代理程式。

在 Application Server 中，**JMS 主機**是指 MQ 代理程式。Application Server 的 Java 訊息服務配置內含 JMS 主機清單(亦稱為 `AddressList`)，其中包括所有即將使用的 JMS 主機。

### 使用管理主控台管理 JMS

在管理主控台中，您可以使用 Java 訊息服務節點為特定的配置設定 JMS 特性。您可以設定 [重新連線間隔] 與 [重新連線嘗試] 等特性。如需更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的第 4 章「配置 Java 訊息服務資源」。

Java 訊息服務節點下的 JMS 主機節點包含 JMS 主機的清單。您可以在清單中增加及移除主機。對於每個主機，您可以設定主機名稱、連接埠號碼，以及管理使用者名稱與密碼。依預設，JMS 主機清單包含一個稱為 `default_JMS_host` 的 MQ 代理程式，其表示與 Application Server 整合的本機 MQ 代理程式。

配置 JMS 主機清單以納入叢集中所有的 MQ 代理程式。例如，若要設定包含三個 MQ 代理程式的叢集，請在 Java 訊息服務內為每個代理程式增加一個 JMS 主機。Message Queue 用戶端使用 Java 訊息服務中的配置資訊，與 MQ 代理程式通訊。

### 使用 asadmin 管理 JMS

除了管理主控台，還可以使用 `asadmin` 指令行公用程式管理 Java 訊息服務與 JMS 主機。使用下列 `asadmin` 指令：

- 配置 Java 訊息服務屬性：`asadmin set`
- 管理 JMS 主機：
  - `asadmin create-jms-host`
  - `asadmin delete-jms-host`

- `asadmin list-jms-hosts`
  - 管理 JMS 資源：
    - `asadmin create-jms-resource`
    - `asadmin delete-jms-resource`
    - `asadmin list-jms-resources`
- 如需有關這些指令的更多資訊，請參閱「Sun Java System Application Server 9.1 Reference Manual」或相對應的線上手冊。

## Java 訊息服務類型

Application Server 與 MQ 代理程式之間的整合有兩種類型：LOCAL 與 REMOTE。您可以在管理主控台的 Java 訊息服務頁面上設定此 Type 屬性。

### LOCAL Java 訊息服務

如果 Type 屬性是 LOCAL，Application Server 會啟動與停止 MQ 代理程式。當 Application Server 啟動時，會隨即啟動指定為預設 JMS 主機的 MQ 代理程式。同理，當應用程式伺服器實例關閉時，會隨即關閉 MQ 代理程式。LOCAL 類型最適合於獨立的應用程式伺服器實例。

對於 LOCAL 類型，請使用 Start Arguments 屬性指定 MQ 代理程式的啟動參數。

### REMOTE Java 訊息服務

如果 Type 屬性是 REMOTE，Application Server 會使用外部配置的代理程式或代理程式叢集。在此情況下，您必須從 Application Server 分別啟動與停止 MQ 代理程式，並使用 MQ 工具配置與調校代理程式或代理程式叢集。REMOTE 類型最適合於應用程式伺服器叢集。

對於 REMOTE 類型，您必須使用 MQ 工具指定 MQ 代理程式啟動參數。可忽略 Start Arguments 屬性。

## 預設 JMS 主機

您可以在管理主控台的 Java 訊息服務頁面上指定預設 JMS 主機。若 Java 訊息服務的類型為 LOCAL，則 Application Server 會在應用程式伺服器實例啟動時啟動預設的 JMS 主機。

若要使用 MQ 代理程式叢集，請刪除預設的 JMS 主機，然後增加叢集中的所有 MQ 代理程式作為 JMS 主機。在此情況下，預設 JMS 主機會成為 JMS 主機清單中的第一個 JMS 主機。

您也可以明確地將預設 JMS 主機設定為其中一個 JMS 主機。當 Application Server 使用 Message Queue 叢集時，預設 JMS 主機會執行 MQ 專屬的指令。例如，為 MQ 代理程式叢集建立實體目標時，預設 JMS 主機會執行指令以建立實體目標，但叢集中的所有代理程式會使用實體目標。

## 部署方案範例

若要符合您的訊息傳送需求，請修改 Java 訊息服務與 JMS 主機清單，以滿足您的部署、效能與可用性需求。以下各節說明一些典型的方案。

為取得最佳的可用性，若訊息傳送需求不單只是為了 Application Server，請在不同的機器上部署 MQ 代理程式與 Application Server。或在每部機器上執行一個應用程式伺服器實例與一個 MQ 代理程式實例，直到訊息傳送能力足夠為止。

### 預設部署

自動安裝 Application Server 會建立 Domain Administration Server (DAS)。依預設，DAS 的 Java 訊息服務類型是 LOCAL。因此，啟動 DAS 也會啟動其預設 MQ 代理程式。

建立新的網域也會建立新的代理程式。依預設，當您將獨立的伺服器實例或叢集增加至網域時，其 Java 訊息服務會配置為 REMOTE，而其預設 JMS 主機是 DAS 所啟動的代理程式。

[第 52 頁的「預設部署」](#)說明使用包含三個實例的應用程式伺服器叢集之預設部署範例。

### 搭配使用 MQ 代理程式叢集與應用程式伺服器叢集

若要配置應用程式伺服器叢集使用 MQ 代理程式叢集，請增加所有 MQ 代理程式作為 Application Server 的 Java 訊息服務中之 JMS 主機。所有建立的 JMS 連線工廠及部署的 MDB 會接著使用指定的 JMS 配置。

下圖描述的部署範例，在代理程式叢集中包含三個 MQ 代理程式，並在叢集中包含三個應用程式伺服器實例。

### 指定特定應用程式的 MQ 代理程式叢集

在某些情況下，應用程式使用的 MQ 代理程式叢集，可能必須與應用程式伺服器叢集使用的不同。[第 52 頁的「指定特定應用程式的 MQ 代理程式叢集」](#)說明此類方案的範例。若要執行此作業，請使用 JMS 連線工廠的 `AddressList` 特性或 MDB 部署描述元中的 `activation-config` 元素，指定 MQ 代理程式叢集。

如需有關配置連線工廠的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「JMS 連線工廠」。如需有關 MDB 的更多資訊，請參閱「Sun Java System Application Server 9.1 Developer's Guide」中的「Using Message-Driven Beans」。

### 應用程式用戶端

當應用程式用戶端或獨立應用程式第一次存取 JMS 管理式物件時，用戶端 JVM 會從伺服器擷取 Java 訊息服務配置。用戶端 JVM 必須重新啟動，才可使用對於 JMS 服務所做的進一步變更。

## 選取拓樸

---

如第 1 章的「Application Server」中所述，評估與效能相關的因素之後。拓樸是機器、應用程式伺服器實例與 HADB 節點的排列方式，及其之間的通訊流程。

基本的部署拓樸有兩種：兩個拓樸有共同的基本結構：叢集中的多個應用程式伺服器實例、HADB 節點的鏡像組，以及 HADB 備用節點。兩個拓樸皆需要一組共用配置設定才可正常運作。

本章內容討論：

- 第 53 頁的「共同需求」(適用於兩個拓樸)。
- 兩個拓樸：
  - 第 55 頁的「並置拓樸」- 應用程式伺服器實例與 HADB 節點位於相同的機器上。
  - 第 59 頁的「個別層級拓樸」- 應用程式伺服器實例與 HADB 節點位於不同的機器上。
- 第 63 頁的「決定要使用的拓樸」

### 共同需求

本節說明兩個拓樸的共同需求：

- 第 53 頁的「一般需求」
- 第 54 頁的「HADB 節點與機器」
- 第 54 頁的「負載平衡器配置」

### 一般需求

兩個拓樸皆須符合下列一般需求：

- 代管 HADB 節點的機器必須成對。亦即機器數目必須為偶數。

- 每個資料備援單元 (DRU) 必須有相同的機器數目。以這種方式建立 HADB 資料庫，使鏡像 (成對) 的節點位於與主要節點不同的 DRU 上。
- 每部代管 HADB 節點的機器皆必須有本機磁碟儲存體，用以儲存 HADB 中所保留的所有資訊。
- 代管 HADB 節點的機器必須執行相同的作業系統。最好使用在配置與效能方面完全相同或幾乎相同的機器。
- 若要將 HTTP 與 SFSB 階段作業資訊保留至 HADB，應用程式伺服器實例必須在叢集中，且滿足所有相關的需求。如需有關配置叢集的更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 6 章「使用 Application Server 叢集」。
- 代管應用程式伺服器實例的機器在配置與效能方面必須儘可能相同。這是因為負載平衡器外掛程式使用循環策略進行負載平衡，若不同類別的機器代管實例，將無法在這些機器之間達成最佳的負載平衡。
- 每個 DRU 最好都有專屬的不斷電系統 (UPS)。

## HADB 節點與機器

每個 DRU 包含 HADB 中完整的資料複本，並可在其他 DRU 無法使用時繼續處理請求。但是，若一個 DRU 中的節點與其在另一個 DRU 中的鏡像節點同時故障，則將會遺失部分資料。因此，設定系統時請特別注意，不要讓兩個 DRU 受到單點故障的影響，例如電源故障或磁碟故障。

---

**備註** – 每個 DRU 必須在完全無關的備援系統上執行。

---

設定 HADB 節點與機器時，請遵循這些指導方針進行：

- 若要增加容量與流量，請成對增加節點，且是每個 DRU 一個節點。
- 設定每個 DRU，使其備用節點數目等於每部機器上執行的節點數目。這是因為如果配置中的每部機器執行  $n$  個資料節點，單一機器故障便會關閉  $n$  個節點。
- 在所有機器上執行相同數目的 HADB 節點，可盡量均勻地平衡負載。



---

**注意** – 請勿於相同的機器上執行不同 DRU 的節點。若必須在相同的機器上執行不同 DRU 的節點，請確定機器可解決所有單點故障的問題 (與磁碟、記憶體、CPU、電源、作業系統當機...等等相關的故障)。

---

## 負載平衡器配置

兩個拓樸在叢集中皆有應用程式伺服器實例。這些實例會將階段作業資訊保留至 HADB。配置負載平衡器，以納入叢集中所有應用程式伺服器實例的配置資訊。

如需有關設定叢集以及將應用程式伺服器實例增加至叢集的更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 6 章「使用 Application Server 叢集」。

## 並置拓樸

在並置拓樸中，應用程式伺服器實例與 HADB 節點皆位於相同的機器上 (因此稱為**並置**)。此拓樸需要的機器數比個別層級拓樸少。並置拓樸可更有效率地使用 CPU—應用程式伺服器實例與 HADB 節點共用一部機器，且在兩者之間平均分散處理。

此拓樸至少需要兩部機器。若要改善流量，請增加更多成對的機器。

---

**備註** - 並置拓樸適用於大型的對稱多重處理 (SMP) 機器，因為您可充份利用這些機器的處理能力。

---

## 配置範例

下圖描述並置拓樸的配置範例。



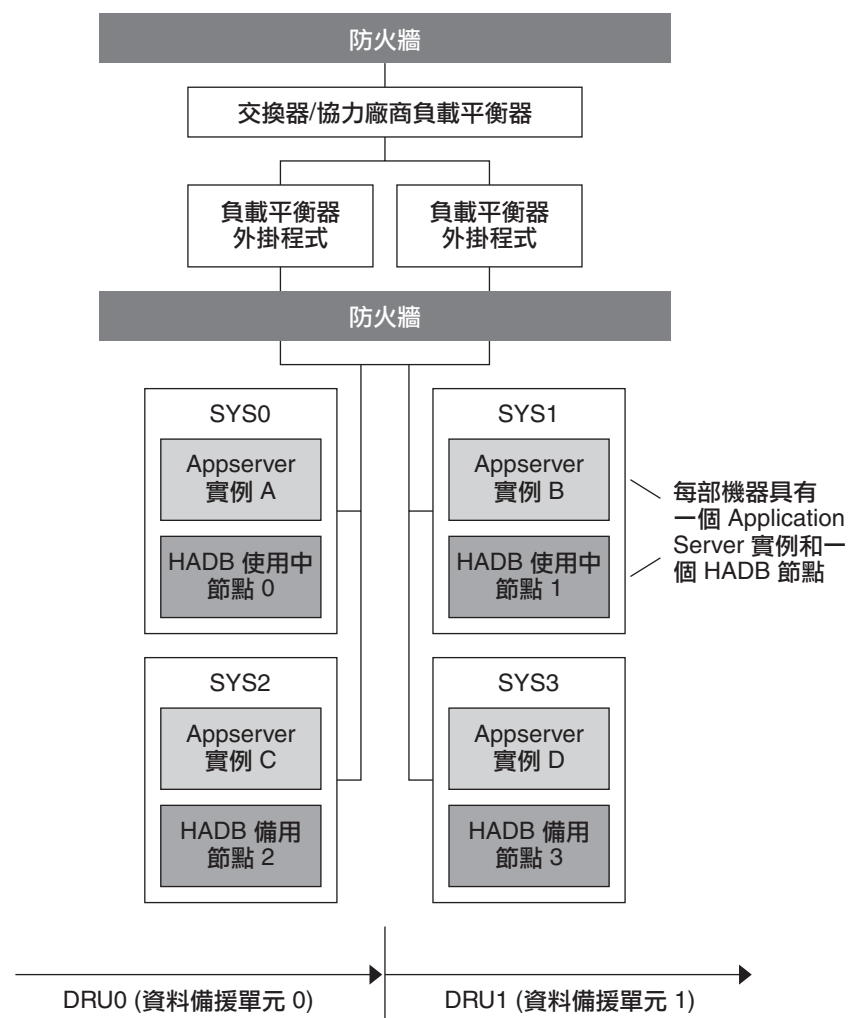


圖 3-1 並置拓樸範例

代管應用程式伺服器實例 A 的機器 SYS0、代管應用程式伺服器實例 B 的機器 SYS1、代管應用程式伺服器實例 C 的機器 SYS2，以及代管應用程式伺服器實例 D 的機器 SYS3。

這四個實例會組成叢集，並保留資訊至兩個 DRU：

- **DRU0** 包含兩部機器 SYS0 與 SYS2。HADB 使用中節點 0 位於機器 SYS0 上。HADB 備用節點 2 位於機器 SYS2 上。
- **DRU1** 包含兩部機器 SYS1 與 SYS3。HADB 使用中節點 1 位於機器 SYS1 上。HADB 備用節點 3 位於機器 SYS3 上。



## 並置拓樸的變體

如需最佳的延展性與流量，請透過增加更多機器來增加應用程式伺服器實例與 HADB 節點的數目。例如，您可以增加兩部機器，各有一個應用程式伺服器實例與一個 HADB 節點。請務必增加成對的 HADB 節點，並為每個 DRU 指定一個節點。[第 57 頁的「並置拓樸的變體」](#)說明此配置。

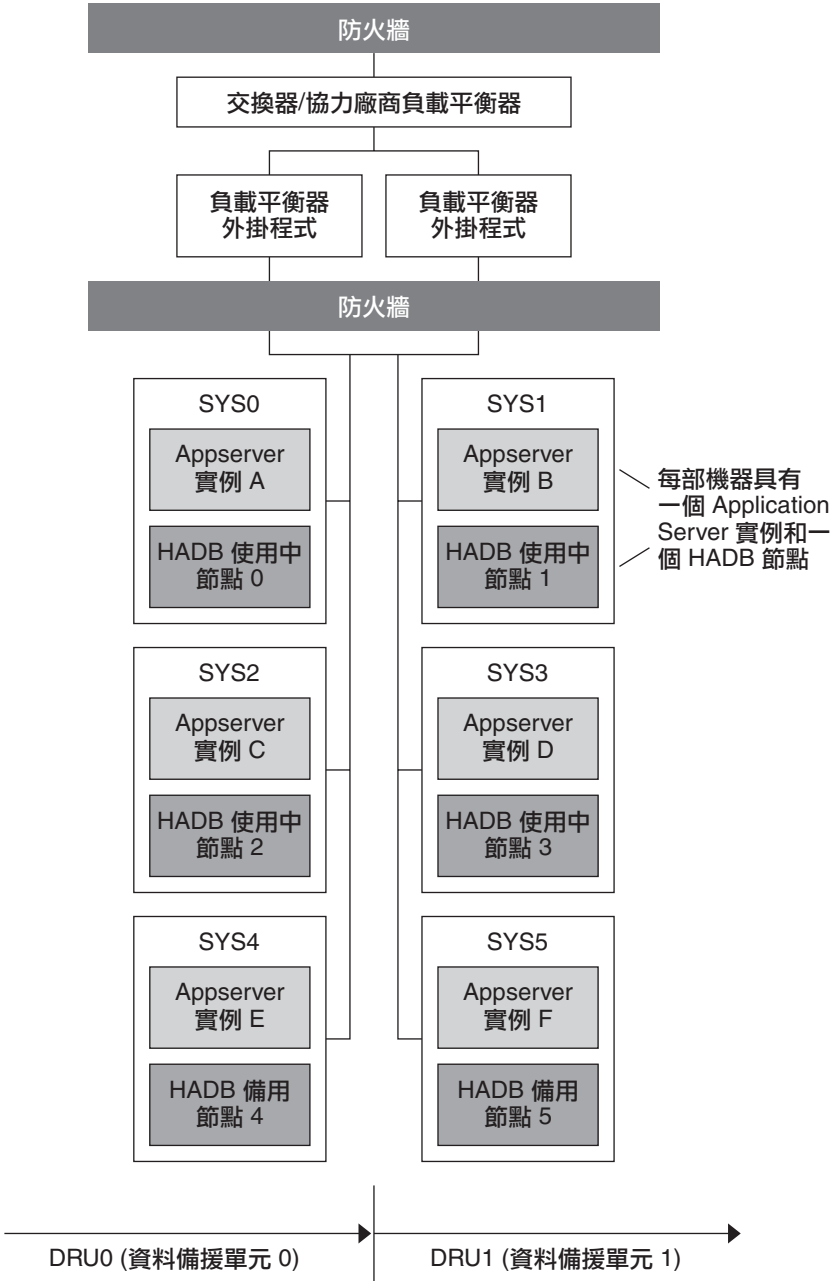


圖 3-2 並置拓模的變體

在此變體中，機器 SYS4 與 SYS5 已如第 55 頁的「配置範例」中所述增加至並置拓模。

應用程式伺服器實例使用下列方式代管：

- 機器 SYS0 代管實例 A
- 機器 SYS1 代管實例 B
- 機器 SYS2 代管實例 C
- 機器 SYS3 代管實例 D
- 機器 SYS4 代管實例 E
- 機器 SYS5 代管實例 F

這些實例會組成叢集，並保留資訊至兩個 DRU：

- **DRU0** 包含機器 SYS0、SYS2 與 SYS4。HADB 使用中節點 0 位於機器 SYS0 上。HADB 使用中節點 2 位於機器 SYS2 上。HADB 備用節點 4 位於機器 SYS4 上。
- **DRU1** 包含機器 SYS1、SYS3 與 SYS5。HADB 使用中節點 1 位於機器 SYS1 上。HADB 使用中節點 3 位於機器 SYS3 上。HADB 備用節點 5 位於機器 SYS5 上。

## 個別層級拓樸

在此拓樸中，應用程式伺服器實例與 HADB 節點位於不同的機器上 (因此稱為**個別層級**)。

此拓樸需要比並置拓樸更多的硬體。若有不同類型的機器，可能適合使用此拓樸—您可以配置一組機器代管應用程式伺服器實例，另一組代管 HADB 節點。例如，應用程式伺服器實例可使用較強大的機器，而 HADB 可使用較不強大的機器。

## 配置範例

下圖描述個別層級拓樸。

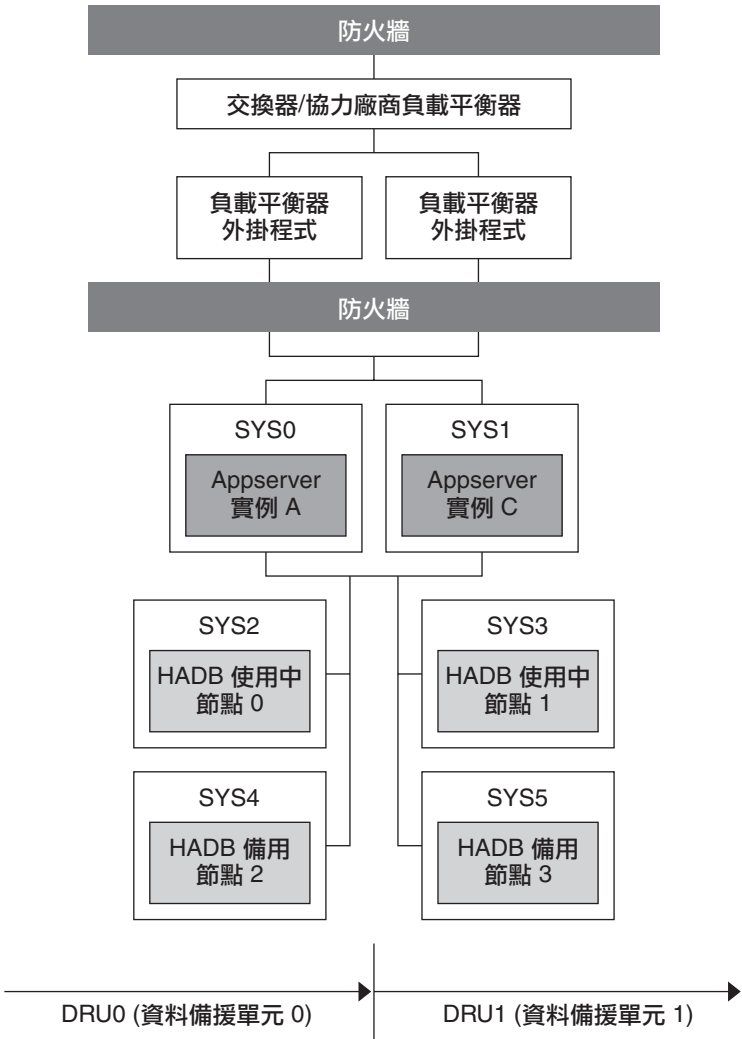


圖 3-3 個別層級拓樸範例

在此拓樸中，機器 SYS0 代管應用程式伺服器實例 A，而機器 SYS1 代管應用程式伺服器實例 B。這兩個實例組成一個叢集，將階段作業資訊保留至兩個 DRU：

- **DRU0** 包含兩部機器 SYS2 與 SYS4。HADB 使用中節點 0 位於機器 SYS2 上，而 HADB 備用節點 2 位於機器 SYS4 上。
- **DRU1** 包含兩部機器 SYS3 與 SYS5。HADB 使用中節點 1 位於機器 SYS3 上，而 HADB 備用節點 3 位於機器 SYS5 上。

DRU 上的所有節點位於不同的機器上，因此即使一部機器故障，仍可繼續在其他機器上使用任何 DRU 的完整資料。

## 個別層級拓樸的變體

個別層級拓樸的變體是透過將更多機器水平增加至配置，以增加應用程式伺服器實例的數目。例如，透過建立新的應用程式伺服器實例，將其他機器增加至配置範例。同理，透過增加更多機器代管 HADB 節點，以增加 HADB 節點的數目。如上所述，您必須成對增加 HADB 節點，且每個 DRU 一個節點。

[第 61 頁的「個別層級拓樸的變體」](#)說明此配置。

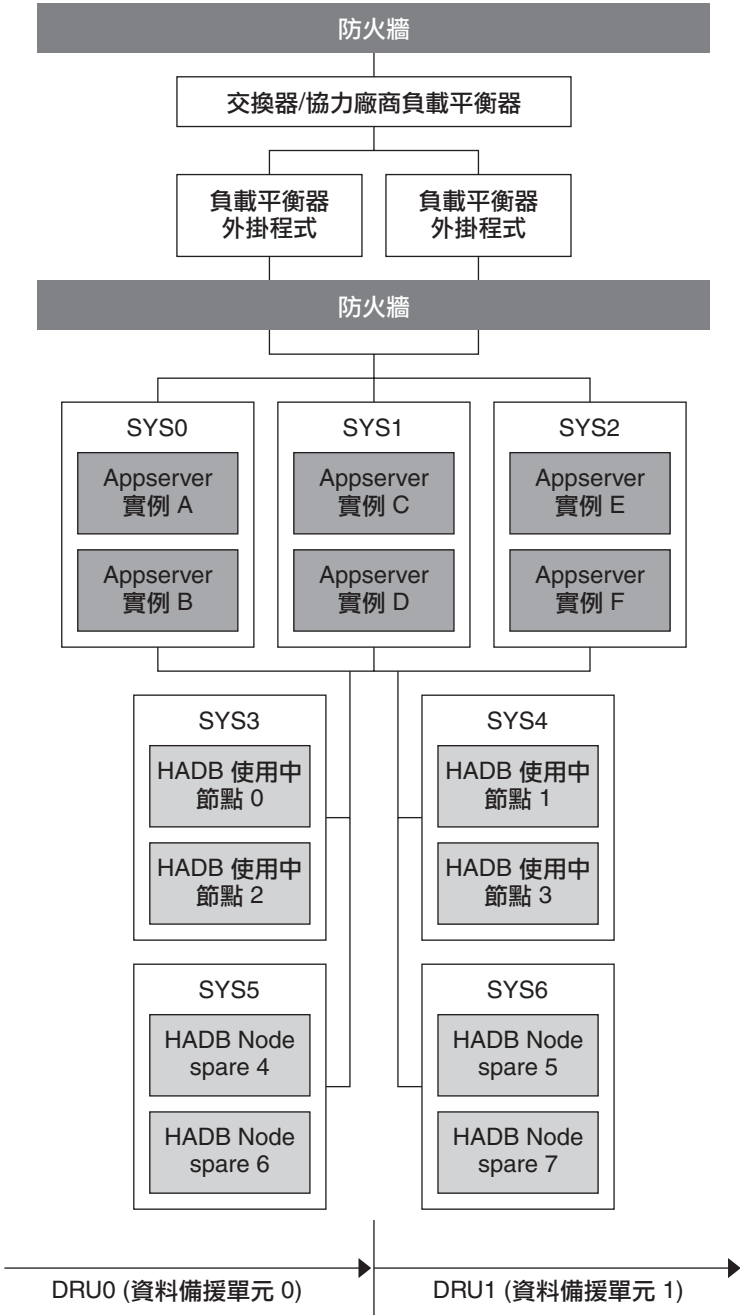


圖 3-4 個別層級拓樸的變體

在此配置中，每部代管應用程式伺服器實例的機器有兩個實例。因此叢集中共有六個應用程式伺服器實例。

HADB 節點位於機器 SYS3、SYS4、SYS5 與 SYS6 上。

DRU0 包含兩部機器：

- SYS3，代管 HADB 使用中節點 0 與 HADB 使用中節點 2。
- SYS5，代管 HADB 備用節點 4 與 HADB 備用節點 6。

DRU1 包含兩部機器：

- SYS4，代管 HADB 使用中節點 1 與 HADB 使用中節點 3。
- SYS6，代管 HADB 備用節點 5 與 HADB 備用節點 7。

每部代管 HADB 節點的機器會代管兩個節點。因此，共有八個 HADB 節點：四個使用中節點與四個備用節點。

## 決定要使用的拓樸

若要決定哪種拓樸 (或變體) 最適合您的效能與可用性需求，請測試拓樸並試驗不同的機器與 CPU 組合。

決定符合目標所需的代價。例如，若重點是要能輕鬆維護，則個別層級拓樸比較適合。而代價是此拓樸需要比並置拓樸更多的機器。

選擇拓樸的一項重要因素是可用的機器類型。若系統包含大型對稱多重處理 (SMP) 機器，則並置拓樸比較適合，因為您可以充份利用這些機器的處理能力。若系統包含各種機器類型，則個別層級拓樸更有用，因為您可以為 Application Server 層和 HADB 層配置不同的機器組合。例如，Application Server 層可能需要使用最強大的機器，而 HADB 層可使用較不強大的機器。

## 拓樸的比較

下表比較並置拓樸與個別層級拓樸。左欄列出拓樸的名稱，中間欄列出拓樸的優點，而右欄列出拓樸的缺點。

表 3-1 拓樸的比較

拓樸	優點	缺點
並置拓樸	<p>需要較少機器。由於 HADB 節點與應用程式伺服器實例皆位於相同的層級，因此您可以在每個備用節點上建立應用程式伺服器實例，以處理額外的負載。</p> <p>改善 CPU 使用率。共用一部機器的應用程式伺服器實例與 HADB 節點之間，會平均分散處理。</p> <p>適用於大型對稱多重處理 (SMP) 機器，因其可充份利用這些處理能力。</p>	<p>增加維護的複雜度。例如，當您必須關閉代管 HADB 節點的機器以執行維護時，機器上的應用程式伺服器實例也將無法使用。</p>
個別層級拓樸	<p>較容易維護。例如，您可以在代管應用程式伺服器實例的機器上執行維護，而無須關閉 HADB 節點。</p> <p>在使用不同的機器類型時很有用。您可以對 Application Server 層與 HADB 層配置不同的機器組合。例如，Application Server 層可以使用更強大的機器，而 HADB 層可以使用較不強大的機器。</p>	<p>需要比並置拓樸更多的機器。由於應用程式伺服器實例與 HADB 節點位於個別層級上，所以應用程式伺服器實例無法位於代管 HADB 備用節點的機器上。</p> <p>降低 CPU 使用率。Application Server 層與 HADB 層的負載可能會不平衡。這在機器數目很少 (四部至六部) 時會格外明顯。</p>



## 部署的檢核清單

---

本附錄提供檢核清單，讓使用者開始評估 Application Server 以及用於生產。

### 部署的檢核清單

表 4-1 檢核清單

元件/功能	說明
應用程式	<p>決定部署應用程式所需符合的下列需求。</p> <ul style="list-style-type: none"><li>■ 必要/可接受的回應時間。</li><li>■ 尖峰負載特性。</li><li>■ 必要的持續性範圍與頻率。</li><li>■ web.xml 中的階段作業逾時。</li><li>■ 容錯移轉與可用性需求。 如需更多資訊，請參閱「Sun Java System Application Server 9.1 Performance Tuning Guide」。</li></ul>
硬體	<ul style="list-style-type: none"><li>■ 使用相同的硬體類型代管 HADB 節點。</li><li>■ 已安裝必要的硬碟空間量與記憶體。</li><li>■ 使用調整大小作業找出部署需求。 如需更多資訊，請參閱「Sun Java System Application Server 9.1 版本說明」。</li></ul>
作業系統	<ul style="list-style-type: none"><li>■ 務必將產品安裝於支援的平台上。</li><li>■ 確定修補程式層級是最新且正確的。 如需更多資訊，請參閱「Sun Java System Application Server 9.1 版本說明」。</li></ul>

表 4-1 檢核清單 (續)

元件/功能	說明
網路基礎架構	<ul style="list-style-type: none"> <li>■ 找出單點故障並加以解決。</li> <li>■ 務必正確配置 NIC 與其他網路元件。</li> <li>■ 執行 <code>ttcp</code> 基準測試以判斷流量是否符合需求/預期的結果。</li> <li>■ 根據您的喜好來設定 <code>rsh/ssh</code>，以正確安裝 HADB 節點。 如需更多資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」。</li> </ul>
後端與其他外部資料來源	向網域專家或廠商確認，以確定可適當地配置這些資料來源。
系統變更/配置	<ul style="list-style-type: none"> <li>■ 務必先完成對 <code>/etc/system</code> 與其在 Linux 上對應項目的變更，再執行任何效能/壓力測試。</li> <li>■ 務必完成對 TCP/IP 設定的變更。</li> <li>■ 依預設，系統隨附許多預先配置的服務。並非所有服務皆須執行。請關閉不需要的服務以節省系統資源。</li> <li>■ 在 Solaris 上，使用 <code>Setoolkit</code> 決定系統的運作方式。解決所有顯示的旗標。 如需更多資訊，請參閱「Sun Java System Application Server 9.1 Performance Tuning Guide」。</li> </ul>
Application Server 與 HADB 安裝	<ul style="list-style-type: none"> <li>■ 確定 NFS 掛載的磁碟區上未安裝這些伺服器。</li> <li>■ 在相同機器上安裝 Application Server 與 HADB 節點時，請檢查是否有足夠的磁碟空間與 RAM。</li> <li>■ 在相同系統上安裝多個 HADB 節點時，請檢查是否有足夠的獨立磁碟。 如需更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 2 章「安裝和設定高可用性資料庫」。</li> </ul>
HADB 配置	<ul style="list-style-type: none"> <li>■ 設定 HADB Data Device 的大小。</li> <li>■ 定義 <code>DataBufferPoolSize</code>。</li> <li>■ 定義 <code>LogBufferSize</code>。</li> <li>■ 定義 <code>InternalBufferSize</code>。</li> <li>■ 設定 <code>NumberOfLocks</code>。</li> <li>■ 設定各種 Application Server 元件的最佳逾時值。</li> <li>■ 在檔案系統上建立 HADB 節點的實體配置。 如需更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的「配置 HADB」。</li> </ul>

表 4-1 檢核清單 (續)

元件/功能	說明
<b>Application Server 配置</b>	<ul style="list-style-type: none"> <li>■ 記錄：啟用自動重建存取記錄。</li> <li>■ 選擇適當的記錄層級。通常會使用「警告」。</li> <li>■ 使用管理主控台配置 J2EE 容器。</li> <li>■ 使用管理主控台配置 HTTP 偵聽程式。</li> <li>■ 使用管理主控台配置 ORB 執行緒池。</li> <li>■ 若使用與原生碼相關的 Type2 驅動程式或呼叫，請務必於 LD_LIBRARY_PATH 中指定 mtmalloc.so。</li> <li>■ 務必使用適當的持續性範圍與頻率，且未在個別 Web/EJB 模組中遭到覆寫。</li> <li>■ 確定僅會針對 SFSB 中的重要方法進行檢查點檢查。 如需有關調校的更多資訊，請參閱「Sun Java System Application Server 9.1 Performance Tuning Guide」。 如需有關配置的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」。</li> </ul>
<b>負載平衡器配置</b>	<ul style="list-style-type: none"> <li>■ 確定已安裝 Web 伺服器。</li> <li>■ 確定已安裝 Web Server 的負載平衡器外掛程式。</li> <li>■ 確定已停用修補程式檢查。</li> <li>■ 調低 KeepAliveQuery 參數的值。值愈低，負載低的系統延時愈低。值愈高，負載高的系統流量愈高。 如需更多資訊，請參閱「Sun Java System Application Server 9.1 Performance Tuning Guide」中的「Keep Alive」。</li> </ul>
<b>Java 虛擬機器配置</b>	<ul style="list-style-type: none"> <li>■ 一開始設定的最小與最大堆疊大小須相同，且每個實例至少要有 1 GB。</li> <li>■ 如需更多資訊，請參閱 <a href="#">Java Hotspot VM Options</a>。</li> <li>■ 執行多個應用程式伺服器實例時，請考慮建立處理器組，並連結至 Application Server。這有助於使用 CMS 控制器清除舊版。</li> </ul>

表 4-1 檢核清單 (續)

元件/功能	說明
配置負載平衡器中的逾時	<ul style="list-style-type: none"><li>■ Response-time-out-in-seconds - 宣告應用程式伺服器實例運作不正常之前，負載平衡器所等候的時間。根據應用程式的回應時間設定此值。如果設定太高，Web 伺服器與負載平衡器外掛程式會等候很久，才會判定應用程式伺服器實例運作不正常。如果設定太低且 Application Server 的回應時間超過此臨界值，則會誤判實例運作不正常。</li><li>■ Interval-in-seconds - 以秒為單位的時間間隔，於此時間之後會檢查運作不正常的實例是否已回到正常狀態。值若太低會在負載平衡器外掛程式與應用程式伺服器實例之間產生多餘的流量；值若太高則會延遲請求路由至恢復正常實例的時間。</li><li>■ Timeout-in-seconds - 運作狀態檢查請求取得回應的持續時間。請根據叢集中系統間的流量調整此值，以確保運作狀態檢查成功。 如需更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的第 5 章「配置 HTTP 負載平衡」。</li></ul>

表 4-1 檢核清單 (續)

元件/功能	說明
配置 HADB 中的逾時	<ul style="list-style-type: none"> <li>■ <code>sql_client_timeout</code> - SQLSUB 等候閒置用戶端的時間。例如，已登入、傳送一些請求，然後等候使用者輸入的用戶端。閒置超過 30 分鐘的用戶端會視為停用，且會終止階段作業。設定太低的值會過早終止 SQL 階段作業。設定太高的值會導致未閒置但已結束的 SQL 階段作業佔用資源，進而使其他 SQL 用戶端無法登入。調校此變數時，另請考慮 <code>nsessions</code> 的設定。如果 HADB JDBC 連線池 <code>steady-pool-size</code> 大於 <code>max-pool-size</code>，則可將 <code>idle-timeout-in-seconds</code> 設定得比 <code>sql_client_timeout</code> 低，使 Application Server 本身在 HADB 關閉連線之前，就先關閉連線。預設值為 1800 秒。</li> <li>■ <code>lock_timeout</code> - 作業事件等候存取資料的最長時間 (以毫秒為單位)。超過此時間，作業事件會產生錯誤訊息：「The transaction timed out. (作業事件逾時。)」這類逾時是由於作業事件等候其他作業事件已保留的鎖定 (死結) 所致，並會導致伺服器大量負載。請勿將此值設定低於 500 毫秒。如果在伺服器記錄中看到「transaction timed out (作業事件逾時)」訊息，請加大此值。將特性如下增加至 HADB 的 JDBC 連線池，以設定鎖定逾時值：<code>&lt;property name=lockTimeout value="x"&gt;</code>。預設值為 5000 毫秒。</li> <li>■ <code>Querytimeout</code> - HADB 等候查詢執行的最長時間 (以毫秒為單位)。若伺服器記錄不斷表示查詢逾時的異常情況，請考慮加大此值。將下列特性增加至 HADB 的 JDBC 連線池，以設定此值：<code>&lt;property name=QueryTimeout value="x"&gt;</code>。預設值為 30 秒。</li> <li>■ <code>loginTimeout</code> - 用戶端等候登入 HADB 的最長時間 (以秒為單位)。將下列特性增加至 HADB 的 JDBC 連線池，以設定此值：<code>&lt;property name=loginTimeout value="x"&gt;</code>。預設值為 10 秒。</li> <li>■ <code>MaxTransIdle</code> - 作業事件傳送回覆給用戶端到收到下一個請求之間，可閒置的最長時間 (以毫秒為單位)。如下所示，您可將特性增加至 HADB 的 JDBC 連線池以變更此值：<code>&lt;property name=maxtransIdle value="x"&gt;</code>。預設值為 40 秒。 如需更多資訊，請參閱：「Sun Java System Application Server Performance Tuning Guide」。</li> </ul>
配置 Application Server 中的逾時	<ul style="list-style-type: none"> <li>■ <code>Max-wait-time-millis</code> - 丟出異常前，為了從池取得連線而等候的時間。預設值為 6 秒。若有高負載系統持續保留大於 50 KB 的資料，請考慮變更此值。</li> <li>■ <code>Cache-idle-timeout-in-seconds</code> - 在 EJB 鈍化之前，允許 EJB 於快取中間置的時間。僅會套用到實體 Bean 與有狀態的階段作業 Bean。</li> <li>■ <code>Removal-timeout-in-seconds</code> - EJB 維持鈍化 (在備份儲存庫中閒置) 的時間。預設值為 60 分鐘。請根據 SFSB 容錯移轉的需求調整此值。</li> </ul> <p>請根據 HADB 的 JDBC 連線池設定 <code>max-wait-time-in-millis</code>，調整全部的值。如需更多資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」中的「配置 JDBC 連線池」。</p>

表 4-1 檢核清單 (續)

元件/功能	說明
調校 VM 資源回收 (GC)	<p>資源回收暫停四秒或更長時間時，可能會導致在持續階段作業狀態至 HADB 時，發生斷斷續續的問題。若要避免此問題，請調校 VM 堆疊。若不接受保留資料有一絲一毫的錯誤，或系統尚未裝配完整，請使用 CMS 控制器或流量控制器。</p> <p>增加下列語法可啓用這些控制器：</p> <pre>&lt;jvm-options&gt;-XX:+UseConcMarkSweepGC&lt;/jvm-options&gt;</pre> <p>此選項可降低流量。</p>

# 索引

---

## A

Apache Web Server, 25  
asadmin 指令, 23

## D

DAS, 23  
Domain Administration Server (DAS), 23

## E

EJB 容器, 20

## H

HADB, 27-33, 43  
  主機, 47  
  系統需求, 28  
  架構, 28  
  故障回復, 31  
  負載, 39  
  節點, 28, 46, 54  
  管理用戶端, 32  
  管理代理程式, 32  
  管理系統, 31  
  管理網域, 32  
  網路配置, 43  
  網路瓶頸, 43  
  儲存庫, 33  
  儲存容量, 48

HTTP 階段作業, 25

## I

InitialContext, 26

## J

J2EE 服務, 20  
J2EE 連接器架構, 21  
Java 2 Enterprise Edition (J2EE), 19  
Java API for XML-based RPC (JAX-RPC), 20  
Java API for XML Registries (JAXR), 20  
Java Authorization Contract for Containers (JACC), 20  
Java Database Connectivity (JDBC), 21  
Java Message Service (JMS), 20, 21, 27, 48  
Java Naming and Directory Interface (JNDI), 20  
JavaMail API, 21

## M

Microsoft Internet Information Server, 25

## O

Object Request Broker (ORB), 20

**S**

Sun Java System Message Queue, 27, 48  
Sun Java System Web Server, 25

**W**

Web Server, 25  
Web 服務, 20  
Web 服務互通性 (WS-I), 21  
Web 服務描述語言 (WSDL), 20  
Web 容器, 20  
乙太網路卡, 43  
已命名的配置, 24  
子網路, 42  
元件, 22  
可用性, 43  
    資料備援單元的, 54  
    與備援, 44  
    與叢集, 44  
用戶端, 20  
    與 JMS, 52  
目前的使用者, 36  
主機, HADB, 47  
代理程式叢集, 49, 52  
本機磁碟儲存體, 54  
回應時間, 37  
安全性, 20  
尖峰負載, 42  
共用拓樸需求, 53-55  
同步運作使用者, 36  
考慮時間, 37  
有狀態的階段作業 Bean, 25, 40-41  
作業事件, 20  
伺服器  
    元件, 22  
    服務, 20  
    負載, 36, 42  
    效能, 35  
    容器, 20  
    節點代理程式, 24  
    網域管理, 23  
    網路配置, 41  
    實例, 22, 46  
    叢集, 23

伺服器的實例, 46  
伺服器實例, 22  
每分鐘請求數, 38  
命名, 20  
拓樸  
    比較, 63  
    共同需求, 53-55  
    並置, 53, 55-59  
    負載平衡, 54  
    個別層級, 42, 53, 59-63  
    基本結構, 53  
    選取, 53-64  
拓樸的比較, 63  
使用中節點, 29  
使用者, 同步運作, 36  
使用者資料封協定 (UDP), 42  
並置拓樸, 53, 55-59  
    使用對稱多重處理機器, 55  
    標準配置, 55-56  
    變體, 57-59  
流量, 36  
故障  
    類別, 44-45  
    類型, 44  
持續性, 階段作業, 25  
持續性範圍, 40  
持續性頻率, 39  
負載  
    HADB, 39  
    伺服器, 36, 42  
負載平衡  
    HTTP, 25  
    IIOP, 26  
    與拓樸, 54  
訊息代理程式, 49  
訊息驅動 Bean, 27  
連接器, 21  
配置, 24  
    預設, 24  
效能, 35  
高可用性資料庫 (HADB), 27-33  
個別層級拓樸, 42, 53, 59-63  
    參考配置, 59-61  
    變體, 61-63



- 容量, 使用備用機器維持, 45
- 容器, 20
- 容錯, 44
- 容錯移轉容量, 規劃, 45
- 設計決策, 45
- 階段作業
  - HTTP, 25
  - 大小, 39
  - 有狀態的階段作業 Bean, 40-41
  - 持續性, 25, 39
  - 持續性範圍, 40
  - 持續性頻率, 39
- 基本結構, 拓樸, 53
- 備用節點, 29, 30, 45
- 備用機器, 維持容量, 45
- 備援, 44-45, 54
- 運作狀態檢查程式, 44
- 資料備援單元, 29-30
  - 改善可用性, 45
  - 電源供應, 54
  - 確保可用性, 54
  - 機器數目, 54
- 資源, 21
- 資源配接卡, 21
- 路由器, 42
- 遠端瀏覽器模擬器, 35
- 節點, 54
- 節點, HADB, 28, 46
- 節點代理程式, 24
- 預設 JMS 主機, 51
- 預設伺服器, 23
- 預設配置, 24
- 預設部署, 52
- 管理主控台, 23
- 管理網域, 22
- 網域, 22
- 網路卡, 43
- 網路配置
  - HADB, 43
  - 伺服器, 41
- 對稱多重處理機器, 供並置拓樸使用, 55
- 部署規劃, 35
  - 方案範例, 52
  - 檢核清單, 65
- 調整大小, 系統, 46-48
- 機器
  - 使用備用機器維持容量, 45
  - 資料備援單元中, 54
- 頻寬, 41
- 應用程式, 19
- 檢查點檢查, 40-41
- 檢核清單, 65
- 叢集, 23
  - Message Queue, 49, 52
  - 與可用性, 44
- 簡易郵件傳輸協定 (SMTP), 21
- 鏡像節點, 29
- 鏡像機器, 45
- 類型, 故障, 44

