



Sun Java Enterprise System 2005Q4 배포 계획 설명서

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

부품 번호: 819-3452
2005년 10월

Copyright 2005 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 모든 권리는 저작권자의 소유입니다.

이 제품 또는 문서는 저작권법의 보호를 받으며 제품의 사용, 복사, 배포 및 디컴파일을 제한하는 사용권에 따라 배포됩니다. 이 제품의 어떠한 부분도 Sun과 해당하는 모든 사용권 보유자의 사전 서면 승인 없이 어떠한 형태와 수단으로도 복제될 수 없습니다. 글꼴 기술을 비롯한 타사 소프트웨어는 저작권법의 보호를 받으며 Sun 공급업체에서 사용이 허가됩니다.

제품 중에는 캘리포니아 대학에서 허가한 Berkeley BSD 시스템에서 파생된 부분이 포함되어 있을 수 있습니다. UNIX는 미국 및 다른 국가에서 X/Open Company, Ltd.를 통해 독점적으로 사용권이 부여되는 등록 상표입니다.

Sun, Sun Microsystems, Sun 로고, docs.sun.com, AnswerBook, AnswerBook2, Java, 및 Solaris는 미국 및 다른 국가에서 Sun Microsystems, Inc.의 상표 또는 등록 상표입니다. 모든 SPARC상표는 라이선스 하에 사용되며 미국 및 다른 국가에서 SPARC International, Inc.의 상표 또는 등록 상표입니다. SPARC 상표를 사용하는 제품은 Sun Microsystems, Inc.가 개발한 구조를 기반으로 하고 있습니다.

OPEN LOOK 및 Sun™ 그래픽 사용자 인터페이스는 Sun Microsystems, Inc.가 자사의 사용자 및 정식 사용자용으로 개발했습니다. Sun은 컴퓨터 업계를 위한 시각적 또는 그래픽 사용자 인터페이스(GUI)의 개념을 연구 개발한 Xerox사의 선구적인 노력을 높이 평가하고 있습니다. Sun은 Xerox와 Xerox 그래픽 사용자 인터페이스(GUI)에 대한 비독점적 사용권을 보유하고 있습니다. 이 사용권은 OPEN LOOK GUI를 구현하는 Sun의 정식 사용자에게도 적용되며 그렇지 않은 경우에는 Sun의 서면 사용권 계약을 준수해야 합니다.

미국 정부의 권리 - 상용 소프트웨어. 정부 사용자는 Sun Microsystems, Inc. 표준 사용권 조항 및 해당 FAR 규정 및 보충 규정을 준수해야 합니다.

설명서는 "있는 그대로" 제공되며 법률을 위반하지 않는 범위 내에서 상품성, 특정 목적에 대한 적합성 또는 비침해에 대한 묵시적인 보증을 포함하여 모든 명시적 또는 묵시적 조건 표현 및 보증을 배제합니다.



051216@13215



목차

머리말 11

1 배포 계획 소개 17

Java Enterprise System 정보 17

시스템 서비스 17

기본 제공 서비스 및 사용자 정의 개발 서비스 19

Java Enterprise System으로 마이그레이션 19

배포 계획 정보 20

솔루션 라이프 사이클 20

비즈니스 분석 단계 22

기술 요구 사항 단계 22

논리적 설계 단계 22

배포 설계 단계 23

구현 단계 23

운영 단계 24

2 비즈니스 분석 25

비즈니스 분석 정보 25

비즈니스 요구 사항 정의 26

비즈니스 목표 설정 26

사용자 요구 이해 27

기업 문화 이해 28

증분 방법 사용 29

서비스 수준 계약 이해 30

비즈니스 제약 조건 정의 30

마이그레이션 문제 30

	일정 요구 사항	30
	예산 제한	31
	소유 비용	31
3	기술 요구 사항	33
	기술 요구 사항 정보	33
	사용 분석	34
	사용 사례	35
	서비스 품질 요구 사항 적용	36
	성능	37
	가용성	37
	가용성	39
	보안 요구 사항	40
	잠재 용량	41
	서비스 가능성 요구 사항	42
	서비스 수준 요구 사항	43
4	논리적 설계	45
	논리적 구조 정보	45
	논리적 구조 설계	46
	Java Enterprise System 구성 요소	47
	구성 요소 종속성	48
	웹 컨테이너 지원	51
	Messaging Server에서 제공되는 논리적으로 별개인 서비스	51
	액세스 구성 요소	52
	다중 계층 구조 설계	52
	논리적 구조 예	54
	Messaging Server 예	54
	신원 기반 통신 예	59
	액세스 영역	62
	배포 시나리오	64
5	배포 설계	65
	배포 설계 정보	65
	프로젝트 승인	66
	배포 설계 출력	66
	배포 설계에 영향을 미치는 요소	67

배포 설계 방법	68
프로세서 요구 사항 예상	68
프로세서 요구 사항 예상 예	69
보안 트랜잭션을 위한 프로세서 요구 사항 예상	73
보안 트랜잭션의 CPU 예상 개수	74
SSL 트랜잭션을 처리하는 특수 하드웨어	75
가용성 전략 결정	76
가용성 전략	76
가용성 설계 예	79
확장성에 대한 전략 결정	83
잠재 용량	83
확장성 예	84
성능 병목 현상 식별	85
디스크 액세스 최적화	86
최적의 자원 사용을 위한 설계	87
위기 관리	88
예제 배포 구조	89
6 배포 설계 구현	91
배포 설계 구현 정보	91
소프트웨어 설치 및 구성	92
파일럿 및 프로토타입 개발	92
파일럿 및 프로토타입 배포 테스트	93
프로덕션 배포 실행	94
색인	95

표

표 1-1	Java Enterprise System 서비스 범주	18
표 3-1	사용 분석 요소	34
표 3-2	서비스 품질 요구사항에 영향을 미치는 시스템 품질	36
표 3-3	1년 내내(8,760시간) 실행되는 시스템의 예정되지 않은 중단 시간	38
표 3-4	우선 순위별 서비스 가용성	38
표 3-5	확장성 요소	40
표 3-6	서비스 가능성 요구 사항 항목	42
표 4-1	Java Enterprise System 구성 요소 종속성	48
표 4-2	Messaging Server 구성	51
표 4-3	원격 액세스를 제공하는 Java Enterprise System 구성 요소	52
표 4-4	다중 계층 구조의 논리적 계층	53
표 4-5	Messaging Server 논리적 구조의 구성 요소	55
표 4-6	보안 액세스 영역 및 영역 내에 있는 구성 요소	63
표 5-1	사용자 시작점 액세스를 포함하는 구성 요소에 대한 CPU 예상 개수	71
표 5-2	지원 구성 요소에 대한 CPU 예상 개수	71
표 5-3	최고 부하를 위해 CPU 예상 개수 조정	72
표 5-4	구성 요소 지원을 위한 CPU 예상 개수 조정	73
표 5-5	보안 트랜잭션에 대한 CPU 예상 개수 수정	74
표 5-6	구성 요소 지원을 위한 CPU 예상 개수 조정	79
표 5-7	데이터 액세스 지점	86
표 5-8	자원 관리 고려 사항	88

그림

그림 1-1	솔루션 라이프 사이클	21
그림 4-1	Java Enterprise System 구성 요소	47
그림 4-2	Java Enterprise System 구성 요소 종속성	50
그림 4-3	다중 계층 구조 모델	53
그림 4-4	Messaging Server 배포의 논리적 구조	55
그림 4-5	액세스 영역에 있는 논리적 구조	63
그림 5-1	신원 기반 통신 시나리오의 논리적 구조	70
그림 5-2	단일 서버 시스템	77
그림 5-3	두 서버를 가진 N+1 페일오버 시스템	77
그림 5-4	두 서버 간 로드 균형 조정 및 페일오버	78
그림 5-5	n 개의 서버 간 로드 배포	78
그림 5-6	Sun Cluster 소프트웨어를 사용하는 페일오버 예	81
그림 5-7	단일 마스터 복제 예	82
그림 5-8	다중 마스터 복제 예	83
그림 5-9	수평 및 수직 확장 예	85
그림 5-10	예제 배포 구조	90

머리말

Sun Java Enterprise System 2005Q4 배포 계획 설명서는 Sun Java™ Enterprise System 기반 엔터프라이즈 배포 솔루션의 계획 및 설계를 소개합니다. 이 설명서는 배포 계획 및 설계에 대한 기본 개념 및 원칙을 소개하고 배포 설계 프로젝트의 단계 및 작업을 캡슐화하는 솔루션 라이프사이클을 설명하며 Java Enterprise System(Java ES)으로 엔터프라이즈 전반의 배포 솔루션을 계획할 경우 사용할 수 있는 높은 수준의 예외 전략을 제공합니다.

대상

이 설명서는 주로 엔터프라이즈 배포의 분석 및 설계 책임이 있는 배포 설계자 및 비즈니스 계획자를 위한 것입니다. 이 설명서는 또한 시스템 통합자 및 엔터프라이즈 응용 프로그램의 다양한 요소를 설계하고 구현할 책임자에게도 유용합니다.

이 설명서를 읽기 전에

이 설명서는 사용자가 엔터프라이즈 수준 응용 프로그램의 설계 및 설치에 익숙하며 **Sun Java Enterprise System 2005Q4 기술 개요**를 읽었다고 가정합니다.

본 설명서의 구성

이 설명서는 배포 계획의 다양한 단계를 설명하는 솔루션 라이프 사이클을 기반으로 합니다. 1 장에서 솔루션 라이프 사이클을 설명합니다.

Java ES 설명서 세트

Java ES 설명서 세트에서는 배포 계획 및 시스템 설치를 설명합니다. 시스템 설명서의 URL은 <http://docs.sun.com/coll/1286.1> 및 <http://docs.sun.com/coll/1397.1>입니다. Java ES 소개는 다음 표에 나열된 설명서를 표시된 순서대로 참조하십시오.

표 P-1 Java Enterprise System 설명서

설명서 제목	목적
Sun Java Enterprise System 2005Q4 릴리스 노트	알려진 문제점을 포함하여 Java ES에 대한 최신 정보를 제공합니다. 또한, 구성 요소마다 릴리스 노트가 있습니다.
Sun Java Enterprise System 2005Q4 설명서 로드맵	시스템과 개별 구성 요소 모두에 대하여 Java ES와 연관된 모든 설명서의 정보를 제공합니다.
Sun Java Enterprise System 2005Q4 기술 개요	Java ES의 기술적 및 개념적 기초를 소개합니다. 구성 요소, 구조, 프로세스 및 기능에 대해 설명합니다.
Sun Java Enterprise System 2005Q4 배포 계획 설명서	Java ES를 기반으로 하는 엔터프라이즈 배포 솔루션의 계획 및 설계를 소개합니다. 배포 계획 및 설계의 기본 개념 및 원칙을 소개하고 솔루션 라이프사이클을 설명하며 Java ES 기반 솔루션 계획 시 사용할 수 있는 고급 예 및 전략을 제공합니다.
Sun Java Enterprise System 2005Q4 설치 계획 설명서	Java ES 배포의 하드웨어, 운영 체제, 네트워크 면에서 구현 사양을 개발하는 데 도움이 됩니다. 구성 요소 종속성 등 설치 및 구성 계획 시 고려해야 할 문제에 대해 설명합니다.
Sun Java Enterprise System 2005Q4 UNIX용 설치 설명서	Solaris Operating System 또는 Linux 운영 체제에 Java ES를 설치하는 과정 전체를 설명합니다. 설치 후에 구성 요소를 구성하고 제대로 작동하는지 확인하는 방법도 보여줍니다.

표 P-1 Java Enterprise System 설명서 (계속)

설명서 제목	목적
Sun Java Enterprise System 2005Q4 설치 참조 설명서	구성 매개 변수에 대한 추가 정보 및 구성 계획에 사용하는 워크시트를 제공하며, 기본 디렉토리 및 포트 번호와 같은 참조 자료를 나열합니다.
Sun Java Enterprise System 2005Q1 배포 예 시리즈: 평가 시나리오	한 시스템에 Java ES을 설치하고 핵심 공유 네트워크 서비스를 구축하며 구축한 서비스에 액세스할 수 있도록 사용자 계정을 설정하는 방법을 설명합니다.
Sun Java Enterprise System 2005Q4 업그레이드 설명서	Solaris Operating System 또는 Linux 운영 환경의 Java ES 업그레이드 지침을 제공합니다.
Java Enterprise System 용어집	Java ES 설명서에서 사용되는 용어를 정의합니다.

활자체 규약

다음 표에는 이 설명서에 사용된 활자체 규칙 변경 사항이 나와 있습니다.

표 P-2 활자체 규약

서체	의미	예
AaBbCc123	명령, 파일, 디렉토리 이름과 화면 상의 컴퓨터 출력	.login 파일을 편집합니다. 모든 파일을 나열하려면 ls -a를 사용합니다. machine_name% you have mail.
AaBbCc123	화면 상의 컴퓨터 출력과 반대로 사용자가 직접 입력하는 내용입니다.	machine_name% su Password:
AaBbCc123	실제 이름 또는 값으로 대체되는 자리 표시자입니다.	파일을 제거하려면 rm filename을 입력하십시오.
AaBbCc123	책 제목, 새로 나오는 용어, 강조 표시할 단어입니다. 단, 온라인에서는 일부 강조되는 항목이 굵게 표시됩니다.	사용자 설명서 의 6장을 참조하십시오. 캐시는 로컬로 저장된 복사본입니다. 파일을 저장하지 마십시오 .

명령 예의 셸 프롬프트

다음 표는 기본 시스템 프롬프트와 슈퍼유저 프롬프트를 보여줍니다.

표 P-3 셸 프롬프트

셸	프롬프트
UNIX 및 Linux 시스템의 C 셸	machine_name%
UNIX 및 Linux 시스템의 C 셸 슈퍼유저	machine_name#
UNIX 및 Linux 시스템의 Bourne 셸 및 Korn 셸	\$
UNIX 및 Linux 시스템의 Bourne 셸 및 Korn 셸 슈퍼유저	#
Microsoft Windows 명령줄	C:\

기호 규칙

다음 표에서는 이 설명서에 사용된 기호를 설명합니다.

표 P-4 기호 규칙

기호	설명	예	의미
[]	선택 인수 및 명령 옵션을 포함합니다.	ls [-l]	-l 옵션은 선택 옵션입니다.
{ }	선택할 수 있는 필수 명령 옵션 집합이 포함됩니다.	-d {y n}	-d 옵션에는 y 인수 또는 n 인수를 사용해야 합니다.
\${ }	변수 참조를 나타냅니다.	\${com.sun.javaRoot}	com.sun.javaRoot 변수 값을 참조합니다.
-	동시에 눌러야 하는 여러 키를 결합합니다.	Control-A	A 키를 누른 상태에서 Control 키를 누릅니다.
+	연속적으로 눌러야 하는 여러 키를 결합합니다.	Ctrl+A+N	Ctrl 키를 눌렀다가 놓은 다음 후속 키를 누릅니다.
→	그래픽 사용자 인터페이스에서 메뉴 항목 선택을 나타냅니다.	파일 → 새로 만들기 → 템플릿	파일 메뉴에서 새로 만들기를 선택합니다. 새로 만들기 하위 메뉴에서 템플릿을 선택합니다.

온라인 Sun 자원에 액세스

docs.sun.comSM 웹 사이트에서 Sun 기술 관련 설명서를 온라인으로 이용할 수 있습니다. docs.sun.com 아카이브를 찾아보거나 특정 책 제목 또는 주제를 검색할 수 있습니다. 설명서는 PDF 및 HTML 형식의 온라인 파일로 제공됩니다. 두 형식 모두 장애인자용 보조 기술로 판독 가능합니다.

다음의 Sun 자원에 액세스하려면 <http://www.sun.com>을 방문하십시오.

- Sun 제품 다운로드
- 서비스 및 솔루션
- 지원(패치 및 업데이트 포함)
- 교육
- 연구
- 커뮤니티(예: Sun 개발자 네트워크)

타사 웹 사이트 참조

이 문서에서 참조하는 타사 URL은 추가 관련 정보를 제공합니다.

주 - Sun은 이 설명서에 명시된 타사 웹 사이트의 가용성에 대해 책임지지 않습니다. 또한 해당 사이트나 리소스를 통해 제공되는 내용, 광고, 제품 및 기타 자료에 대해 어떠한 보증도 하지 않으며 그에 대한 책임도 지지 않습니다. Sun은 해당 사이트 또는 자원을 통해 사용 가능한 내용, 제품 또는 서비스의 사용과 관련해 발생하거나 발생했다고 간주되는 손해나 손실에 대해 책임이나 의무를 지지 않습니다.

사용자 의견

Sun은 설명서의 내용 개선에 노력을 기울이고 있으며, 여러분의 의견과 제안을 환영합니다. 의견을 나누려면 <http://docs.sun.com>으로 가서 의견 보내기를 누릅니다. 온라인 양식에 전체 설명서 제목과 부품 번호를 기입해 주시기 바랍니다. 부품 번호는 책 제목 페이지나 설명서 URL에 있는 7자리 또는 9자리 번호입니다. 예를 들어, 이 설명서의 부품 번호는 819-3452입니다. 사용자 의견을 제출할 때 해당 양식에 영문 설명서 제목과 부품 번호를 입력해야 할 수도 있습니다. 본 설명서의 영문 부품 번호와 제목은 819-2326, Sun Java Enterprise System 2005Q4 Deployment Planning Guide입니다.

배포 계획 소개

이 장에서는 Sun Java™ Enterprise System (Java ES)에 대한 간략한 개요를 제공하며 배포 계획 개념 설명 및 솔루션 라이프 사이클을 소개하고 엔터프라이즈 소프트웨어 시스템을 계획 및 설계하는 다양한 단계를 보여줍니다. 이 장의 내용은 다음과 같습니다.

- 17 페이지 “Java Enterprise System 정보”
- 20 페이지 “배포 계획 정보”

Java Enterprise System 정보

Java Enterprise System은 소프트웨어 인프라로서 네트워크 또는 인터넷 환경 전체에 분산된 엔터프라이즈 응용 프로그램을 지원하기 위한 완전한 미들웨어 서비스 세트를 제공합니다. 서비스를 제공하는 Java Enterprise System 구성 요소는 공통된 설치 프로그램을 사용하여 설치되며 공통된 공유 라이브러리 세트에서 동기화되고 통합된 사용자 아이디 및 보안 관리 시스템을 공유합니다.

시스템 서비스

Java Enterprise System 구성 요소가 제공하는 주 인프라 서비스는 다음과 같이 분류될 수 있습니다.

- **포털 서비스.** 이 서비스는 이동이 잦은 직원, 재택 근무자, 지식 근로자, 비즈니스 파트너, 공급자 및 고객이 회사 네트워크 외부의 아무 위치에서나 인터넷을 통해 개인화된 회사 포털에 안전하게 액세스할 수 있도록 지원합니다. 이 서비스는 언제 어디서나 액세스할 수 있는 기능을 사용자 커뮤니티에 제공하며 통합, 집계, 개인화, 보안, 모바일 액세스 및 검색 서비스를 포함합니다.
- **통신 및 공동 작업 서비스.** 이 서비스는 다양한 사용자 커뮤니티 간의 안전한 정보 교환을 지원하고 사용자 비즈니스 환경에서 메시징, 실시간 공동 작업, 달력, 일정 등의 기능을 제공합니다.

- **네트워크 아이디 및 보안 서비스.** 이 서비스는 모든 커뮤니티, 응용 프로그램 및 서비스에서 적절한 액세스 제어 정책이 전역적으로 시행되게 함으로써 주요 회사 정보 자산에 대한 보안과 보호 기능을 향상시켜 줍니다. 이러한 서비스는 아이디 프로필, 액세스 권한, 응용 프로그램 정보 및 네트워크 자원 정보를 저장하고 관리하기 위한 저장소에서 사용됩니다.
- **웹 및 응용 프로그램 서비스.** 이 서비스는 분산된 구성 요소들이 서로 통신할 수 있도록 하며 광범위한 서버, 클라이언트 및 장치를 위한 응용 프로그램의 개발, 배포 및 관리를 지원합니다. 이 서비스의 기반은 Java 2 Platform, Enterprise Edition(J2EE™) 기술입니다.
- **가용성 서비스.** 이 서비스는 응용 프로그램 및 웹 서비스에 대해 지속적인 가용성과 확장성을 제공합니다.

다음 표에서는 이전 서비스 범주를 나열하고 각 범주에 대해 서비스를 제공하는 Java Enterprise System 구성 요소를 지정합니다.

표 1-1 Java Enterprise System 서비스 범주

서비스 범주	Java Enterprise System 구성 요소
포털 서비스	Portal Server
	Portal Server Secure Remote Access
	Access Manager
	Directory Server
	Application Server 또는 Web Server
통신 및 공동 작업 서비스	Messaging Server
	Calendar Server
	Instant Messaging
	Access Manager
	Directory Server
Network Identity 서비스	Application Server 또는 Web Server
	Access Manager
	Directory Server
웹 및 응용 프로그램 서비스	Web Server
	Application Server
	Message Queue
가용성 서비스	Web Server
	Sun Cluster
	Sun Cluster Agents

Java Enterprise System 서비스, 구성 요소 및 Java Enterprise System 구조 개념에 대한 자세한 내용은 **Sun Java Enterprise System 2005Q4 기술 개요**를 참조하십시오.

기본 제공 서비스 및 사용자 정의 개발 서비스

Java Enterprise System을 기반으로 하는 배포 솔루션은 일반적으로 다음과 같은 두 개의 일반 범주로 나뉩니다.

- **80:20 배포.** 이 솔루션은 주로 Java Enterprise System에서 제공하는 서비스로 구성됩니다. Java Enterprise System에서 약 80% 이상의 서비스를 제공합니다.
- **20:80 배포.** 이 솔루션은 상당수의 사용자 정의 개발 서비스와 타사 응용 프로그램으로 이루어져 있습니다.

80:20 및 20:80은 포괄적 개념입니다. 제공된 서비스 유형에 대한 정확한 백분율은 중요하지 않습니다. 그러나 백분율로 솔루션이 포함하는 사용자 정의의 양을 표시합니다.

Java Enterprise System은 Java ES가 제공하는 다양한 서비스 세트가 있어 80:20 배포에 적합합니다. 예를 들어, Java Enterprise System이 제공하는 서비스를 사용하여 엔터프라이즈 전체의 통신 시스템이나 엔터프라이즈 전체의 포털 시스템을 배포하는 것이 비교적 쉽습니다.

사용자 정의 개발이 필요한 경우 Java Enterprise System은 사용자 정의 개발 서비스 및 응용 프로그램의 작성 및 통합을 지원합니다.

17 페이지 “시스템 서비스”에 나열된 대부분의 서비스 범주는 80:20 배포를 전달하는데 사용할 수 있습니다. 예를 들면 통신 및 공동 작업 서비스는 전자 우편, 달력, 인스턴트 메시징 서비스를 제공하여 최종 사용자가 내용을 집계 및 개인화할 수 있습니다. 마찬가지로 서비스의 엔터프라이즈 포털 범주 및 네트워크 아이디도 사용자 정의 서비스를 개발 또는 통합하지 않고도 엔터프라이즈 전체 응용 프로그램을 설치 및 구성할 수 있게 합니다.

J2EE 플랫폼 서비스의 사용자 정의 개발을 필요로 하는 엔터프라이즈 솔루션은 Java Enterprise System 웹 및 응용 프로그램 서비스와 함께 제공되는 Application Server, Message Queue, 또는 Web Server를 활용할 수 있습니다.

엔터프라이즈 배포는 필요로 하는 사용자 정의 개발 서비스의 수에 따라 매우 다양할 수 있습니다. Java Enterprise System 서비스 간에는 상호 운용성이 있기 때문에 사용자의 특정 엔터프라이즈 요구 사항에 맞게 고유한 서비스 제품군을 작성할 수 있습니다.

Java Enterprise System으로 마이그레이션

Java Enterprise System을 사용하는 엔터프라이즈 솔루션의 계획, 설계, 구현은 주로 현재 배포 전략에 달려 있습니다. 배포 솔루션을 처음 계획하고 있는 기업에 있어서 계획, 설계 및 구현은 대부분 기업의 특정 요구에 의해 구성됩니다. 그러나 최초 배포 솔루션이라고 해서 특별하지는 않습니다. 기존 엔터프라이즈 솔루션을 향상하기 위해 또는 Java Enterprise System 구성 요소의 이전 버전을 업그레이드하기 위해 Java Enterprise System을 사용하는 솔루션과 거의 비슷합니다.

기존 솔루션을 교체 및 업그레이드하는 경우 기존 데이터를 보존하고 소프트웨어를 현재 버전으로 제대로 업그레이드하도록 추가 계획, 설계, 구현 단계를 취해야 합니다. 이 설명서에서 설명한 분석 및 설계를 진행하면서 기존 소프트웨어 시스템의 교체 및 업그레이드에 필요한 준비 및 계획도 고려해야 합니다.

Java Enterprise System의 현재 버전으로 업그레이드하는 방법 및 다른 응용 프로그램으로부터 마이그레이션하기 위한 전략에 대한 자세한 내용은 *Java Enterprise System 업그레이드 및 마이그레이션 설명서*를 참조하십시오.

배포 계획 정보

배포 계획은 Java Enterprise System 솔루션의 성공적인 구현에 있어서 중요한 단계입니다. 각 기업에는 고려할 고유의 목표, 요구 사항 및 우선 순위들이 있습니다. 성공적인 계획은 이러한 목표를 달성하기 위해 기업의 목표를 분석하고 비즈니스 요구 사항을 판별하는 것으로 시작합니다. 그 다음에 비즈니스 요구 사항을 기술 요구 사항으로 변환해야 기업의 목표를 달성할 수 있는 시스템을 설계 및 구현할 수 있는 기반으로 사용할 수 있습니다.

성공적인 배포 계획은 치밀한 준비, 분석 및 설계의 결과입니다. 계획 과정 중에 어디에서든 발생할 수 있는 오류 및 실수는 시스템의 여러면에서 문제를 일으킬 수 있습니다. 제대로 계획되지 못한 시스템에서 중대한 문제가 발생할 수 있습니다. 예를 들면 시스템의 성능 저하, 유지 보수의 어려움, 운영에 너무 많은 비용 사용, 자원 낭비 또는 증가하는 요구를 충족하기 위한 크기 조정 불가와 같은 문제가 발생할 수 있습니다.

솔루션 라이프 사이클

다음 그림에 나타난 솔루션 라이프사이클은 Java Enterprise System을 기반으로 하는 엔터프라이즈 소프트웨어 솔루션의 계획, 설계 및 구현 단계를 보여줍니다. 라이프 사이클은 배포 계획이 제대로 진행되도록 하는데 유용한 도구입니다.

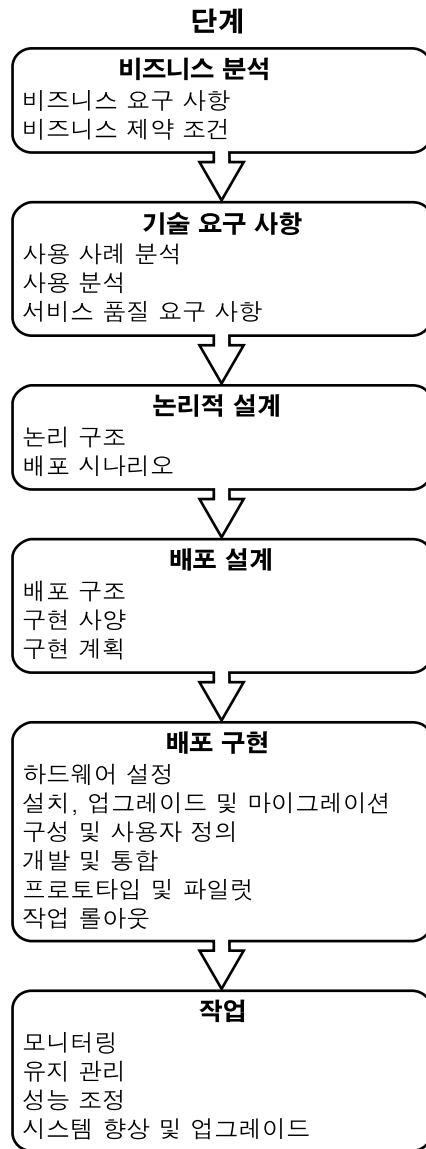


그림 1-1 솔루션 라이프 사이클

라이프 사이클은 정해진 순서의 단계로 구성됩니다. 각 단계는 결과가 후속 단계에 대한 입력으로 전달되는 관련 작업으로 이루어집니다. 각 단계 내 작업은 그에 대한 결과를 생성하기 전에 분석과 설계를 해야 하는 반복 작업입니다. 초기 단계 또한 반복적일 수 있습니다. 예를 들면 배포 설계 단계 중에 초기 단계의 분석이 불충분하여 추가 작업이 필요함을 발견할 수도 있습니다.

이 장의 다음 절에서는 각 라이프 사이클 단계를 간략하게 설명합니다.

비즈니스 분석 단계

비즈니스 분석 중에 배포 프로젝트의 비즈니스 목표를 정의하고 해당 목표를 달성하기 위해 충족해야 하는 비즈니스 요구 사항을 기술합니다. 비즈니스 요구 사항을 기술할 때 비즈니스 목표를 달성하기 위한 기능에 영향을 줄 수 있는 비즈니스 제약 조건을 고려합니다. 라이프 사이클 전체에서 비즈니스 분석 단계에서 수행된 분석에 따라 배포 설계 성공 및 궁극적으로 배포 솔루션을 측정합니다.

비즈니스 분석 단계 중에 나중에 기술 요구 사항 단계에 대한 입력으로 사용할 비즈니스 요구 사항 문서를 작성합니다.

비즈니스 분석 단계에 대한 자세한 내용은 2 장을 참조하십시오.

기술 요구 사항 단계

기술 요구 사항 단계는 비즈니스 분석 단계 중에 정의한 비즈니스 요구 사항 및 제약 조건으로 시작되며 이 요구 사항을 나중에 배포 구조를 설계하는 데 사용할 수 있는 기술 사양으로 변환합니다. 기술 요구 사항은 성능, 가용성, 보안 등과 같은 서비스 품질(QoS) 기능을 지정합니다.

기술 요구 사항 단계 중에 다음과 같은 정보를 포함하는 문서를 작성합니다.

- 사용자 작업 및 사용 패턴 분석
- 사용자와 계획된 시스템 간 상호 작용을 모델링하는 사용 사례
- 사용자 작업 및 사용 패턴 분석을 고려하여 비즈니스 요구 사항에서 기초한 서비스 품질 사항

결과로 만들어지는 사용 분석, 사용 사례 및 서비스 품질 요구 사항 문서는 솔루션 라이프 사이클의 논리적 설계 단계로 입력됩니다. 사용 분석은 또한 배포 설계 단계에서 중요한 역할을 합니다.

기술 요구 사항 단계 중에 나중에 서비스 수준 계약(SLA)을 만드는 기반인 서비스 수준 요구 사항을 지정할 수도 있습니다. 서비스 수준 계약은 시스템을 유지 보수하기 위해 어떤 고객 지원을 제공해야 하는지에 대한 조건을 지정하고 일반적으로 배포 설계 단계에서 프로젝트 승인의 일부로 서명됩니다.

기술 요구 사항에 대한 자세한 내용은 3 장을 참조하십시오.

논리적 설계 단계

논리적 설계 중에 기술 요구 사항 단계의 사용 사례를 입력으로 사용하여 솔루션을 구현하는 데 필요한 Java Enterprise System 구성 요소를 식별합니다. Java ES 구성 요소를 지원할 수 있는 구성 요소를 식별하고 또한 비즈니스 요구 사항을 충족하는 데 필요한 추가 사용자 정의 개발 구성 요소도 식별합니다. 그 다음에 구성 요소 간 상호 관련성을 보여주는 논리적 구조 안에 구성 요소를 매핑합니다. 논리적 구조는 솔루션을 구현하는데 필요한 어떤 하드웨어도 지정하지 않습니다.

논리적 설계 단계 결과는 논리적 구조입니다. 논리적 구조 그 자체는 배포 설계를 시작하기에 충분하지 않습니다. 기술 요구 사항에서 추출한 서비스 품질 요구 사항도 필요합니다. 기술 요구 사항 단계에서 추출한 논리적 구조 및 서비스 품질 요구 사항으로 배포 시나리오를 형성합니다. 이 배포 시나리오는 배포 설계 단계로 입력됩니다.

논리적 설계에 대한 자세한 내용은 4 장을 참조하십시오.

배포 설계 단계

배포 설계 중에 논리적 구조에서 지정한 구성 요소를 물리적 환경에 매핑하여 높은 수준의 배포 구조를 만듭니다. 또한 배포 구조 구축 방법을 지정하는 낮은 수준의 설명을 제공하는 구현 사양도 만듭니다. 소프트웨어 솔루션 구현의 다른 요소를 설명하는 일련의 계획 및 사양도 추가로 만듭니다.

프로젝트는 배포 설계 단계 중에 승인됩니다. 프로젝트 승인 중에 배포에 대한 비용을 평가합니다. 승인되는 경우 배포 구현 계약에 서명하고 프로젝트를 구축하기 위한 자원을 취득합니다. 종종 구현 사양을 자세히 기술한 이후에 프로젝트가 승인되기도 합니다. 그러나 배포 구조 완료 시 승인되는 경우도 있습니다.

배포 설계 단계에는 다음과 같은 결과가 포함됩니다.

- **배포 구조.** 네트워크 하드웨어 및 소프트웨어에 대한 구성 요소 매핑을 나타내는 높은 수준의 설계 문서입니다.
- **구현 사양.** 배포를 구축하기 위한 청사진으로 사용되는 세부 사양입니다.
- **구현 계획.** 엔터프라이즈 소프트웨어 솔루션 구현의 다양한 요소를 처리하는 계획 및 사양 그룹입니다. 구현 계획은 마이그레이션 계획, 설치 계획, 사용자 관리 계획, 테스트 계획 등을 포함합니다.

배포 설계에 대한 자세한 내용은 5 장을 참조하십시오.

구현 단계

구현 단계 중에 배포 설계 동안 만든 사양과 계획으로 작업하여 배포 구조를 구축하고 솔루션을 구현합니다. 배포 프로젝트의 특성에 따라 이 단계에는 다음 작업 중 일부 또는 전부가 포함됩니다.

- 하드웨어 인프라 설치 및 구성
- 소프트웨어 설치 및 구성
- LDAP 디렉토리 설계 내 사용자 및 자원 모델링
- 사용자 관리 계획에 따라 기존 디렉토리 및 데이터베이스로 부터 데이터 마이그레이션
- 테스트 환경에서 파일럿 및 프로토타입 배포 작성 및 배포
- 시스템 요구 사항 호환성을 측정하기 위한 기능 테스트 설계 및 실행
- 최고 부하 시 성능을 측정하기 위한 스트레스 테스트 설계 및 실행

- 모든 사용자 정의 엔터프라이즈 응용 프로그램 개발 및 통합
- 단계적으로 생산에 들어갈 수 있는 생산 배포 작성
 - 일단 배포가 작업 환경으로 들어가면 솔루션 라이프 사이클의 운영 단계로 진행합니다.

구현 단계에 대한 자세한 내용은 6 장을 참조하십시오.

운영 단계

운영 단계에서 배포 구현을 원활하게 수행하도록 유지하는데 필요한 작업을 처리합니다. 이 단계에 다음이 포함됩니다.

- 시스템이 계획에 따라 실행되도록 배포 모니터링
- 배포된 소프트웨어가 최적의 수준에서 실행되도록 성능 조정
- 원활한 운영을 위한 예정된 유지 보수 및 필요에 따라 예정되지 않은 유지 보수 제공
- 필요에 따라 소프트웨어 및 하드웨어 업그레이드

운영 단계에 대한 자세한 설명은 이 설명서 범위에서 벗어납니다.

비즈니스 분석

솔루션 라이프 사이클의 비즈니스 분석 단계 중에 비즈니스 문제를 분석하고 목표를 충족시킬 비즈니스 요구 사항 및 비즈니스 제약 조건을 식별하여 비즈니스 목표를 정의합니다.

이 장의 내용은 다음과 같습니다.

- 25 페이지 “비즈니스 분석 정보”
- 26 페이지 “비즈니스 요구 사항 정의”
- 30 페이지 “비즈니스 제약 조건 정의”

비즈니스 분석 정보

비즈니스 분석은 비즈니스 목표를 기술하는 것으로 시작합니다. 그런 다음 해결해야 할 비즈니스 문제를 분석하고 비즈니스 목표를 달성하기 위해 충족시켜야 할 비즈니스 요구 사항을 식별합니다. 목표를 달성할 수 있는 능력을 제한하는 비즈니스 제약 조건도 고려합니다. 비즈니스 요구 사항 및 제약 조건 분석 결과가 일련의 비즈니스 요구 사항 문서로 만들어집니다.

비즈니스 요구 사항 문서는 기술 요구 사항 단계에서 기술 요구 사항을 이끌어내기 위한 기준으로 사용됩니다. 솔루션 라이프 사이클을 통해서 배포 계획 및 궁극적으로 비즈니스 분석 단계에서 수행된 분석에 따른 솔루션의 성공을 측정합니다.

비즈니스 요구 사항 정의

모든 비즈니스 요구 사항을 식별할 수 있는 간단한 공식은 없습니다. 소프트웨어 솔루션이 필요한 이해 관계자와의 공동 작업과 비즈니스 영역에 대한 지식 및 응용된 창조적 생각을 기반으로 요구 사항을 판별해야 합니다.

이 절은 비즈니스 요구 사항을 정의할 때 고려해야 할 몇몇 요소를 제공합니다.

비즈니스 목표 설정

비즈니스 분석은 배포 프로젝트의 목표를 명확하게 나타내야 합니다. 명확한 목표는 설계 결정에 집중할 수 있게 하며 프로젝트가 다른 길로 빠지지 않도록 합니다. 비즈니스 목표와 현재 운영을 비교하면 설계 결정을 내리는 데 도움이 될 수 있습니다.

범위

비즈니스 요구 사항에 배포 프로젝트 범위를 기술해야 합니다. 해결할 수 있는 영역을 식별하고 목표를 불명확하거나 도달할 수 없게 만드는 제한 없는 요구 사항을 방지하도록 합니다. 범위를 제대로 정의하지 못하면 비즈니스 요구 사항을 불충분하게 설명하거나 자원을 지나치게 사용하는 배포 설계가 만들어질 수 있습니다.

우선 순위

목표의 우선 순위를 지정하면 배포의 가장 중요한 요소를 먼저 달성할 수 있습니다. 자원이 제한될 경우 일부 목표를 지연하거나 수정해야 할 수 있습니다. 예를 들면 크고 복잡한 배포에는 일반적으로 솔루션의 단계적인 구현이 필요할 수 있습니다. 우선 순위를 기술하면 이해 관계자의 승인을 받기 위해 배포 설계에서 결정해야 할 사항에 대한 지침을 제공할 수 있습니다.

중요한 품질

중요한 영역을 확인하면 이해 관계자와 설계자가 가장 중요한 기준에 집중할 수 있습니다.

성장 요소

비즈니스 목표를 설정하는 경우 조직의 현재 요구를 고려할 뿐만 아니라 이러한 요구가 장기간에 걸쳐 어떻게 변경되고 커질 수 있는 지도 예측해야 합니다. 그렇지 않으면 솔루션이 너무 빨리 구식이 되어버립니다.

안전 여유분

솔루션의 설계는 이 비즈니스 분석 단계 동안 만들어지는 가정을 기반으로 합니다. 이 가정은 불충분한 데이터 판단 상의 오류 또는 예기치 못한 외부 사고와 같은 다양한 이유로 인해 정확하지 않을 수 있습니다. 설계한 솔루션이 갑작스러운 사고에 대처할 수 있도록 비즈니스 목표에서만 아니라 계획 전반에 걸쳐서 안전을 위한 여유를 갖도록 계획을 세우십시오.

사용자 요구 이해

솔루션의 대상이 되는 사용자 유형과 그들의 요구, 기대되는 혜택을 이해할 수 있도록 필요한 연구를 하십시오. 예를 들면 다음 목록은 사용자를 범주화하는 한 방법입니다.

- 현재 직원 전용
- 현재 및 이전 직원
- 관리자
- 적극적인 고객
- 모든 고객
- 구성원 사이트
- 일반 대중
- 제한된 액세스

예상되는 혜택을 명확하게 기술하면 사용자가 설계 결정을 추진하는 데 도움이 됩니다. 예를 들면 다음은 솔루션으로 사용자가 받을 수 있는 혜택입니다.

- 회사 자원에 대한 원격 액세스
- 엔터프라이즈 공동 작업
- 일일 작업 단순화
- 원격 팀 간 자원 공유
- 생산성 향상
- 최종 사용자에게 의한 자체 관리

운영 상 요구 사항 개발

운영 상 요구 사항을 목표와 직접 관련된 일련의 기술 요구 사항으로 표현하십시오. 일반적으로 영역에 대한 운영 사양은 다음과 같이 만듭니다.

- 최종 사용자 기능
- 응답 시간 감소
- 가용성 및 가동 시간
- 오류 비율 감소
- 정보 보관 및 유지

운영 상 요구 사항은 모든 이해 관계자가 이해할 수 있는 측정 가능한 용어로 표현해야 합니다. “적절한 최종 사용자 응답 시간”과 같은 모호한 언어는 피해야 합니다. 운영 상 요구 사항에 대한 예는 다음과 같습니다.

- 중단된 서비스의 10분 내 복구 기능

- 인바운드 메시지 전달에 대한 최종 시간 재생 기능
- 최고 기간 동안 60초 내 완료된 온라인 트랜잭션
- 최고 기간 동안 4초 내 완료되는 최종 사용자 인증

기존 사용 패턴 지원

기존 사용 패턴을 명확히 측정할 수 있는 목표로 표현하십시오. 다음 질문은 그러한 목표를 결정하는데 도움을 줄 수 있습니다.

- 현재 서비스가 어떻게 이용됩니까?
- 사용 패턴은 무엇입니까(예:가끔, 자주 또는 다량 사용)?
- 일반적으로 사용자가 어떤 사이트에 연결합니까?
- 일반적으로 사용자가 어떤 크기의 메시지를 보냅니까?
- 일반적으로 사용자가 하루 당 또는 한시간 당 얼마나 많은 트랜잭션을 완료합니까?

서비스에 액세스하는 사용자를 연구하십시오. 사용자가 기존 서비스에 언제 그리고 얼마나 오래 액세스하는지와 같은 요소는 목표를 식별하는 열쇠가 됩니다. 조직의 경험이 이러한 패턴을 제공할 수 없는 경우 유사한 조직의 경험을 연구하십시오.

기업 문화 이해

요구 사항 분석은 기업 문화와 정책에 대한 다양한 요소를 고려해야 합니다. 기업 문화에 대한 주의 부족이 솔루션을 잘 받아들이지 않게 하거나 구현하기 어렵게 만들 수 있습니다.

이해 관계자

제안된 솔루션의 성공에 따라 수익을 얻는 개인이나 조직을 식별하십시오. 모든 이해 관계자가 비즈니스 목표와 요구 사항을 정의하는 데 적극적으로 참여해야 합니다. 이해 관계자가 참여하지 않거나 계획의 변경에 대한 정보를 듣지 못한 경우 계획에 중대한 결함이 발생할 수 있습니다. 그러한 이해 관계자는 심지어 배포의 구현을 방해할 수도 있습니다.

표준과 정책

솔루션을 요청한 조직의 표준과 정책을 이해해야 합니다. 이러한 표준과 정책이 설계, 제품 선택 및 배포 방법의 기술적인 요소에 영향을 미칠 수 있습니다.

한 예로 인사 데이터의 기밀이 인사부나 부서장에 의해 소유되거나 통제될 수 있습니다. 또 다른 예는 변경 사항 관리에 대한 회사 절차입니다. 변경 사항 관리 정책은 솔루션의 승인과 구현 방법 및 시간표에 상당한 영향을 미칠 수 있습니다.

규정 요구 사항

규정 요구 사항은 비즈니스 특성에 따라 크게 다릅니다. 배포에 영향을 미칠 수 있는 모든 규정 요구 사항을 연구하고 이해해야 합니다. 많은 회사와 정부 기관은 액세스 표준의 준수를 요구합니다. 전세계적인 솔루션을 배포할 때는 외국 법과 규정을 고려해야 합니다. 예를 들어 많은 유럽 국가들의 경우 개인 정보 저장에 대해 엄격히 규제하고 있습니다.

보안

식별한 목표에 강조해야 할 보안 문제가 암시되어 있을 수 있습니다. 솔루션에 필수적인 특정 보안 목표를 생각해 보아야 합니다. 예를 들면 다음과 같습니다.

- 독점 정보에 대한 승인된 액세스
- 기밀 정보에 대한 역할 기반 액세스
- 원격 위치 간의 보안 통신
- 로컬 시스템에서 원격 응용 프로그램 호출
- 타사 비즈니스 및 조직과의 보안 트랜잭션
- 보안 정책 시행

사이트 배포

사이트의 지형적 배포 및 사이트 내 대역폭이 설계 결정에 영향을 미칠 수 있습니다. 또한 어떤 사이트는 로컬 관리가 필요할 수 있습니다.

이러한 지형적 고려가 프로젝트의 교육비와 복잡도 등을 증가시킬 수 있습니다. 사이트의 지형적 배포 결과 나타난 요구 사항을 명확히 기술해야 합니다. 설계의 성공을 위해 어떤 사이트가 중요한지 강조해야 합니다.

증분 방법 사용

일반적으로 소프트웨어 솔루션은 전체적이고 포괄적인 시스템으로 간주됩니다. 그러나 종종 전체 시스템의 배포를 정확한 단계에 따라 증분적으로 달성하기도 합니다.

증분 방법을 적용할 경우 일반적으로 최종의 포괄적인 솔루션에 이르는 이정표를 제공하는 로드맵을 설계합니다. 또한 나중에 구현하기 위해 연기된 포괄적인 솔루션 요소에 대해 단기 계획을 고려해야 할 수 있습니다.

증분 방법은 다음과 같은 장점이 있습니다.

- 비즈니스 성장에 따른 요구 사항 변경에 적응할 수 있습니다.
- 최종 배포 구현으로 전환할 때 기존 인프라를 활용할 수 있습니다.
- 자본 지출 요구 사항을 고려할 수 있습니다.
- 적은 인적 자원 공급을 활용할 수 있습니다.
- 제휴 관계 가능성을 고려할 수 있습니다.

서비스 수준 계약 이해

서비스 수준 계약(SLA)은 최소 성능 요구 사항과 해당 요구 사항 달성 실패, 제공되어야 할 고객 지원의 수준 및 범위에 대해 지정합니다. 서비스 수준 계약은 비즈니스 분석 중에 정의된 비즈니스 요구 사항을 기반으로 하며 이 요구 사항은 나중에 기술 요구 사항 단계 중 서비스 수준 요구 사항으로 정의됩니다. SLA는 프로젝트 승인 과정 동안 서명되며 배포 설계 단계에서 나타납니다.

SLA를 가동 시간과 응답 시간, 메시지 전달 시간, 재해 복구 등과 같은 영역에서 개발해야 합니다. SLA는 시스템 개요와 지원 조직의 책임 역할, 서비스 수준 측정 방법 및 요청 변경 등과 같은 항목을 고려해야 합니다. 시스템 가용성에 대한 조직의 기대를 식별하는 것이 SLA의 범위를 결정하는 열쇠입니다.

비즈니스 제약 조건 정의

비즈니스 제약 조건은 배포 프로젝트의 특성을 결정하는 데 중요한 역할을 수행합니다. 성공적인 배포 설계의 열쇠는 알려진 비즈니스 제약 조건 내에서 비즈니스 요구 사항을 충족시킬 수 있는 최적의 방법을 찾는 것입니다. 비즈니스 제약 조건은 회계 상의 제한과 물리적인 제한(예를 들면 네트워크 용량)과 시간 제한(예를 들면 다음 연례 회의와 같은 중요한 이벤트 전 완료) 또는 비즈니스 목표 달성에 영향을 미칠 요소로서 예측할 수 있는 기타 다른 제한이 될 수 있습니다.

이 절에서는 비즈니스 제약 조건을 정의할 때 고려해야 할 요소를 설명합니다.

마이그레이션 문제

일반적으로 배포 프로젝트는 기존 소프트웨어 인프라와 데이터를 대체하거나 보완합니다. 모든 새로운 솔루션은 데이터와 절차를 기존 인프라로부터 종종 기존 응용 프로그램과 상호 운용성을 보유한 새로운 솔루션으로 마이그레이션할 수 있어야 합니다. 현재 인프라 분석은 제안된 솔루션에 대해 수행할 마이그레이션 문제의 범위를 결정하는데 필수적입니다.

일정 요구 사항

솔루션 구현을 위한 일정이 설계 결정에 영향을 미칠 수 있습니다. 빡빡한 일정으로 인해 목표가 축소되거나, 우선 순위가 변경되거나, 충분 해결 방법이 채택될 수 있습니다. 일정 내에는 고려할 만한 중요한 이정표가 있을 수도 있습니다. 이정표는 예정된 서비스 발표와 같은 내부 이벤트 또는 학교의 개학일과 같은 외부 이벤트에 의해 설정될 수 있습니다.

예산 제한

대부분의 배포 프로젝트는 예산을 고수해야 합니다. 제안된 솔루션 구축 비용과 솔루션을 특정 수명 기간 동안 유지 보수하는데 필요한 자원을 고려할 때 다음 사항을 포함해야 합니다.

- **기존 하드웨어 및 네트워크 인프라.** 기존 인프라에 대한 의존이 시스템 설계에 영향을 미칠 수 있습니다.
- **솔루션 구현에 필요한 개발 자원.** 하드웨어, 소프트웨어 및 인적 자원 등을 포함한 개발 자원이 제한된 경우 증분 배포를 제안할 수 있습니다. 구현하는 각 증분 단계마다 동일한 자원이나 개발 팀을 다시 사용해야 할 수도 있습니다.
- **유지 보수, 관리 및 지원.** 시스템의 사용자를 관리, 유지 보수 및 지원하는데 사용할 수 있는 자원을 분석합니다. 제한된 자원이 설계 결정에 영향을 줄 수 있습니다.

소유 비용

유지 보수, 관리 및 지원 외에 소유 비용에 영향을 미치는 다른 요소도 분석해야 합니다. 하드웨어 및 소프트웨어의 업그레이드가 필요할 수 있으며 전선망에서의 솔루션 충돌, 텔레커뮤니케이션 비용 및 기타 요소가 현금 지출에 영향을 미칩니다. 솔루션에 대한 가용성 수준을 지정하는 서비스 수준 계약도 중복의 증가로 인해 소유 비용에 영향을 미칩니다.

솔루션의 구현이 솔루션에 대한 투자를 돌려주어야 합니다. 일반적으로 투자 수익 분석에서는 자본 지출에 따른 재무 이익을 측정합니다.

솔루션의 재무 이익을 평가할 때는 달성된 목표와 해당 목표를 달성하는 다른 방법과 비교하거나 아무 것도 수행하지 않을 경우의 비용과 비교하여 세밀하게 분석합니다.

기술 요구 사항

솔루션 라이프 사이클의 기술 요구 사항 단계 중에 사용 분석을 수행하고 사용 사례를 식별하며 제안된 배포 솔루션에 대한 서비스 품질 요구 사항을 결정합니다.

이 장의 내용은 다음과 같습니다.

- 33 페이지 “기술 요구 사항 정보”
- 34 페이지 “사용 분석”
- 35 페이지 “사용 사례”
- 36 페이지 “서비스 품질 요구 사항 적용”
- 43 페이지 “서비스 수준 요구 사항”

기술 요구 사항 정보

기술 요구 사항 분석은 솔루션 라이프 사이클의 비즈니스 분석 단계 중에 작성된 비즈니스 요구 사항 문서로 시작됩니다. 비즈니스 분석을 기반으로 하여 다음을 수행합니다.

- 예상되는 로드 조건을 판별할 수 있도록 사용 분석을 수행합니다.
- 시스템과의 일반적인 사용자 상호 작용을 모델로 사용할 수 있는 사용 사례를 만듭니다.
- 응답 시간, 가용성 등과 같은 영역에서 배포된 솔루션이 수행해야 하는 방법을 정의하는 서비스 품질(QoS) 요구 사항을 만듭니다.

이전에 식별한 비즈니스 요구 사항 및 제약 조건을 고려하면서 사용 분석 및 사용 사례에서 서비스 품질 요구 사항을 이끌어 냅니다.

서비스 품질 요구 사항은 논리적 설계 단계의 논리적 구조와 나중에 쌓이 되어 배포 시나리오를 구성합니다. 배포 시나리오는 솔루션 라이프 사이클의 배포 설계 단계에 대한 주요 입력이 됩니다.

비즈니스 분석에서와 마찬가지로 사용 분석, 사용 사례 및 시스템 요구 사항을 생성하는 기술 요구 사항 분석을 위한 단순한 공식은 없습니다. 기술 요구 사항 분석에는 비즈니스 영역, 비즈니스 목표 및 기본 시스템 기술에 대한 이해가 필요합니다.

사용 분석

사용 분석에서는 설계하려는 솔루션의 다양한 사용자를 식별하고 이러한 사용자들의 사용 패턴을 관별합니다. 수집한 정보는 시스템의 로드 조건을 예상하기 위한 기반을 제공합니다. 또한 사용 분석 정보는 35 페이지 “사용 사례”에 설명되어 있는 사용 사례에 가중치를 할당할 때 사용할 수 있습니다.

사용 분석 동안에는 가능하면 사용자를 인터뷰하고 사용 패턴에 대한 기존 데이터를 조사하며 이전 시스템의 설계자와 관리자도 인터뷰해야 합니다. 다음 표에서는 사용 분석을 수행할 때 고려해야 할 요소를 나열합니다.

표 3-1 사용 분석 요소

항목	설명
사용자 수 및 유형	<p>솔루션에서 지원해야 할 사용자 수를 식별하고 필요하면 해당 사용자를 범주화합니다.</p> <p>예를 들면 다음과 같습니다.</p> <ul style="list-style-type: none"> ■ B2C(기업-사용자) 솔루션에는 방문자가 많을 수 있지만 비즈니스 트랜잭션에 등록 및 관여하는 사용자 수는 적습니다. ■ B2E(기업-직원) 솔루션에는 일부 직원이 기업 네트워크 밖에서 액세스할 필요가 있을 수도 있겠지만 일반적으로 각 직원을 수용합니다. B2E 솔루션에서 관리자는 정규 직원이 액세스할 수 없는 영역에 대한 권한 부여가 필요할 수 있습니다.
활성 및 비활성 사용자	<p>활성 및 비활성 사용자의 사용 패턴과 비율을 식별합니다.</p> <p>활성 사용자는 시스템에 로그인하여 시스템의 서비스와 상호 작용하는 사용자입니다. 비활성 사용자는 로그인하지 않은 사용자, 로그인했지만 시스템 구성 요소와 상호 작용하지 않는 사용자 또는 데이터베이스에는 있지만 한 번도 로그인하지 않은 사용자입니다.</p>
관리 권한이 있는 사용자	<p>배포된 시스템을 액세스하는 배포를 모니터, 업데이트 및 지원하는 사용자를 식별합니다.</p> <p>기술 요구 사항에 영향을 미칠 수 있는 특정 관리 사용 패턴(예를 들면 방화벽 밖에서 배포 관리)을 관별합니다.</p>

표 3-1 사용 분석 요소 (계속)

항목	설명
사용 패턴	<p>다양한 유형의 사용자가 시스템을 액세스하고 예상되는 사용의 대상을 제공하는 방법을 식별합니다.</p> <p>예를 들면 다음과 같습니다.</p> <ul style="list-style-type: none"> ■ 사용이 급상승하는 최고 시간이 있습니까? ■ 업무 시간은 언제입니까? ■ 사용자가 전 세계에 분산되어 있습니까? ■ 예상되는 사용자 연결 시간은 얼마입니까?
사용자 증가	<p>사용자 기반 크기가 고정되어 있는지 또는 배포에서 사용자 수의 증가를 예상하는지 여부를 결정합니다.</p> <p>사용자 기반이 증가할 것으로 예상되는 경우 이러한 증가를 합리적으로 예측해 봅니다.</p>
사용자 트랜잭션	<p>지원해야 할 사용자 트랜잭션 유형을 식별합니다. 이 사용자 트랜잭션을 사용 사례로 변환할 수 있습니다.</p> <p>예를 들면 다음과 같습니다.</p> <ul style="list-style-type: none"> ■ 사용자가 수행할 작업은 무엇입니까? ■ 사용자가 로그인하면 로그인한 상태로 남아 있습니까? 대개 몇 가지 작업을 수행하고 로그아웃합니까? ■ 공통 달력, 웹 회의실 및 내부 웹 페이지 배포가 필요한 사용자 간에 중요한 공동 작업이 있습니까?
사용자 연구 및 통계 데이터	<p>기존 사용자 연구 및 기타 소스를 사용하여 사용자 동작 패턴을 결정합니다.</p> <p>종종 기업이나 산업 조직에서는 사용자에게 대한 유용한 정보를 추출할 수 있는 사용자 조사 연구를 수행합니다. 기존 응용 프로그램의 로그 파일에는 시스템 예측을 하는 데 유용한 통계 데이터가 포함되어 있을 수 있습니다.</p>

사용 사례

사용 사례에서는 설계 중인 솔루션과 사용자 간의 일반적인 상호 작용을 모델링하여 최종 사용자 관점에서 전체 작업 흐름을 설명합니다. 사용 사례의 전체 집합과 관련하여 설계의 우선 순위를 정하면 예상되는 기능의 제공에 계속 초점을 맞출 수 있습니다. 사용 사례는 논리적 설계에 대한 중요한 입력이 됩니다.

가중치가 가장 높은 사용 사례가 가장 일반적인 사용자 작업을 나타내도록 상대적인 가중치를 사용 사례에 할당합니다. 사용 사례 가중치는 가장 많이 사용되는 시스템 서비스의 설계 결정에 초점을 맞출 수 있게 합니다.

다음 두 가지 수준으로 사용 사례를 설명할 수 있습니다.

- **사용 사례 보고서.** 기본 및 대체 이벤트 흐름을 포함하는 개별 사용 사례에 대한 설명입니다.
- **사용 사례 다이어그램.** 이벤트 흐름의 보다 형식적인 조직을 나타내면서 작업자와 사용 사례 간 관계를 설명하는 다이어그램입니다. 사용 사례 다이어그램은 장기 또는 복잡한 사용 사례를 모델링하는데 유용합니다. 일반적으로 통합 모델링 언어(UML) 표준을 사용하여 사용 사례 다이어그램을 작성합니다.

서비스 품질 요구 사항 적용

서비스 품질(QoS) 요구 사항은 성능, 가용성, 확장성 및 서비스 가능성과 같은 기능의 시스템 품질을 지정하는 기술 사양입니다. 서비스 품질 요구 사항은 비즈니스 요구 사항에 지정된 비즈니스 요구에 의해 구성됩니다. 예를 들면 서비스를 일년 내내 24시간 사용할 수 있어야 하는 경우 가용성 요구 사항에서 이 비즈니스 요구 사항을 처리해야 합니다.

다음 표에서는 일반적으로 서비스 품질 요구 사항의 기반을 구성하는 시스템 품질을 나열합니다.

표 3-2 서비스 품질 요구 사항에 영향을 미치는 시스템 품질

시스템 품질	설명
성능	사용자 로드 조건에 따른 응답 시간 및 처리 능력 측정
가용성	최종 사용자가 시스템의 자원 및 서비스에 액세스할 수 있는 빈도에 대한 측정이며 대개 시스템의 가동 시간 으로 표시됩니다.
가용성	시간에 따라 배포된 시스템에 용량 및 사용자를 추가할 수 있는 기능. 확장성은 일반적으로 시스템에 자원을 추가하는 것을 포함하지만 배포 구조 변경을 요구해서는 안 됩니다.
보안	시스템과 그 사용자의 무결성을 설명하는 요소들의 복잡한 조합. 보안은 사용자의 권한 부여 및 인증, 데이터 보안 및 배포된 시스템으로의 보안 액세스를 포함합니다.
잠재 용량	추가 자원 없이 비정상적인 최고 로드 사용을 처리할 수 있는 시스템의 기능. 잠재 용량은 가용성, 성능 및 확장성 품질의 요소입니다.
서비스 가능성	시스템 모니터링, 발생하는 문제 처리, 하드웨어 및 소프트웨어 구성 요소 업그레이드 등을 포함하여 배포된 시스템의 관리 용이성.

시스템 품질은 서로 밀접하게 관련되어 있습니다. 한 가지 시스템 품질의 요구 사항이 다른 시스템 품질의 요구 사항 및 설계에 영향을 미칠 수 있습니다. 예를 들어 높은 수준의 보안은 성능에 영향을 미칠 수 있고 성능은 다시 가용성에 영향을 미칠 수 있습니다. 가용성 문제를 처리하기 위해 서버를 추가하면 서비스 가능성(유지 보수 비용)에 영향을 미칠 수 있습니다.

시스템 품질이 어떻게 연관되어 있는지 이해하고 다른 품질 간의 균형을 조절하는 것이 비즈니스 요구 사항과 비즈니스 제약 조건을 모두 성공적으로 충족시키는 시스템을 설계할 수 있는 비결입니다.

다음 절에서는 서비스 품질 요구 사항을 구성할 때 고려할 요소에 대한 지침을 제공하면서 더 나아가 배포 설계에 영향을 미치는 시스템을 설명합니다. 서비스 수준 계약의 기반을 구성하는 서비스 수준 요구 사항에 관한 절도 포함됩니다.

성능

비즈니스 요구 사항은 일반적으로 비기술적 용어로 응답 시간을 지정하는 성능을 표현합니다. 예를 들어 웹 기반 액세스에 대한 비즈니스 요구 사항에서 다음과 같이 기술할 수 있습니다.

사용자는 로그인 시 일반적으로 4초 이하의 적당한 응답 시간을 기대합니다.

이 비즈니스 요구 사항을 출발점으로 하여 모든 사용 사례를 조사하고 이 요구 사항을 시스템 수준으로 표현하는 방법을 결정합니다. 어떤 경우에 사용 분석 중에 결정된 사용자 로드 조건을 포함하고자 할 수도 있습니다. 각 사용 사례의 성능 요구 사항을 지정한 로드 조건에 따른 응답 시간이나 응답 시간 및 처리 능력으로 표현합니다. 허용 가능한 오류 수를 지정할 수도 있습니다.

다음은 성능에 대한 시스템 요구 사항을 지정하는 방법의 두 가지 예입니다.

- 웹 페이지 새로 고침에 대한 응답은 하루종일 4초보다 길어서는 안 되며 15분 간격으로 측정되고 오류가 백만 개의 트랜잭션 당 3.4개보다 적어야 합니다.
- 정의한 최고 기간 중에 시스템은 모든 사용자에게 대해 응답 시간이 12초 이상이어서는 안 되며 오류가 백만 개의 트랜잭션 당 3.4개보다 적은 상태로 초 당 보안 로그인을 25개 허용해야 합니다.

성능 요구 사항은 가용성 요구 사항(폐일오버가 성능에 어떤 영향을 미치는지) 및 잠재 용량(비정상적인 최고 로드를 처리하기 위한 용량이 얼마나 있는지)과 밀접하게 관련되어 있습니다.

가용성

가용성은 시스템의 가동 시간을 지정하는 한 방법으로 일반적으로 사용자가 시스템을 액세스할 수 있는 시간 백분율로 측정합니다. 시스템에 액세스할 수 없는 시간(중단 시간)은 하드웨어, 소프트웨어 또는 네트워크의 오류나 시스템을 중단시키는 기타 요소(예: 정전)로 인한 것일 수 있습니다. 서비스의 예정된 중단 시간(유지 보수 및 업그레이드)은 중단 시간으로 고려하지 않습니다. 시스템 가용성을 계산하는 기본 등식을 가동 시간에 대한 백분율로 보면 다음과 같습니다.

$$\text{Availability} = \text{uptime} / (\text{uptime} + \text{downtime}) * 100\%$$

대개 사용자가 달성할 수 있는 “9의 개수”로 가용성을 측정합니다. 예를 들어 99% 가용성은 9가 두 개입니다. 9를 추가로 지정하면 배포 설계에 상당한 영향을 미칩니다. 다음 표는 하루 24시간 365일(총 8,760시간) 실행되는 시스템의 가용성에 9를 추가하여 예정되지 않은 중단 시간을 계산한 것입니다.

표 3-3 1년 내내(8,760시간) 실행되는 시스템의 예정되지 않은 중단 시간

9의 개수	사용 가능한 백분율	예정되지 않은 중단 시간
2	99%	88시간
3	99.9%	9시간
4	99.99%	45분
5	99.999%	5분

고장 허용 시스템

9가 네 개 또는 다섯 개인 가용성 요구 사항에서는 일반적으로 고장 허용 시스템을 요구합니다. 고장 허용 시스템은 하드웨어나 소프트웨어 오류 중에도 서비스를 계속할 수 있어야 합니다. 일반적으로 고장 허용은 중요 서비스를 제공하는 하드웨어(예: CPU, 메모리 및 네트워크 장치)와 소프트웨어 모두의 중복을 통해 달성됩니다.

단일 오류 지점은 중요 경로의 일부이지만 중복 구성 요소가 백업하지 않는 하드웨어 또는 소프트웨어 구성 요소입니다. 이 구성 요소의 실패는 시스템의 서비스가 손실을 일으킵니다. 고장 허용 시스템을 설계할 때는 잠재적인 단일 오류 지점을 식별하여 제거해야 합니다.

고장 허용 시스템은 구현 및 유지 보수 비용이 많이 들 수 있습니다. 가용성에 대한 비즈니스 요구 사항 특성을 이해하고 이러한 요구 사항을 충족시키는 가용성 솔루션의 전략과 비용을 고려해야 합니다.

서비스 가용성 우선 순위 지정

사용자 관점에서 가용성은 종종 전체 시스템의 가용성보다는 서비스별로 적용합니다. 예를 들면 Instant Messaging Service의 비가용성은 대개 다른 서비스의 가용성에 영향을 적게 미치거나 영향을 미치지 않습니다. 그러나 많은 다른 서비스가 종속된 서비스(예: Directory Server)의 비가용성은 보다 폭 넓은 영향을 미칩니다. 높은 가용성 사양은 가용성 증가가 필요한 특정 사용 사례 및 사용 분석을 확실하게 참조해야 합니다.

정렬된 우선 순위 집합에 따라 가용성 요구 사항을 나열하는 것이 도움이 될 수 있습니다. 다음 표에서는 여러 서비스 유형의 가용성에 대한 우선 순위를 지정합니다.

표 3-4 우선 순위별 서비스 가용성

우선 순위	서비스 유형	설명
1	임무 결정적	항상 사용 가능해야 하는 서비스 예를 들면 응용 프로그램에 대한 데이터베이스 서비스(예: LDAP 디렉토리)입니다.

표 3-4 우선 순위별 서비스 가용성 (계속)

우선 순위	서비스 유형	설명
2	사용 가능해야 함	사용 가능해야 하지만 성능은 떨어져도 관계 없는 서비스예를 들면 메시지 서비스 가용성은 일부 비즈니스 환경에서는 중요하지 않을 수도 있습니다.
3	연기할 수 있음	지정한 기간 내에 사용 가능해야 하는 서비스예를 들면 달력 서비스 가용성은 일부 비즈니스 환경에서 필수적이지 않을 수도 있습니다.
4	선택 사항	무기한 연기할 수 있는 서비스예를 들면 일부 환경에서는 Instant Messaging Service 가 유용하지만 필수적이지는 않다고 간주할 수 있습니다.

서비스 손실

가용성 설계는 가용성이 문제가 될 때 또는 구성 요소를 손실했을 때 발생하는 상황에 대한 고려를 포함합니다. 이것은 연결된 사용자가 세션을 재시작해야 하는지 및 한 영역의 실패가 시스템의 다른 영역에 어떻게 영향을 미치는지에 대한 고려를 포함합니다. 서비스 품질 요구 사항은 이러한 시나리오를 고려하여 이러한 상황에서 배포가 반응하는 방법을 지정해야 합니다.

가용성

확장성은 시스템에 용량을 추가하여 시스템이 기존 사용자 또는 증가된 사용자 기반으로부터의 추가 로드를 지원할 수 있도록 하는 기능입니다. 대개 확장성은 자원 추가를 요구하지만 배포 구조의 설계 변경이나 자원 추가에 필요한 시간으로 인한 서비스 손실을 요구해서는 안 됩니다.

가용성처럼 확장성도 전체 시스템보다는 시스템에서 제공하는 개별 서비스에 적용되는 경우가 많습니다. 그러나 **Directory Server**처럼 다른 서비스가 종속되어 있는 서비스의 경우 확장성은 시스템 전체에 영향을 미칠 수 있습니다.

비즈니스 요구 사항에서 예상되는 배포의 증가를 명확하게 기술하지 않는 경우 서비스 품질 요구 사항과 함께 확장성 요구 사항을 지정할 필요는 없습니다. 그러나 솔루션 라이프 사이클의 배포 설계 단계 중에 배포 구조는 확장성을 위한 서비스 품질 요구 사항을 지정하지 않았다고 하더라도 시스템을 확장하기 위한 일부 허용을 추가해야 합니다.

증가 예상

확장성 요구 사항을 결정하기 위한 시스템 증가 예상은 달성할 수 없을지도 모르는 예상, 예측 및 추측 작업을 포함합니다. 확장 가능한 시스템 요구 사항을 개발하기 위한 세 가지 비결은 다음과 같습니다.

- **고성능 설계 전략.** 성능 요구 사항 지정 중에 시간에 따라 증가할 수 있는 로드를 처리하기 위한 잠재 용량을 포함시킵니다. 또한 예산 제약 조건 내에서 가용성을 최대화합니다. 이 전략을 사용하면 성장에 대처하고 시스템 확장의 중요 시점을 더 잘 예약할 수 있습니다.
- **충분 배포.** 충분 배포는 자원 추가 예약에 도움이 될 수 있습니다. 시스템 확장에 대한 명확한 시점을 지정합니다. 시점은 일반적으로 확장성을 평가하는 특정 날짜로 조정된 로드 기반 요구 사항입니다.
- **광범위한 성능 모니터링.** 모니터링 성능은 시스템에 자원을 언제 추가할 지 결정할 수 있게 합니다. 모니터링 성능에 대한 요구 사항은 유지 보수 및 업그레이드를 책임진 운영자 및 관리자에게 지침을 제공할 수 있습니다.

다음 표에서는 확장성 요구 사항을 결정하는데 고려할 요소를 나열합니다.

표 3-5 확장성 요소

항목	설명
사용 패턴 분석	기존 데이터를 조사하여 현재 또는 예상된 사용자 기반의 사용 패턴을 이해합니다. 현재 데이터가 없을 경우 산업 데이터나 시장 예측을 분석합니다.
합리적인 최대 규모에 대한 설계	알려진 요구와 가능한 요구 모두에 대한 최대 필수 규모와 관련된 목표를 사용하여 설계합니다. 종종 이 규모는 기존 사용자 로드와 합리적으로 예상된 장애 로드와 대한 성능 평가를 기준으로 24개월에 대해 예측하는 양입니다. 예측 기간은 예상의 신뢰성에 따라 상당히 다릅니다.
적절한 중요 시점 설정	예상치 못한 증가를 수용하는 버퍼가 포함된 단기간 요구 사항을 충족시키기 위한 충분 배포 설계를 구현합니다. 시스템 자원을 추가할 중요 시점을 설정합니다. 예를 들면 다음과 같습니다. <ul style="list-style-type: none"> ■ 자본 취득(예:분기별 또는 연별) ■ 하드웨어 및 소프트웨어 구입 선행 시간(예:1주일에서 6주까지) ■ 버퍼(증가 예상에 따라 10% - 100%)
최신 기술 통합	최신 기술(예:더 빠른 프로세서와 웹서버)과 이 기술이 기본 구조의 성능에 어떻게 영향을 미칠 수 있는지 이해합니다.

보안 요구 사항

보안은 배포된 시스템의 모든 수준과 관련된 복잡한 사안입니다. 보안 요구 사항 개발은 보안 위협 식별 및 그 위협을 해결할 전략 개발을 고려합니다. 이 보안 분석은 다음과 같은 단계를 포함합니다.

1. 중요 자산 식별
2. 해당 자산에 대한 위협 식별
3. 조직에 대한 위기를 일으키는 위협을 드러내는 취약점 식별

4. 조직에 대한 위기를 완화시키는 보안 계획 개발

보안 요구 사항 분석은 관리자, 비즈니스 분석자 및 정보 기술 직원을 비롯한 모든 이해 관계자의 크로스섹션을 포함해야 합니다. 종종 어떤 조직에서는 보안 설계자를 정하여 보안 방법의 설계 및 구현에서 선두에 섭니다.

다음 절에서는 보안 계획의 일부 영역을 설명합니다.

보안 계획 요소

시스템 보안 계획은 구현 성공에 필수적인 배포 설계의 일부입니다. 보안 계획 시 다음을 고려하십시오.

- **물리적 보안.** 물리적 보안은 라우터, 서버, 서버실, 데이터 센터 및 기타 인프라의 부분에 대한 물리적 액세스입니다. 다른 보안 방법은 허가되지 않은 사람이 서버실로 들어가 라우터의 플러그를 뽑는 경우에 고려할 수 있습니다.
- **네트워크 보안.** 네트워크 보안은 방화벽, 보안 액세스 영역, 액세스 제어 목록 및 포트 액세스를 통해 네트워크에 액세스하는 것입니다. 네트워크 보안을 위해서 허가되지 않은 액세스, 변조 및 해커의 공격에 대한 거부에 대한 전략을 개발합니다.
- **응용 프로그램 및 응용 프로그램 데이터 보안.** 응용 프로그램 및 응용 프로그램 데이터 보안은 인증 및 권한 부여 절차와 정책을 통해 사용자 계정, 기업 데이터 및 엔터프라이즈 응용 프로그램을 모두 처리합니다. 이 영역에는 다음과 같은 정책에 대한 정의가 포함됩니다.
 - 비밀번호 정책
 - 액세스 권한(예: 관리자 액세스와 반대로 사용자에게 위임된 관리)
 - 계정 비활성화
 - 액세스 제어
 - 데이터의 보안 전송을 포함하며 데이터를 사인하기 위한 인증서를 사용하는 암호화 정책
- **개인 보안 사례.** 조작 전반 보안 정책은 다른 보안 조치가 설계된 대로 수행되도록 하기 위해 모든 사용자가 지켜야 하는 작업 환경 및 사례를 정의합니다. 일반적으로 보안에 대한 안내서 또는 설명서를 개발하며 보안 사례에 대해 사용자에게 교육을 제안하기도 합니다. 효과적인 전체 보안 정책을 위해서 견고한 보안 사례가 조직 문화의 일부가 되어야 합니다.

잠재 용량

잠재 용량은 자원을 추가하지 않고 비정상적인 최고 로드 사용을 처리할 수 있는 배포 기능입니다. 일반적으로 잠재 용량과 관련된 서비스 품질 요구 사항을 직접 지정하지는 않지만 이 시스템 품질은 시스템의 가용성, 성능 및 확장성 요구 사항의 한 요소입니다.

서비스 가능성 요구 사항

서비스 가능성은 시스템 모니터링, 발생한 문제 복구, 시스템에 사용자 추가 및 제거, 하드웨어 및 소프트웨어 구성 요소 업그레이드 등을 포함하여 배포된 시스템을 얼마나 쉽게 유지 보수할 수 있는가를 말합니다.

서비스 가능성 요구 사항을 계획할 때는 다음 표에 나열된 항목을 고려합니다.

표 3-6 서비스 가능성 요구 사항 항목

항목	설명
중단 시간 계획	<p>특정 서비스를 사용할 수 없게 하거나 부분적으로 사용할 수 없게 해야 하는 유지 보수 작업을 식별합니다.</p> <p>일부 유지 보수 및 업그레이드는 사용자 중단 없이 이루어지지만 서비스를 중단해야 하는 경우도 있습니다. 가능하면 사용자와 함께 중단 시간이 필요한 유지 보수 작업을 예약하여 사용자가 중단 시간을 대비할 수 있도록 합니다.</p>
사용 패턴	<p>유지 보수 일정을 예약하기 위한 최상의 시간을 결정하는 사용 패턴을 식별합니다.</p> <p>예를 들면 일반 업무 시간 중에 사용량이 최고인 시스템은 저녁이나 주말에 유지 보수 일정을 예약합니다. 지리적으로 분산된 시스템의 경우 이 시간을 식별하기가 더 어려울 수 있습니다.</p>
가용성	<p>서비스 가능성은 종종 가용성 설계를 반영합니다. 유지 보수 및 업그레이드를 위한 중단 시간을 최소화하기 위한 전략의 중요한 부분은 가용성 전략입니다. 높은 수준의 가용성을 요구하는 시스템에서는 유지 보수, 업그레이드 및 복구를 할 기회가 훨씬 적습니다.</p> <p>가용성 요구 사항을 처리하기 위한 전략은 유지 보수 및 업그레이드를 처리하는 방법에 영향을 미칩니다. 예를 들어 지리적으로 분산된 시스템의 경우 서비스 가능성은 유지 보수 기간 중에 작업 로드를 원격 서버에 라우트할 수 있는 기능에 달려있을 수 있습니다.</p> <p>또한 높은 수준의 가용성이 필요한 시스템에는 사용자 간섭 없이도 시스템을 자동으로 다시 시작하는 더 복잡한 솔루션이 필요할 수 있습니다.</p>
진단 및 모니터링	<p>정기적으로 진단 및 모니터링 도구를 실행하여 문제 영역을 식별하면 시스템의 안정성을 개선할 수 있습니다.</p> <p>시스템을 정규적으로 모니터링하면 문제가 발생하기 전에 방지하고 가용성 전략에 따라 작업 로드의 균형을 조정하며 유지 보수 및 중단 시간을 더 잘 계획할 수 있습니다.</p>

서비스 수준 요구 사항

서비스 수준 계약(SLA)은 최소 성능 요구 사항과 해당 요구 사항 달성 실패, 제공되어야 할 고객 지원의 수준 및 범위에 대해 지정합니다. 서비스 수준 계약은 SLA가 기반으로 하는 조건을 지정한 시스템 요구 사항입니다.

서비스 품질 요구 사항과 마찬가지로 서비스 수준 계약은 비즈니스 요구 사항에서 나오며 배포된 시스템이 달성해야 할 전반적인 시스템 품질에 대한 보증을 나타냅니다. 서비스 수준 계약은 계약으로 간주되기 때문에 서비스 수준 요구 사항의 사양은 모호하지 않아야 합니다. 서비스 수준 요구 사항에서는 요구 사항을 검사하는 기준과 요구 사항을 충족시키는 데 실패한 것으로 간주되는 경우 정확하게 정의해야 합니다.

논리적 설계

솔루션 라이프 사이클의 논리적 설계 단계 중에 솔루션의 논리적 구성 요소의 상호 관계를 보여주면서 논리적 구조를 설계합니다. 기술 요구 사항 단계의 논리적 구조 및 사용 분석은 배포 설계 단계로 입력될 배포 시나리오를 구성합니다.

이 장의 내용은 다음과 같습니다.

- 45 페이지 “논리적 구조 정보”
- 46 페이지 “논리적 구조 설계”
- 47 페이지 “Java Enterprise System 구성 요소”
- 54 페이지 “논리적 구조 예”
- 62 페이지 “액세스 영역”
- 64 페이지 “배포 시나리오”

논리적 구조 정보

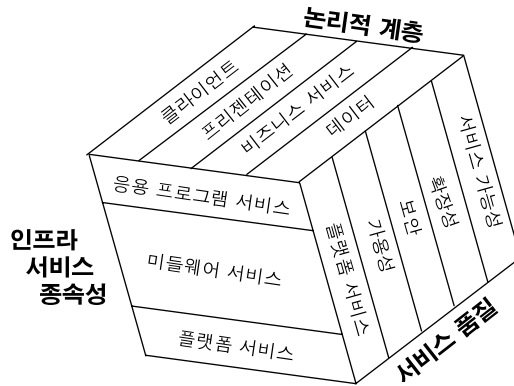
논리적 구조는 구성 요소 간 상호 관계를 보여주면서 솔루션 구현에 필요한 소프트웨어 구성 요소를 식별합니다. 기술 요구 사항 단계에서 정해진 논리적 구조 및 서비스 품질 요구 사항이 배포 시나리오를 구성합니다. 배포 시나리오는 다음 단계에서 발생하는 배포 설계, 즉 배포 구조를 설계하는 기본입니다.

논리적 구조를 개발하는 경우 사용자에게 서비스를 제공하는 구성 요소 뿐 아니라 필요한 미들웨어 및 플랫폼 서비스를 제공하는 기타 구성 요소를 식별해야 합니다. 인프라 서비스 종속성 및 논리적 계층은 이 분석을 수행하는 두 개의 보완적인 방법을 제공합니다.

인프라 서비스 종속성 및 논리적 계층은 Sun Java™ Enterprise System에서 기반으로 하는 솔루션 구조의 3개의 차원 중에서 2차원에 해당됩니다. 3개의 차원은 아래 나열되어 있으며 이는 45 페이지 “논리적 구조 정보”에도 나와 있습니다.

- **인프라 서비스 종속성.** 엔터프라이즈 서비스를 제공하는 상호 작용 소프트웨어 구성 요소입니다. 소프트웨어 구성 요소에는 분산 구성 요소가 서로 통신하고 상호 운영할 수 있게 하는 인프라 서비스의 기본 집합이 필요합니다.

- **논리적 계층.** 소프트웨어 구성 요소의 논리적 조직으로, 제공하는 서비스의 특성을 토대로 소프트웨어 구성 요소의 논리적 및 물리적 독립성을 나타내는 계층으로 되어 있습니다.
- **서비스 품질.** 성능, 가용성, 확장성 및 소프트웨어 솔루션의 설계와 운영의 특정 측면을 나타내는 기타 기능 등의 시스템 서비스 품질입니다.



주 - Java Enterprise System 구조 개념에 대한 자세한 내용은 **Sun Java Enterprise System 2005Q4 기술 개요**의 “Java Enterprise System 구조” 장을 참조하십시오.

논리적 구조는 필요한 구성 요소 및 해당 종속성을 나타냄으로써 인프라 서비스 수준을 묘사합니다. 논리적 구조는 또한 표현, 비즈니스 및 궁극적으로 클라이언트 계층이 액세스할 수 있는 데이터 서비스를 나타내는 논리적 계층 내에 구성 요소를 분산합니다. 서비스 품질 요구 사항은 논리적 구조에서 모델링되지 않지만 배포 시나리오에서 논리적 구조와 쌍을 이룹니다.

논리적 구조 설계

논리적 구조를 설계하는 경우 기술 요구 사항 단계 중에 식별된 사용 사례를 사용하여 솔루션에 필요한 서비스를 제공하는 Java Enterprise System 구성 요소를 결정합니다. 또한 초기에 식별한 구성 요소에 대한 서비스를 제공하는 모든 구성 요소를 식별합니다.

제공하는 서비스 유형에 따라 다중 계층 구조 컨텍스트 내에 Java Enterprise System 구성 요소를 배치합니다. 다중 계층 구조의 일부로 구성 요소를 이해하면 나중에 구성 요소가 제공한 서비스를 분산하는 방법을 결정할 수 있으며 확장성, 가용성 등과 같은 서비스 품질을 구현하기 위한 전략을 결정할 수 있습니다.)

또한 보안 액세스 영역 내에 있는 논리적 구성 요소에 대한 또 다른 시각을 제공합니다.
62 페이지 “액세스 영역” 절은 보안 액세스 영역의 예를 제공합니다.

Java Enterprise System 구성 요소

Java Enterprise System은 엔터프라이즈 서비스를 제공하는 상호 작용 소프트웨어로 이루어져 있으며 이 엔터프라이즈 서비스를 사용하여 엔터프라이즈 솔루션을 구축할 수 있습니다. 다음 그림은 Java Enterprise System에서 제공하는 주요 소프트웨어 구성 요소를 보여줍니다. **Sun Java Enterprise System 2005Q4 기술 개요**는 Java Enterprise System 구성 요소와 구성 요소가 제공하는 서비스에 대한 추가 정보를 제공합니다.

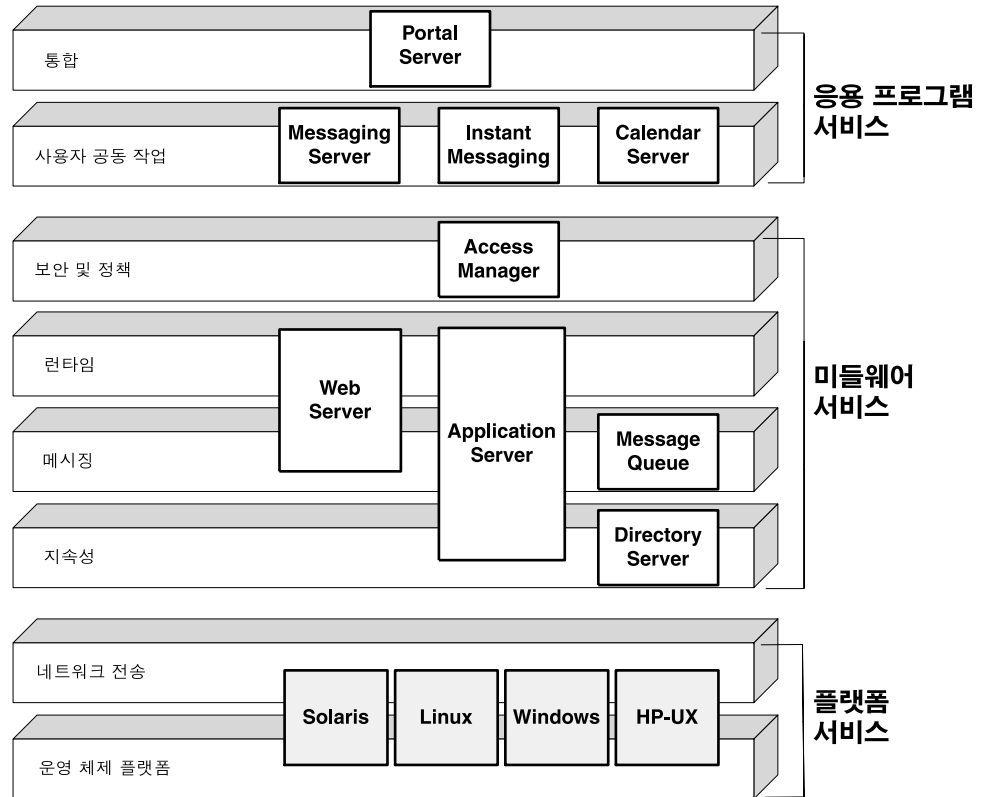


그림 4-1 Java Enterprise System 구성 요소

구성 요소 종속성

논리적 구조를 위해 Java Enterprise System 구성 요소를 식별하는 경우 지원 구성 요소 또한 식별해야 합니다. 예를 들면 논리적 구조에 필요한 구성 요소로 Messaging Server를 식별한 경우 논리적 구조는 Directory Server를 포함해야 하며 또한 Access Manager도 포함할 수 있습니다. Messaging Server는 디렉토리 서비스에 대해서는 Directory Server에, 단일 사인 온(SSO)이 필요한 솔루션에 대해서는 Access Manager에 의존합니다.

다음 표에는 Java Enterprise System 구성 요소의 종속성이 나열되어 있습니다. 주요 구성 요소 간 종속성에 대한 시각적 설명은 48 페이지 “구성 요소 종속성” 그림을 참조하십시오. 논리적 구조를 설계하는 경우 이 표와 그림을 사용하여 설계상 종속 구성 요소를 결정할 수 있습니다.

표 4-1 Java Enterprise System 구성 요소 종속성

Java Enterprise System 구성 요소	종속 대상
Application Server	Message QueueDirectory Server(선택 사항)
Calendar Server	Messaging Server(전자 메일 알림 서비스용)Access Manager(단일 사인 온(SSO)용)Web Server(웹 인터페이스용)Directory Server
Communications Express	Access Manager(단일 사인 온(SSO)용)Calendar ServerMessaging ServerInstant MessagingWeb Server(웹 인터페이스용)Directory Server
Directory Proxy Server	Directory Server
Directory Server	없음
Access Manager	Application Server 또는 Web ServerDirectory Server
Instant Messaging	Access Manager(단일 사인 온(SSO)용)Directory Server
Message Queue	Directory Server(선택 사항)
Messaging Server	Access Manager(단일 사인 온(SSO)용)Web Server(웹 인터페이스용)Directory Server
Portal Server	Portal Server 채널을 사용하도록 구성한 경우: Calendar ServerMessaging ServerInstant Messaging Access Manager(단일 사인 온(SSO)용)Application Server 또는 Web ServerDirectory Server
Portal Server Secure Remote Access	Portal Server
Web Server	Access Manager(선택 사항, 액세스 제어용)

주 - 48 페이지 “구성 요소 종속성”에 나열된 Java Enterprise System 구성 요소 간 종속성이 모든 구성 요소 종속성을 나열한 것은 아닙니다. 48 페이지 “구성 요소 종속성”은 설치를 계획할 때 고려해야 하는 종속성은 나열하지 않았습니다. Java Enterprise System 종속성의 전체 목록을 보려면 **Sun Java Enterprise System 2005Q4 UNIX용 설치 설명서**를 참조하십시오.

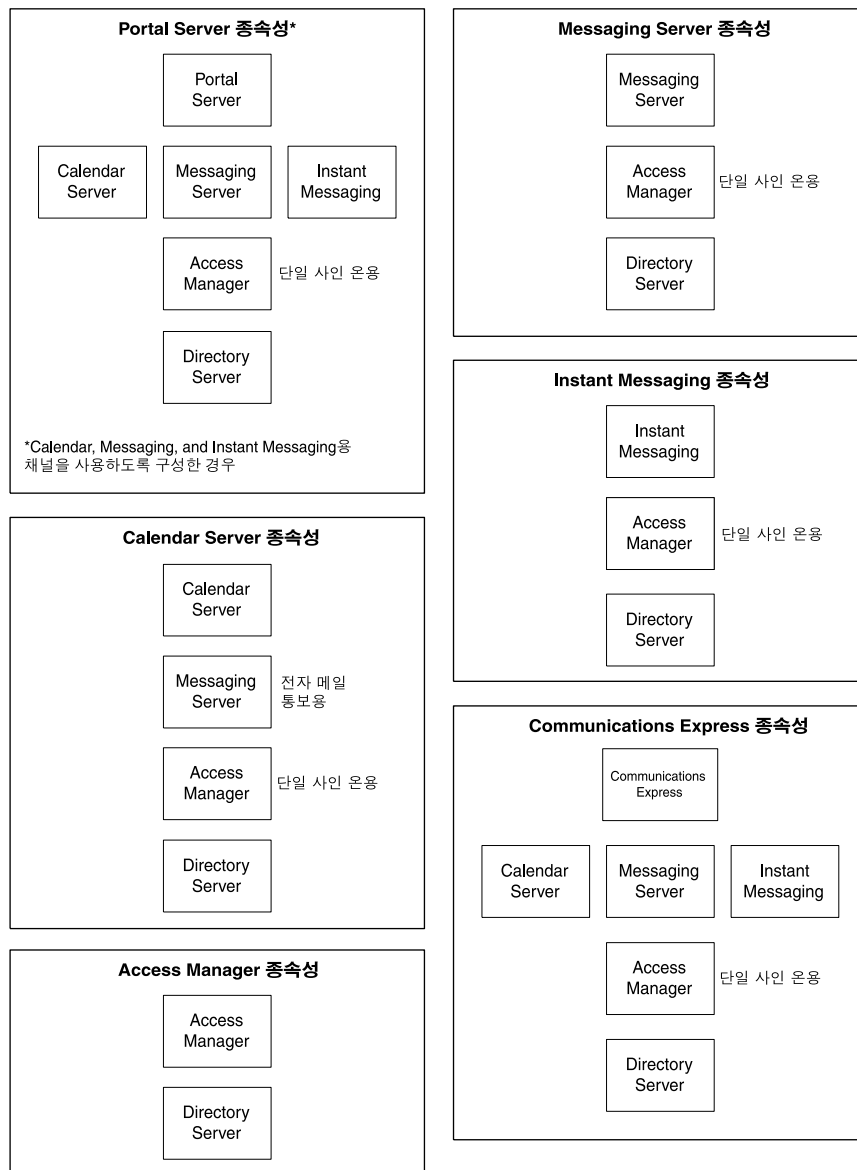


그림 4-2 Java Enterprise System 구성 요소 종속성

웹 컨테이너 지원

이전 절인 48 페이지 “구성 요소 종속성”에서는 Portal Server 및 Access Manager를 실행하는 웹 컨테이너를 고려하지 않습니다. Application Server, Web Server 또는 타사 제품에서 이 웹 컨테이너를 제공할 수 있습니다. Portal Server 또는 Access Manager를 포함하는 논리적 구조를 설계하는 경우 이러한 구성 요소에 필요한 웹 컨테이너를 고려해야 합니다.

Messaging Server에서 제공되는 논리적으로 별개인 서비스

Java Enterprise System Messaging Server는 다음과 같이 논리적으로 별개인 서비스를 제공하는 개별 인스턴스를 제공하도록 구성할 수 있습니다.

- 메시지 전송 에이전트
- Message Multiplexor
- Message Express Multiplexor
- 메시지 저장소

이러한 Messaging Server의 다양한 구성은 별개의 물리적 서버에 배포되고 논리적 구조에서 각기 다른 계층에 나타낼 수 있는 기능을 제공합니다. Messaging Server의 이러한 구성이 개별 계층에 있는 논리적으로 별개인 서비스를 나타내므로 논리적 구조를 설계할 때 그 구성을 논리적으로 별개인 구성 요소로서 고려하십시오.

54 페이지 “논리적 구조 예” 절은 논리적으로 별개인 구성 요소의 예를 제공합니다.

다음 표에서는 Messaging Server의 논리적으로 별개인 구성을 설명합니다.

표 4-2 Messaging Server 구성

하위 구성 요소	설명
메시지 전송 에이전트(MTA)	SMTP 연결을 처리하고 전자 메일 경로를 지정하며 올바른 메시지 저장소로 메시지를 전달함으로써 전자 메일 전송을 지원합니다. MTA 구성 요소를 기업 외부에서 전자 메일을 전달(인바운드)하거나 기업 내에서 전달(송신)하는 것을 지원하도록 구성할 수 있습니다.
메시지 저장소(STR)	전자 메일 메시지의 검색 및 저장을 제공합니다.
Message Multiplexor(MMP)	IMAP 또는 POP 프로토콜을 사용하여 전자 메일 클라이언트를 위한 메시지 저장소에 액세스하는 전자 메일 검색을 지원합니다.
Messenger Express Multiplexor(MEM)	웹 기반(HTTP) 클라이언트를 위해 메시지 저장소에 액세스하는 전자 메일 검색을 지원합니다.

액세스 구성 요소

Java Enterprise System은 또한 종종 엔터프라이즈 방화벽 외부로부터 시스템 서비스에 액세스할 수 있는 구성 요소도 포함합니다. Messaging Server의 일부 구성으로 네트워크 액세스를 할 수 있습니다(예: Message Multiplexor를 위해 Messaging Server 구성)다음 표에서는 시스템 서비스에 원격으로 액세스할 수 있는 Java Enterprise System 구성 요소를 설명합니다.

표 4-3 원격 액세스를 제공하는 Java Enterprise System 구성 요소

구성 요소	설명
Directory Proxy Server	여러 Directory Server 인스턴스에 대해 향상된 디렉토리 액세스, 스키마 호환성, 경로 지정 및 로드 균형 조정을 제공합니다.
Portal Server, Portal Server Secure Remote Access	내부 포털 및 인터넷 응용 프로그램을 포함한 Portal Server 콘텐츠와 서비스에 대해 회사 방화벽 외부에서의 보안 인터넷 액세스를 제공합니다.
Portal Server, Portal Server Mobile Access	모바일 장치로부터의 무선 액세스 및 Portal Server로의 음성 액세스를 제공합니다.
Messaging Server Message Multiplexor(MMP)	웹 기반(HTTP) 클라이언트를 위해 메시지 저장소에 액세스하는 전자 메일 검색을 지원합니다.

원격 액세스를 제공하는 구성 요소는 62 페이지 “액세스 영역” 절의 예에서 설명한 대로 일반적으로 보안 액세스 영역에 배포됩니다.

다중 계층 구조 설계

Java Enterprise System은 서비스가 제공하는 기능에 따라 서비스를 계층에 배치하는 다중 계층 구조 설계에 적합합니다. 각 서비스는 논리적으로 독립적이며 같은 계층 또는 다른 계층의 서비스가 액세스할 수 있습니다. 다음 그림은 클라이언트, 표현, 비즈니스 서비스 및 데이터 계층을 보여주면서 엔터프라이즈 응용 프로그램에 대한 다중 계층 구조 모델을 보여줍니다.

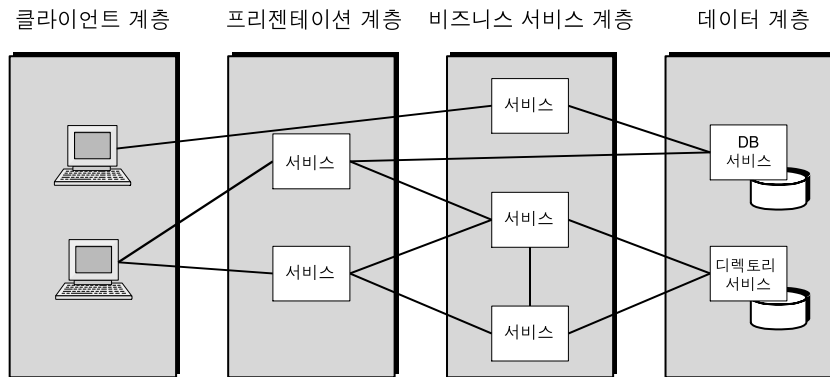


그림 4-3 다중 계층 구조 모델

다음 표에서는 52 페이지 “다중 계층 구조 설계”에서 묘사한 논리적 계층을 설명합니다.

표 4-4 다중 계층 구조의 논리적 계층

계층	설명
클라이언트 계층	최종 사용자에게 정보를 설명하는 클라이언트 응용 프로그램을 포함합니다. Java Enterprise System의 경우 응용 프로그램은 일반적으로 메일 클라이언트, 웹 브라우저 또는 Mobile Access 클라이언트입니다.
프리젠테이션 계층	사용자가 표현을 처리하고 조작할 수 있도록 최종 사용자에게 데이터를 표시하는 서비스를 제공합니다. 예를 들어 웹 메일 클라이언트 또는 Portal Server 구성 요소를 사용하여 사용자는 수신한 정보의 표현을 수정할 수 있습니다.
비즈니스 서비스 계층	표현 계층 또는 비즈니스 서비스 계층 내의 다른 서비스에게 또는 클라이언트 계층의 클라이언트에게 직접 제공하기 위한 데이터 계층으로부터 일반적으로 데이터를 검색하는 백엔드 서비스를 제공합니다. 예를 들면 Access Manager는 다른 Java Enterprise System 구성 요소에 Identity 서비스를 제공합니다.
데이터 계층	표현 계층 또는 비즈니스 서비스 계층 내의 서비스가 액세스하는 데이터베이스 서비스를 제공합니다. 예를 들면 Directory Server는 다른 서비스에 LDAP 디렉토리 액세스를 제공합니다.

다중 계층 구조 설계는 여러가지 이점을 제공합니다. 배포 설계 단계 중에 다중 계층 구조의 기능에 따른 서비스 배치는 네트워크에서 서비스를 분산하는 방법을 관별할 수 있게 합니다. 또한 구조 내의 구성 요소가 다른 구성 요소의 서비스에 어떻게 액세스하는 지 볼 수 있습니다. 이렇게 시각적으로 표현하면 가용성, 확장성, 보안 및 다른 서비스 품질 솔루션을 계획할 수 있습니다.

논리적 구조 예

이 절에서는 Java Enterprise System 솔루션을 위한 논리적 구조의 예를 제공합니다. 이 예는 논리적 구성 요소를 다중 계층 구조의 알맞은 계층에 배치하는 방법을 보여준 다음 사용 사례를 연구하여 구성 요소 간 관계를 분석합니다. 이 절의 논리적 구조 예를 Java Enterprise System 솔루션의 논리적 구조 설계를 이해하는 기초로 사용하십시오.

첫 번째 예는 기본 Messaging Server 솔루션으로 Messaging Server의 논리적으로 별개인 구성 요소가 다른 구성 요소와 상호 작용하는 방법을 보여줍니다. 두 번째 예는 약 1,000명에서 5,000명 정도의 직원이 있는 중소기업에게 적합한 신원 기반 배포 솔루션의 논리적 구조를 보여줍니다.

Messaging Server 예

다음 그림은 Messaging Server 배포의 기본 논리적 구조를 보여줍니다. 이 논리적 구조는 Messaging Server에 필요한 논리적으로 별개인 구성 요소만을 보여줍니다. 이 다음 그림에서 구성 요소 간 관계를 설명합니다.

주 - 일반적으로 Messaging Server의 배포는 엔터프라이즈 솔루션의 일부로 59 페이지 “신원 기반 통신 예”에서 설명한 대로 다른 Java Enterprise System 구성 요소를 포함합니다.

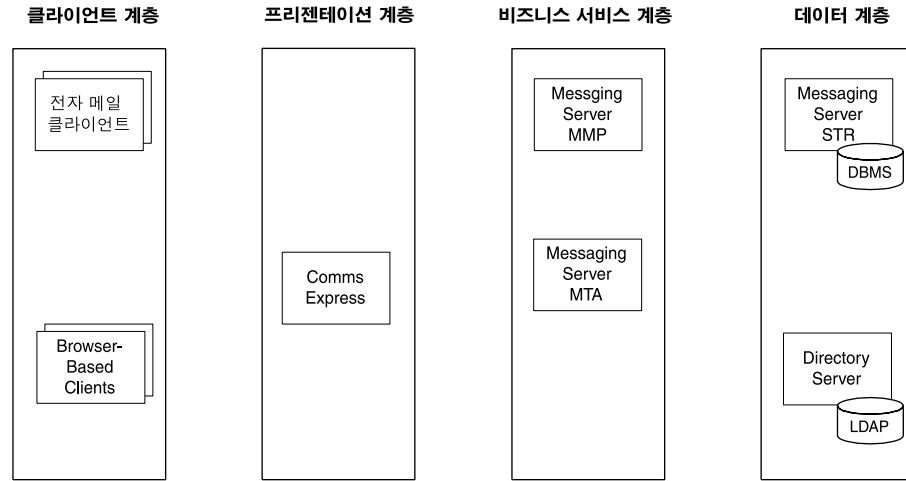


그림 4-4 Messaging Server 배포의 논리적 구조

다음 표에서는 54 페이지 “Messaging Server 예”에서 묘사한 구성 요소를 설명합니다.

표 4-5 Messaging Server 논리적 구조의 구성 요소

구성 요소	설명
전자 메일 클라이언트	전자 메일 읽기 및 쓰기를 위한 클라이언트 응용 프로그램
Messaging Server MTA	전자 메일 메시지를 수신, 경로 지정, 전송 및 전달하기 위한 메시지 전송 에이전트(MTA)로 구성된 Messaging Server.
Messaging Server MMP	검색 및 저장을 위해 알맞은 메시지 저장소로 연결 경로를 지정하는 Message Multiplexor(MMP)로 구성된 Messaging Server. MMP는 Directory Server에 액세스하여 알맞은 메시지 저장소를 결정하는 디렉토리 정보를 조회합니다.
Messaging Server STR	전자 메일 메시지 검색 및 저장을 위한 메시지 저장소로 구성된 Messaging Server.
Directory Server	LDAP 디렉토리 데이터로 액세스를 제공합니다.

논리적 구조가 Messaging Server 구성 요소에 대한 서비스 복제를 지정하지는 않습니다. 예를 들어 엔터프라이즈 배포는 일반적으로 별개의 인바운드 및 아웃바운드 MTA 인스턴스를 여러 개 만들지만 54 페이지 “Messaging Server 예”의 그림에서는 한 개의 MAT 구성 요소만을 보여줍니다. 다중 인스턴스로 논리적 구성 요소를 복제하는 것은 배포 설계 단계 중에 해야 하는 설계 결정입니다.

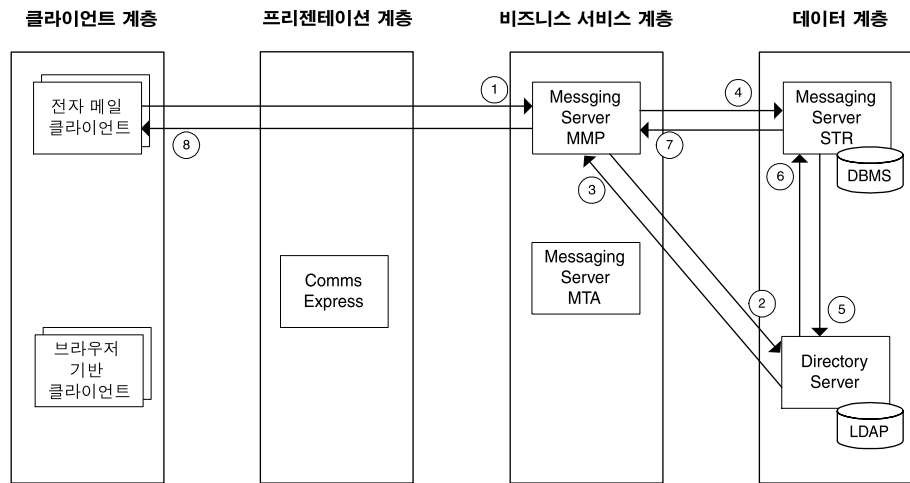
Messaging Server 사용 사례

사용 사례는 구조내 논리적 구성 요소 간 관계를 식별할 수 있게 해줍니다. 사용 사례에 따른 구성 요소 간 상호 작용을 매핑하여 배포 설계에 도움이 되는 구성 요소 상호 작용을 시각화할 수 있습니다.

일반적으로 각 사용 사례를 분석하여 배포 설계 전에 구성 요소의 상호 작용을 관별합니다. 다음은 Messaging Server에 전형적인 세 개의 사용 사례로, 논리적 구성 요소 간 상호 작용을 보여줍니다.

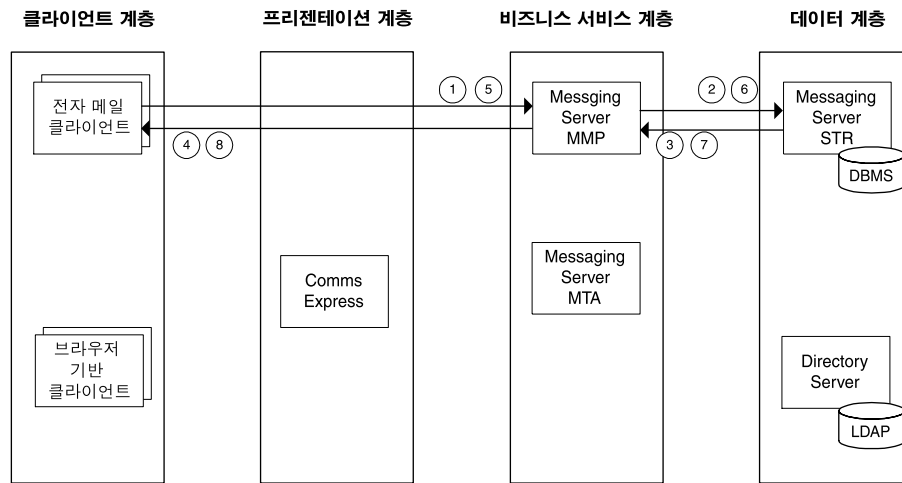
▼ 사용 사례 1: 사용자가 Messaging Server에 성공적으로 로그인합니다

- 단계
1. 전자 메일 클라이언트가 로그인 정보를 Messaging Server Multiplexor(MMP)로 보냅니다.
 2. MMP가 Directory Server에서 사용자 아이디 및 비밀번호 검증을 요청합니다.
 3. Directory Server가 검증을 MMP로 반송합니다.
 4. MMP가 Messaging Server 메시지 저장소(STR)로부터 메시지 목록을 요청합니다.
 5. STR이 Directory Server에서 사용자의 LDAP 레코드를 요청합니다.
 6. Directory Server가 사용자의 LDAP 레코드를 STR로 반송합니다.
 7. STR이 MMP로 메시지 목록을 반송합니다.
 8. MMP가 메시지 목록을 전자 메일 클라이언트에게 전달합니다.



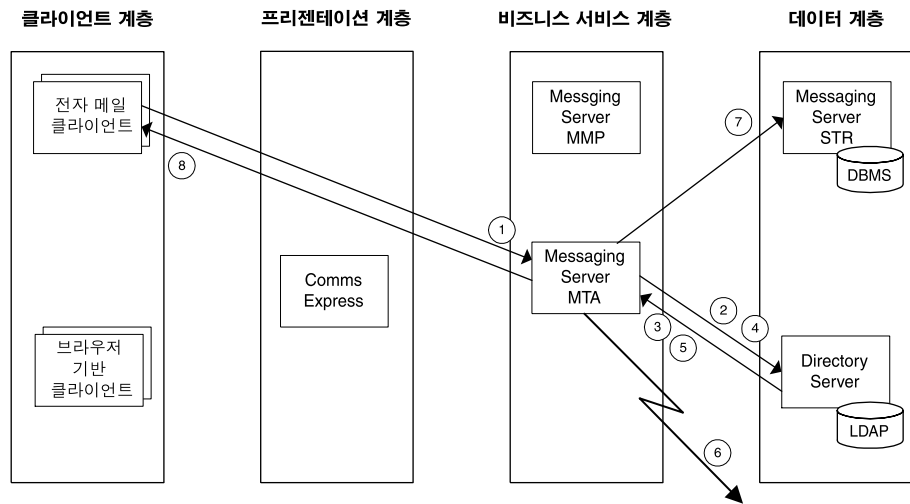
▼ 사용 사례 2: 로그인한 사용자가 메일을 읽고 삭제합니다.

- 단계
1. 전자 메일 클라이언트가 Messaging Server Multiplexor(MMP)에서 읽을 메시지를 요청합니다.
 2. MMP가 Messaging Server 메시지 저장소(STR)로부터 메시지를 요청합니다.
 3. STR이 MMP로 메시지를 반송합니다.
 4. MMP가 메시지를 전자 메일 클라이언트에게 전달합니다.
 5. 전자 메일 클라이언트가 메시지 삭제 작업을 MMP로 보냅니다.
 6. MMP가 메시지 삭제 작업을 STR로 전달합니다.
 7. STR이 메시지를 데이터베이스에서 삭제하고 MMP로 확인을 보냅니다.
 8. MMP가 삭제 확인을 전자 메일 클라이언트에게 전달합니다.



▼ **사용 사례 3: 로그인한 사용자가 전자 메일 메시지를 보냅니다.**

- 단계
1. 전자 메일 클라이언트가 클라이언트에서 작성된 메시지를 Messaging Server 메시지 전송 에이전트(MTA)로 보냅니다.
 2. MTA가 Directory Server에서 사용자 아이디 및 비밀번호 검증을 요청합니다.
 3. Directory Server가 검증을 MTA로 반송합니다.
 4. MTA가 각 수신자의 대상 도메인에 대해 Directory Server를 확인합니다.
 5. Directory Server가 각 수신자의 대상 도메인을 MTA로 반송합니다.
 6. MTA가 각 수신자에게 메시지를 전달합니다.
 7. MTA가 메시지를 Messaging Server 메시지 저장소(STR)에 전달하여 보낸 메일함에 메시지를 저장합니다.
 8. MTA가 전자 메일 클라이언트에게 확인을 보냅니다.



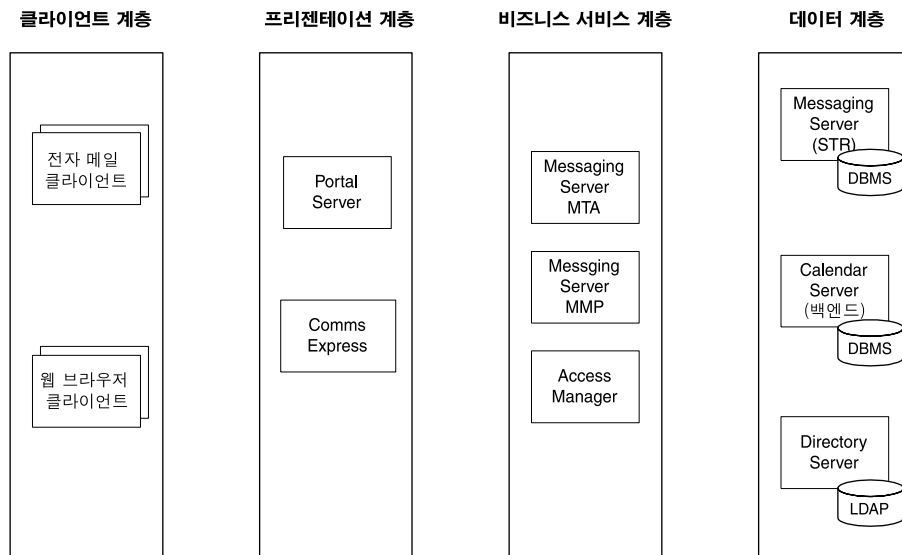
신원 기반 통신 예

이 예에서는 1,000명에서 5,000명의 직원이 있는 중소기업을 위한 신원 기반 통신 솔루션을 설명합니다. 일반적으로 자세한 기술 요구 사항 분석에 이어지는 철저한 비즈니스 분석이 논리적 구조 설계에 필요합니다. 그러나 이것은 이론적인 예로 다음과 같은 비즈니스 요구 사항을 판별했다고 가정합니다.

- 기업의 직원은 내부 웹 사이트, 통신 서비스, 달력 서비스 및 기타 자원에 사용자 정의된 액세스를 해야 합니다.
- 엔터프라이즈 차원의 인증 및 권한 부여가 내부 웹 사이트 및 기타 서비스에 액세스할 수 있게 합니다.
- 내부 웹 사이트 및 기타 서비스에 액세스할 수 있는 단일 사인 온을 사용할 수 있게 하여 전체 엔터프라이즈 서비스에 걸쳐 단일 아이디를 사용합니다.

이 예의 사용 사례는 로그인 절차, 전자 메일 읽기, 전자 메일 보내기, 포털의 사용자 정의, 달력 동기화 및 기타 유사한 사용자 작업을 자세히 설명합니다.

다음 그림은 이 유형의 신원 기반 통신 솔루션에 대한 논리적 구조를 보여줍니다.



신원 기반 통신에 대한 사용 사례 예

이러한 특성을 가진 배포 솔루션에는 일반적으로 솔루션이 제공하는 서비스에 대한 사용자 상호 작용을 설명하는 자세한 사용 사례가 있습니다. 이 예는 사용자가 웹 브라우저 클라이언트에서 포털로 로그인하는 경우 구성 요소 간의 상호 작용에 대해 집중합니다. 예에서 이 로그인 시나리오를 다음과 같이 두 개의 사용 사례로 나눕니다.

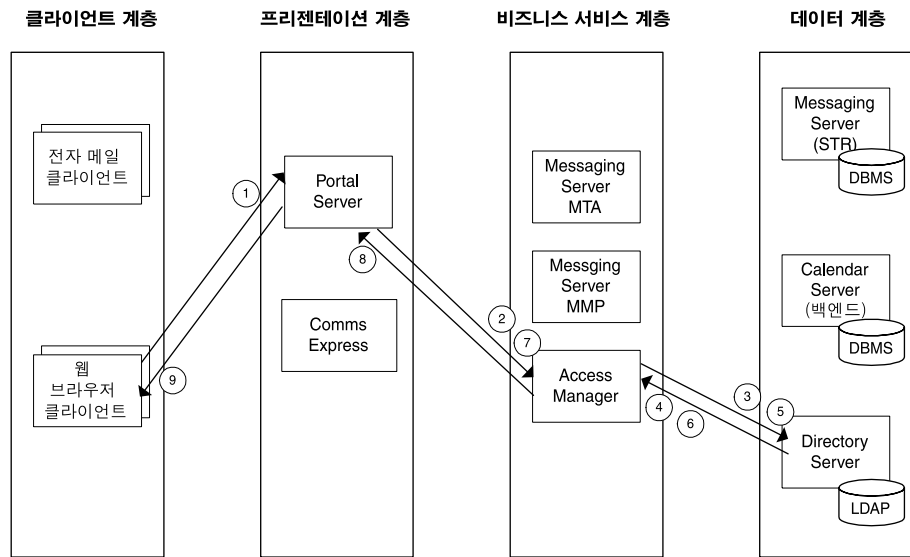
- 사용자가 로그인하고 인증을 받고 나서 Portal Server가 사용자의 포털 구성을 검색합니다.
- Portal Server가 전자 메일 및 달력 정보를 검색하여 웹 클라이언트에 표시합니다.

두 사용 사례를 한 개의 확장된 사용 사례로 간주할 수 있습니다. 그러나 이 예의 경우 단순화하기 위해 사용 사례를 분리합니다.

▼ 사용 사례 1: 사용자가 성공적으로 로그인하고 Portal이 사용자의 구성을 검색합니다

- 단계
1. 웹 브라우저 클라이언트가 사용자 아이디 및 비밀번호를 Portal Server로 보냅니다.
 2. Portal Server가 Access Manager에서 인증을 요청합니다.
 3. Access Manager가 Directory Server에서 사용자 아이디 및 비밀번호 검증 요청합니다.
 4. Directory Server가 사용자 아이디 및 비밀번호를 검증합니다.

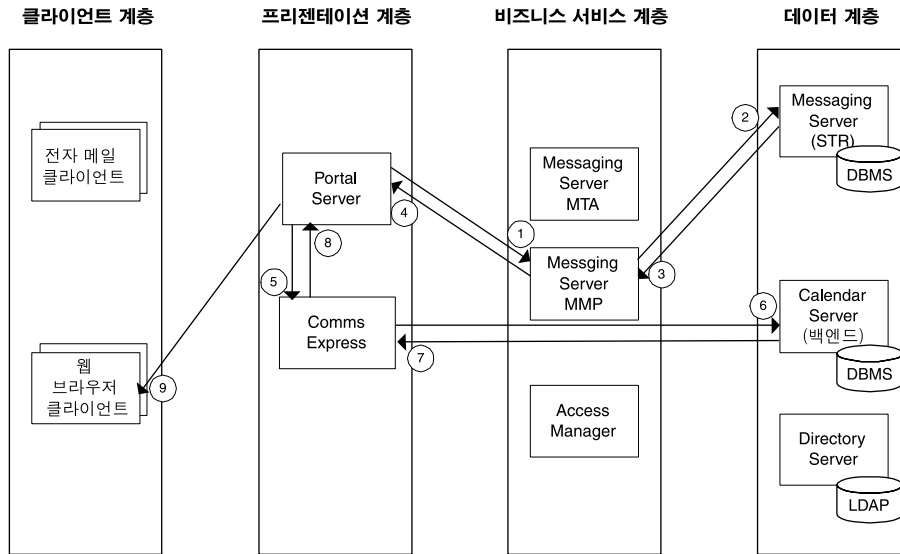
5. Access Manager가 Directory Server에서 사용자 프로필을 요청합니다.
6. Directory Server가 사용자 프로필을 반송합니다.
7. Portal Server가 Access Manager에서 사용자 디스플레이 프로필을 요청합니다.
8. Access Manager가 포털 구성을 반송합니다.
9. 포털 구성이 웹 브라우저 클라이언트에 표시됩니다.



▼ 사용 사례 2: Portal Server가 전자 메일 및 달력 정보를 표시합니다.

- 단계
1. 성공적으로 로그인하고 인증 및 포털 구성 검색을 하고 나면 Portal Server가 Messaging Server MMP에서 전자 메일 메시지를 요청합니다.
 2. MMP가 Messaging Server로부터 메시지 목록을 요청합니다.
 3. STR이 MMP로 메시지 목록을 반송합니다.
 4. MMP가 메시지 헤더를 Portal Server로 전달합니다.
 5. Portal Server가 Communications Express에서 달력 정보를 요청합니다.
 6. Communications Express가 Calendar Server 백엔드에서 달력 정보를 요청합니다.
 7. Calendar Server 백엔드가 Communications Express로 달력 정보를 반송합니다.

8. Communications Express가 달력 정보를 Portal Server로 전달합니다.
9. Portal Server가 모든 채널 정보를 웹 브라우저 클라이언트로 보냅니다.



액세스 영역

논리적 구조의 구성 요소를 나타내는 또 다른 방법은 구조가 보안 액세스를 제공하는 방법을 보여주는 액세스 영역을 사용하는 것입니다. 다음 그림은 Java Enterprise System 구성 요소 배포를 위한 액세스 영역을 보여줍니다. 각 액세스 영역은 인터넷 및 인트라넷에 구성 요소가 Secure Remote Access를 제공하는 방법을 보여줍니다.

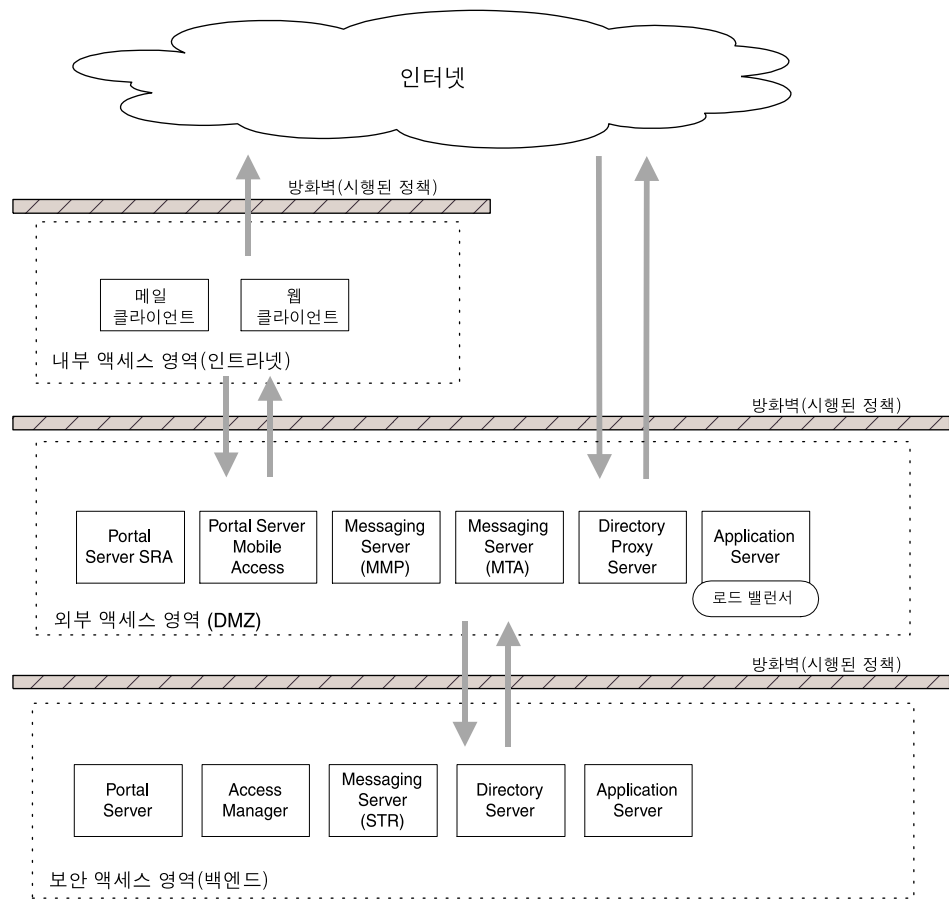


그림 4-5 액세스 영역에 있는 논리적 구조

다음 표에서는 62 페이지 “액세스 영역”에서 묘사한 액세스 영역을 설명합니다.

표 4-6 보안 액세스 영역 및 영역 내에 있는 구성 요소

액세스 영역	설명
내부 액세스 영역(인트라넷)	인트라넷 및 인터넷 간 방화벽에 의해 강화된 정책을 통해 인터넷에 액세스합니다. 내부 액세스 영역은 일반적으로 최종 사용자가 웹을 찾아보거나 전자 메일을 보내는데 사용합니다. 일부 경우에는 웹 찾아보기를 허용하는 인터넷에 직접 액세스합니다. 그러나 일반적으로 외부 액세스 영역을 통해 제공되는 인터넷에 보안 액세스를 합니다.
외부 액세스 영역(DMZ)	중요한 백엔드 서비스에 대한 보안 버퍼로 작용하면서 인터넷에 보안 액세스를 제공합니다.

표 4-6 보안 액세스 영역 및 영역 내에 있는 구성 요소 (계속)

액세스 영역	설명
보안 액세스 영역(백엔드)	중요한 백엔드 서비스로의 제한된 액세스를 제공하여 외부 액세스 영역에서만 액세스할 수 있습니다.

62 페이지 “액세스 영역”에서는 이전 예에서 묘사한 논리적 계층을 보여주지는 않지만 대신 원격 및 내부 액세스를 제공하는 구성 요소, 방화벽과 같은 보안 방법에 대한 구성 요소 관계 및 강화해야 할 액세스 규칙에 대한 시각적 묘사에 초점을 맞추었습니다. 액세스 영역을 보여주는 설계와 결합된 다중 계층 구조 설계를 사용하여 계획된 배포의 논리적 모델을 제공합니다.

배포 시나리오

완료된 논리적 구조 설계 자체만으로는 솔루션 라이프 사이클의 배포 설계 단계로 이동하기에 충분하지 않습니다. 논리적 구조를 기술 요구 사항 단계 중에 결정된 서비스 품질(QoS)과 쌍으로 만들어야 합니다. 서비스 품질 요구 사항을 논리적 구조와 쌍으로 만들어 배포 시나리오를 구성합니다. 배포 시나리오는 5장에서 설명하는 배포 구조 설계의 출발점입니다.

배포 설계

솔루션 라이프 사이클의 배포 설계 단계 중에 높은 수준의 배포 구조와 낮은 수준의 구현 사양을 설계하고 솔루션을 구현하는데 필요한 일련의 계획과 사양을 준비합니다. 배포 설계 단계에서 프로젝트 승인이 발생합니다.

이 장의 내용은 다음과 같습니다.

- 65 페이지 “배포 설계 정보”
- 68 페이지 “배포 설계 방법”
- 68 페이지 “프로세서 요구 사항 예상”
- 73 페이지 “보안 트랜잭션을 위한 프로세서 요구 사항 예상”
- 76 페이지 “가용성 전략 결정”
- 83 페이지 “확장성에 대한 전략 결정”
- 85 페이지 “성능 병목 현상 식별”
- 87 페이지 “최적의 자원 사용을 위한 설계”
- 88 페이지 “위기 관리”
- 89 페이지 “예제 배포 구조”

배포 설계 정보

배포 설계는 솔루션 라이프 사이클의 논리적 설계와 기술적 요구 사항 단계 중에 생성된 배포 시나리오로 시작합니다. 배포 시나리오에는 솔루션의 논리적 구조와 서비스 품질(QoS) 요구 사항이 포함됩니다. 논리적 구조에서 식별된 구성 요소를 물리적 서버와 기타 네트워크 장치에 매핑하여 배포 구조를 생성합니다. 서비스 품질 요구 사항은 성능과 가용성, 확장성 및 기타 관련 서비스 품질 사양의 하드웨어 구성에 대한 지침을 제공합니다.

배포 구조를 설계하는 것은 반복 프로세스입니다. 일반적으로 서비스 품질 요구 사항을 다시 확인하고 예비 설계를 재점검합니다. 균형 조정 및 소유 비용 문제의 균형을 유지하면서 서비스 품질 요구 사항의 상호 관계를 고려해야 궁극적으로 프로젝트의 비즈니스 목표를 달성할 수 있습니다.

프로젝트 승인

배포 설계 단계 중에 일반적으로 배포 구조를 생성한 이후에 프로젝트 승인이 발생합니다. 배포 구조 및 아래 설명된 구현 사양을 사용하여 배포의 실제 비용을 추정하고 이해 관계자의 승인을 위해 제출합니다. 프로젝트가 승인되면 배포 완료에 위한 계약에 서명하고 프로젝트를 구현하기 위한 자원을 취득하고 할당합니다.

배포 설계 출력

배포 설계 단계 중에 다음과 같은 사양 및 계획을 준비해야 합니다.

- **배포 구조.** 논리적 구조를 물리적 환경에 매핑하는 높은 수준의 구조입니다. 물리적 환경에는 인트라넷이나 인터넷 환경의 컴퓨팅 노드, 프로세서, 메모리, 저장 장치 및 기타 하드웨어 및 네트워크 장치가 포함됩니다.
- **구현 사양.** 배포를 구축하기 위한 청사진으로 사용된 세부 사양입니다. 이 사양은 구입할 컴퓨터와 네트워크 하드웨어에 대한 명세를 제공하며 배포를 위한 네트워크 레이아웃을 설명합니다. 또한 구현 사양에는 디렉토리 정보 트리(DIT)와 디렉토리 액세스를 위해 정의된 그룹과 역할에 대한 세부 정보를 비롯하여 디렉토리 서비스 사양이 포함됩니다.
- **구현 계획.** 엔터프라이즈 소프트웨어 솔루션을 구현하는 다양한 요소를 포함하는 계획 그룹입니다. 구현 계획은 다음과 같습니다.
 - **마이그레이션 계획.** 엔터프라이즈 데이터를 마이그레이션하고 엔터프라이즈 소프트웨어를 업그레이드하기 위한 전략 및 과정을 설명합니다. 마이그레이션한 데이터는 새로 설치한 엔터프라이즈 응용 프로그램의 형식과 표준을 따라야 합니다. 모든 엔터프라이즈 소프트웨어는 상호 운영을 위한 올바른 릴리스 버전이어야 합니다.
 - **설치 계획.** 배포 구조에 기초하여 하드웨어 서버 이름과 설치 디렉토리, 설치 순서, 각 노드 당 설치 유형, 설치 및 분산 배포 구성에 필요한 구성 정보를 지정합니다.
 - **사용자 관리 계획.** 기존 디렉토리 및 데이터베이스에 있는 데이터를 위한 마이그레이션 전략과 배포 구조에 지정된 복제 설계를 고려한 디렉토리 설계 사양, 새로운 내용이 있는 디렉토리를 관리하기 위한 절차를 포함합니다.
 - **테스트 계획.** 프로토타입 및 파일럿 구현 개발을 위한 특정 계획, 프로젝트 로드를 처리할 수 있는 능력을 판별하는 스트레스 테스트, 계획한 기능이 예상대로 작동하는지 판별하는 기능 테스트를 포함하여 배포된 소프트웨어를 테스트하기 위한 절차를 설명합니다.
 - **롤아웃 계획.** 계획 및 테스트 환경에서 작업 환경으로 구현을 이동하기 위한 절차와 일정을 설명합니다. 작업 환경으로 구현을 이동하는 것은 다양한 단계에서 발생합니다. 예를 들면 첫 단계에서 소프트웨어를 제한된 사용자 그룹에 배포하고 전체 배포를 완료할 때까지 각 단계에서 사용자 기반을 증가시킬 수 있습니다. 또한 단계적인 구현에 전체 배포를 완료할 때까지 특정 소프트웨어 패키지의 예정된 구현이 포함될 수 있습니다.
 - **재난 복구 계획.** 예기치 못한 시스템 전반의 장애로부터 시스템을 복구하는 방법에 대한 절차를 설명합니다. 복구 계획은 넓은 범위나 적은 범위의 장애 모두에 대한 절차를 포함합니다.

- **운영 계획(실행서).**모니터링과 유지 보수, 설치 및 업그레이드 절차를 설명하는 운영 설명서입니다.
- **교육 계획.**새로 설치한 엔터프라이즈 소프트웨어를 사용하는 운영자와 관리자, 최종 사용자를 교육하기 위한 과정과 절차를 포함합니다.

배포 설계에 영향을 미치는 요소

일부 요소는 배포 설계 동안 결정에 영향을 줍니다. 고려해야 할 주요 요소는 다음과 같습니다.

- **논리적 구조.**논리적 구조는 제안된 솔루션의 기능적 서비스와 해당 서비스를 제공하는 구성 요소의 상호 관계를 설명합니다. 논리적 구조를 서비스를 배포하는 최상의 방법을 판별하는 열쇠로 사용하십시오. 배포 시나리오에는 아래 설명된 대로 서비스 품질 요구 사항과 쌍을 이루는 논리적 구조가 포함됩니다.
- **서비스 품질 요구 사항.**서비스 품질(QoS) 요구 사항은 솔루션 운영의 다양한 요소를 지정합니다. 서비스 품질 요구 사항을 사용하면 성능, 가용성, 확장성, 서비스 가능성 및 기타 서비스 목표를 달성할 수 있는 전략을 개발할 수 있습니다. 배포 시나리오에는 이미 설명한 대로 서비스 품질 요구 사항과 쌍을 이루는 논리적 구조를 포함합니다.
- **사용 분석.**솔루션 라이프 사이클의 기술적 요구 사항 단계 중에 개발된 사용 분석은 배포 시스템의 로드와 스트레스를 예측할 수 있는 사용 패턴에 대한 정보를 제공합니다. 사용 분석을 사용하면 성능 병목 현상을 분리하고 서비스 품질 요구 사항을 충족시킬 전략을 개발할 수 있습니다.
- **사용 사례.**사용 사례는 솔루션 라이프 사이클의 기술적 요구 사항 단계 중에 개발되며 배포에 사용할 수 있도록 알려진 특정 사용자 상호 작용을 나열하며 가장 일반적인 사용 사례를 보여줍니다. 사용 사례는 사용 분석에 포함되어 있지만 배포 설계를 평가할 때 사용 사례를 참조하여 제대로 처리되는지 확인해야 합니다.
- **서비스 수준 계약.**서비스 수준 계약(SLA)은 최소 성능 요구 사항 및 해당 요구 사항이 충족되지 않는 경우 제공해야 할 고객 지원 수준 및 범위를 지정합니다. 배포 설계는 서비스 수준 계약에 지정된 성능 요구 사항을 손쉽게 충족해야 합니다.
- **총 소유 비용.**배포 설계 중에는 가용성과 성능, 확장성 등에 대한 서비스 품질 요구 사항을 처리할 잠재적인 솔루션을 분석하게 됩니다. 그러나 고려하는 각 솔루션에 대해 해당 솔루션 비용 및 그 비용이 총 소유 비용에 어떤 영향을 주는지도 고려해야 합니다. 결정으로 인한 균형 조정을 고려하고 자원을 최적화하도록 확인하여 비즈니스 제약 조건 하에서 비즈니스 요구 사항을 달성합니다.
- **비즈니스 목표.**비즈니스 목표는 솔루션 라이프 사이클의 비즈니스 분석 단계 중에 기술되며 해당 목표를 충족하는 비즈니스 요구 사항과 비즈니스 제약 조건이 포함됩니다. 배포 설계는 궁극적으로 비즈니스 목표를 충족시킬 능력에 의해 평가됩니다.

배포 설계 방법

배포 계획의 다른 요소와 마찬가지로 배포 설계는 과학일 뿐만 아니라 예술이며 특정 절차와 과정으로는 모두 설명될 수는 없습니다. 성공적인 배포 설계에 기여하는 요소는 과거 설계 경험과 시스템 구조에 대한 지식, 도메인 지식, 응용된 창조적 생각입니다.

배포 설계는 일반적으로 서비스 품질 요구 사항을 충족함과 동시에 성능 요구 사항을 달성하는 것이 중요합니다. 사용하는 전략은 솔루션을 최적화할 수 있도록 설계 결정의 균형 조정을 유지해야 합니다. 사용하는 방법은 일반적으로 다음과 같은 작업을 포함합니다.

- **프로세서 요구 사항 예상.** 배포 설계는 종종 논리적 구조의 각 구성 요소에 대해 필요한 CPU 수를 예상하는 것으로 시작합니다. 가장 과중한 로드를 나타내는 사용 사례로 시작하여 각 사용 사례 전체를 계속합니다. 사용 사례에 대한 지원을 제공하는 모든 구성 요소의 로드를 고려하고 그에 따라 예상 개수를 수정합니다. 또한 엔터프라이즈 시스템을 설계하면서 가졌던 모든 이전 경험을 고려합니다.
- **보안 트랜잭션을 위한 프로세서 요구 사항 예상.** 보안 전송에 필요한 사용 사례를 연구하고 그에 따른 CPU 예상 개수를 수정합니다.
- **가용성 및 확장성을 위한 서비스 복제.** 프로세서 예상 개수에 만족하는 경우 가용성 및 확장성을 위한 서비스 품질 요구 사항을 고려하여 설계를 수정합니다. 가용성 및 페일오버 고려 사항을 처리할 로드 균형 조정 솔루션을 고려합니다.
분석 과정 동안 설계 결정에 대한 균형 조정을 고려합니다. 예를 들면 가용성과 확장성 전략이 시스템의 서비스 가용성(유지 보수)에 어떤 영향을 미치는지 기타 전략 비용은 무엇인지 등을 고려합니다.
- **병목 현상 식별.** 분석을 계속하면서 배포 설계를 검사하여 데이터 전송이 요구 사항에 미치지 못하도록 만드는 모든 병목 현상을 식별하고 조정합니다.
- **자원 최적화.** 자원 관리 측면에서 배포 설계를 검토하고 요구 사항을 충족함과 동시에 비용을 최소화할 수 있는 선택 사항을 고려합니다.
- **위기 관리.** 초기 계획에서 예견할 수 없었던 이벤트 또는 상황에 대해 수정하면서 설계에 대한 비즈니스와 기술적 분석을 다시 고려합니다.

프로세서 요구 사항 예상

이 절에서는 배포 설계의 서비스를 지원하는데 필요한 CPU 프로세서와 해당 메모리 수를 예상하기 위한 과정을 설명합니다. 이 절은 예제 통신 배포 시나리오에 대한 예상 과정의 간략한 설명을 포함합니다.

CPU 컴퓨터 성능의 예상은 반복 프로세스로 다음과 같은 사항을 고려합니다.

- 논리적 구성 요소 및 해당 인터페이스(논리적 구조의 구성 요소에서 지정한 바와 같음)

- 식별된 사용 사례에 대한 사용 분석
- 서비스 품질 요구 사항
- Java Enterprise System 배포 설계에 대한 과거 경험
- 다양한 배포 시나리오 유형을 설계하고 구현한 경험이 있는 Sun 전문 서비스의 자문

예상 과정은 다음 단계를 포함합니다. 이 단계의 순서는 중요하지 않으며 단지 최종 결과에 영향을 미치는 요소들을 고려하는 방법을 제공할 뿐입니다.

1. 시스템에 대한 사용자의 시작점으로 식별된 구성 요소를 위한 기본 CPU 예상 개수를 결정합니다.

설계 결정의 하나는 CPU를 완전히 또는 부분적으로 로드할 것인지 하는 문제입니다. 완전히 로드된 CPU는 시스템 용량을 최대화합니다. 용량을 증가하려면 유지 보수 비용과 CPU를 추가함으로써 발생할 수 있는 중단 시간을 감수해야 합니다. 일부 경우에는 증가하는 성능 요구 사항을 충족하기 위해 시스템을 추가하도록 선택할 수 있습니다.

부분적으로 로드된 CPU는 유지 보수 비용을 즉시 감수하지 않고도 초과 성능 요구 사항을 처리할 여지를 줄 수 있습니다. 그러나 사용되지 않는 시스템을 위한 선불 비용이 추가됩니다.
2. 구성 요소 간의 상호 작용을 고려하여 CPU 예상 개수를 조정해야 합니다.

논리적 구조의 구성 요소 간 상호 작용을 연구하여 종속 구성 요소로 인한 추가 로드를 결정하십시오.
3. 특정 사용 사례에 대한 사용 분석을 연구하여 시스템의 최고 부하를 판별하고 최고 부하를 처리할 구성 요소를 조정합니다.

대부분의 로드를 필요로 하는 가장 과중한 사용 사례로부터 시작하여 각 사용 사례를 계속하며 예상되는 모든 사용 시나리오를 고려했는지 확인합니다.
4. 보안, 가용성, 확장성 요구 사항을 반영하여 CPU 예상 개수를 조정합니다.

이 예상 과정은 필요한 실제 처리 능력을 판별하기 위한 출발점을 제공합니다. 일반적으로 이 예상치를 기반으로 프로토타입 배포를 만든 다음 예상되는 사용 사례에 대해 정밀한 테스트를 수행합니다. 반복 테스트를 한 후에만 배포 설계를 위한 실제 처리 요구 사항을 판별할 수 있습니다.

프로세서 요구 사항 예

이 절에서는 예제 배포에서 요구되는 처리 능력을 예상하기 위한 한 방법을 보여줍니다. 배포 예는 59 페이지 “신원 기반 통신 예” 절에서 설명한 대로 약 1,000명에서 5,000명까지의 직원을 가진 중소 기업을 위한 신원 기반 통신 솔루션에 대한 논리적 구조를 기반으로 합니다.

이 예에서 사용한 CPU 및 메모리 수치는 설명을 위한 임의적인 예상치입니다. 이 수치는 이론적인 예가 기반으로 하는 임의적인 데이터를 기반으로 합니다. 다양한 요소에 대한 철저한 분석은 프로세서 요구 사항 예상에 필수적입니다. 이 분석은 다음과 같은 정보를 포함하지만 여기에 제한되지는 않습니다.

- 철저한 비즈니스 분석을 기반으로 하는 자세한 사용 사례 및 사용 분석

- 비즈니스 요구 사항 분석으로 판별된 서비스 품질 요구 사항
- 프로세싱 및 네트워킹 하드웨어의 특정 비용 및 사양
- 유사한 배포를 구현한 과거 경험



주의 - 이 예에서 나타난 정보는 시스템을 설계할 때 사용할 수 있는 과정을 보여 주는 것 이외에 특정 구현 정보를 나타내지는 않습니다.

사용자 시작점에 대한 기본 CPU 예상치 결정

먼저 사용자 시작점인 각 구성 요소에서 예상되는 로드를 처리하는 데 필요한 CPU 개수를 예상합니다. 다음 수치는 앞서 59 페이지 “신원 기반 통신 예”에서 설명한 신원 기반 통신 시나리오에 대한 논리적 구조를 보여 줍니다.

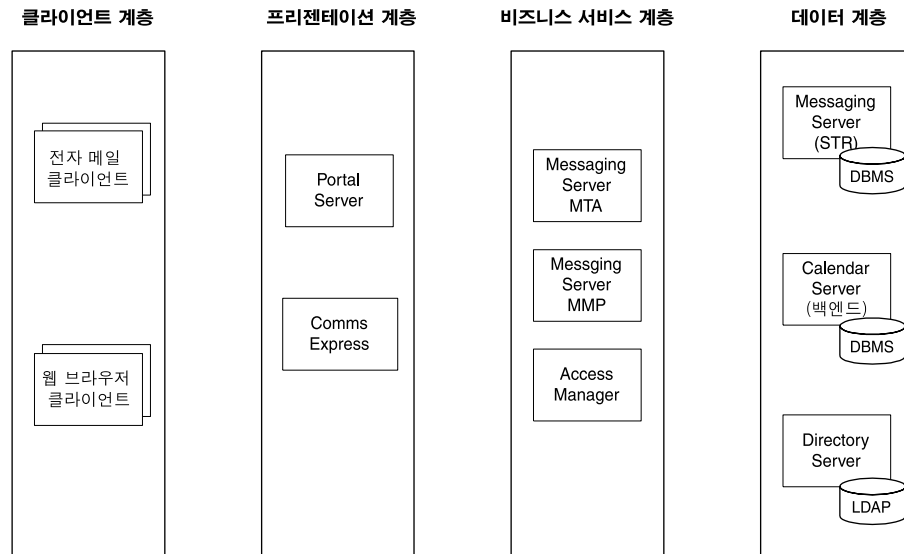


그림 5-1 신원 기반 통신 시나리오의 논리적 구조

다음 표는 배포의 최종 사용자와 직접 상호 작용하는 논리적 구조의 프리젠테이션 계층의 구성 요소를 나열합니다. 표에는 기술적 요구 사항 분석, 사용 사례, 특정 사용 분석 및 해당 배포 유형에 대한 과거 경험에 기초한 기본 CPU 예상 개수가 포함되어 있습니다.

표 5-1 사용자 시작점 액세스를 포함하는 구성 요소에 대한 CPU 예상 개수

구성 요소	CPU 수	설명
Portal Server	4	사용자 시작점인 구성 요소
Communications Express	2	Portal Server 메시징 및 달력 채널로 데이터 경로를 지정합니다.

서비스 종속성에 대한 CPU 예상 개수 포함

사용자 시작점을 제공하는 구성 요소는 다른 Java Enterprise System 구성 요소에서 지원해야 합니다. 성능 요구 사항을 계속 지정하려면 다른 구성 요소에서 필요로 하는 지원을 고려하여 성능 예상치를 포함합니다. 구성 요소 간 상호 작용 유형은 54 페이지 “논리적 구조 예” 절에서 논리적 구조의 예로 설명한 바와 같이 논리적 구조를 설계할 때 자세히 나타내야 합니다.

표 5-2 지원 구성 요소에 대한 CPU 예상 개수

구성 요소	CPU	설명
Messaging Server MTA(인바운드)	1	Communications Express에서 들어오는 메일 메시지 및 전자 메일 클라이언트의 경로를 지정합니다.
Messaging Server MTA(아웃바운드)	1	수신자에게 보내는 메일 메시지 경로를 지정합니다.
Messaging Server MMP	1	전자 메일 클라이언트에 대한 Messaging Server 메시지 저장소에 액세스합니다.
Messaging Server STR(메시지 저장소)	1	전자 메일 메시지를 검색하고 저장합니다.
Access Manager	2	권한 부여 및 인증 서비스를 제공합니다.
Calendar Server(백엔드)	2	Communications Express, Calendar Server 프런트엔드에 대한 달력 데이터를 검색하고 저장합니다.
Directory Server	2	LDAP 디렉토리 서비스를 제공합니다.
Web Server	0	Portal Server 및 Access Manager에 대해 웹 컨테이너 지원을 제공합니다. (추가 CPU 사이클이 필요 없습니다.)

최고 부하 사용에 대한 사용 사례 연구

사용 사례와 사용 분석으로 돌아가서 최고 부하 사용 영역을 식별하고 CPU 예상 개수를 조정합니다.

예를 들어 다음과 같이 최고 부하 조건을 식별하는 예를 가정합니다.

- 사용자가 동시에 로그인한 경우 발생하는 초기 경사로
- 지정된 시간 프레임 동안의 전자 메일 교환

최고 부하 사용을 고려하려면 이러한 서비스를 제공하는 구성 요소를 조정해야 합니다. 다음 표는 이러한 최고 부하 사용을 고려할 수 있는 조정에 대해 설명합니다.

표 5-3 최고 부하를 위해 CPU 예상 개수 조정

구성 요소	CPU(조정됨)	설명
Messaging Server MTAinbound	2	받는 전자 메일의 최고 부하를 위해 CPU 1개 추가
Messaging Server MTAoutbound	2	보내는 전자 메일의 최고 부하를 위해 CPU 1개 추가
Messaging ServerMMP	2	추가 로드를 위해 CPU 1개 추가
Messaging Server STR(메시지 저장소)	2	추가 로드를 위해 CPU 1개 추가
Directory Server	3	추가 LDAP 조회를 위해 CPU 1개 추가

기타 로드 조건에 대한 예상치 수정

다음과 같이 로드에는 영향을 줄 수 있는 기타 서비스 품질 요구 사항을 고려하여 CPU를 계속 예상합니다.

- **보안.**기술적 요구 사항 단계에서부터 데이터의 보안 전송이 로드 요구 사항에 어떻게 영향을 미칠 수 있는지 판별하고 그에 따라 예상치를 수정합니다. 다음 73 페이지 “보안 트랜잭션을 위한 프로세서 요구 사항 예상” 절에 조정을 위한 과정이 설명되어 있습니다.
- **서비스 복제.**가용성, 로드 균형 조정 및 확장성 고려 사항에 대한 서비스 복제를 고려하여 CPU 예상 개수를 조정합니다. 다음 76 페이지 “가용성 전략 결정” 절에 가용성 솔루션을 위한 크기 조정이 설명되어 있습니다. 83 페이지 “확장성에 대한 전략 결정” 절에는 디렉토리 서비스에 사용 가능한 액세스에 관련된 솔루션이 설명되어 있습니다.
- **잠재 용량 및 확장성.**배포의 예기치 못한 큰 로드를 위한 잠재 용량을 허용할 수 있도록 CPU 예상 개수를 수정합니다. 예상되는 확장 시점 및 시간에 따라 예상되는 로드 증가를 살펴보고 예상되는 시점에 도달하도록 시스템을 수평으로 또는 수직으로 확장합니다.

CPU 예상 개수 업데이트

일반적으로 CPU 개수를 짝수로 반올림합니다. 짝수로 반올림하면 두 개의 물리적 서버를 균등하게 나눌 수 있으며 잠재 용량을 위해 적은 요소를 추가할 수 있습니다. 그러나 서비스 복제에 필요한 특정 요구에 따라 반올림해야 합니다.

일반적으로 각 CPU 당 2GB의 메모리가 필요합니다. 필요한 실제 메모리는 특정 사용에 따라 다르며 테스트 중에 결정할 수 있습니다.

다음 표는 신원 기반 통신 예에 대한 최종 예상치를 나열합니다. 이 예상치에는 보안 및 가용성을 위해 추가할 수 있는 기타 추가 컴퓨팅 성능은 포함되지 않습니다. 보안 및 가용성을 위한 전체는 다음 절에서 추가될 것입니다.

표 5-4 구성 요소 지원을 위한 CPU 예상 개수 조정

구성 요소	CPU	메모리
Portal Server	4	8GB
Communications Express	2	4GB
Messaging Server(MTA, 인바운드)	2	4GB
Messaging Server(MTA, 아웃바운드)	2	4GB
Messaging Server(MMP)	2	4GB
Messaging Server(메시지 저장소)	2	4GB
Access Manager	2	4GB
Calendar Server	2	4GB
Directory Server	4	8GB(3CPU/6GB 메모리에서 반올림됨)
Web Server	0	0

보안 트랜잭션을 위한 프로세서 요구 사항 예상

데이터의 보안 전송에는 SSL(Secure Sockets Layer) 또는 Transport Layer Security와 같은 보안 전송 프로토콜을 통한 트랜잭션 처리가 포함됩니다. 보안 전송을 통해 처리된 트랜잭션은 일반적으로 먼저 보안 세션(핸드셰이크라고 함)을 설정하고, 그 다음 전송된 데이터를 암호화 및 해독하기 위한 추가 컴퓨팅 성능을 필요로 합니다. 사용된 암호화 알고리즘(예:40비트 또는 128비트 암호화 알고리즘)에 따라 추가 컴퓨팅 성능이 상당히 많이 필요할 수 있습니다.

비보안 트랜잭션과 동일한 수준에서 보안 트랜잭션을 수행하려면 추가 컴퓨팅 성능을 마련해야 합니다. 트랜잭션과 이를 처리하는 Sun Java™ Enterprise System 서비스에 따라 보안 트랜잭션에는 비보안 트랜잭션보다 네 배 이상의 컴퓨팅 성능이 필요할 수 있습니다.

보안 트랜잭션을 처리하기 위한 처리 능력을 예상할 때는 먼저 사용 사례를 분석하여 보안 전송이 필요한 트랜잭션의 비율을 결정합니다. 보안 트랜잭션의 성능 요구 사항이 비보안 트랜잭션과 동일한 경우 보안 트랜잭션에 필요한 추가 컴퓨팅 성능을 고려하여 CPU 예상 개수를 수정합니다.

일부 사용 시나리오에서는 보안 전송이 인증에만 필요할 수 있습니다. 사용자가 시스템에 인증되면 데이터 전송을 위한 추가 보안 조치가 필요하지 않습니다. 다른 시나리오의 경우 보안 전송이 모든 트랜잭션에 필요할 수 있습니다.

예를 들면 온라인 전자 상거래의 제품 카탈로그를 찾아볼 때, 고객이 선택을 마치고 “지불”할 준비가 될 때까지 모든 트랜잭션이 비보안 상태일 수 있습니다. 그러나 은행이나 중개업의 배포와 같은 일부 사용 시나리오에서는 모든 또는 대부분의 트랜잭션이 보안되어야 하고 보안 및 비보안 트랜잭션 모두에 동일한 성능 표준이 적용되어야 합니다.

보안 트랜잭션의 CPU 예상 개수

이 절에서는 보안 및 비보안 트랜잭션을 포함하는 이론적 사용 사례를 위한 CPU 요구 사항을 계산하는 방법을 설명하는 배포 예제가 계속됩니다.

보안 트랜잭션을 위한 CPU 요구 사항을 예상하려면 다음과 같이 계산하십시오.

1. 앞 절 69 페이지 “프로세서 요구 사항 예”에서 설명한 대로 CPU 예상 개수에 대한 기본 수치로 시작합니다.
2. 보안 전송이 필요한 트랜잭션의 비율을 계산한 다음 보안 트랜잭션에 대한 CPU 예상 개수를 계산합니다.
3. 비보안 트랜잭션에 대해서는 감소된 CPU 예상 개수를 계산합니다.
4. 총 CPU 예상 개수를 계산하려면 보안 예상치 및 비보안 예상치를 계산합니다.
5. 총 CPU 예상 개수를 짝수로 반올림합니다.

74 페이지 “보안 트랜잭션의 CPU 예상 개수”는 다음을 가정한 상태에서 Portal Server의 사용 사례 및 사용 분석을 기반으로 한 계산 예를 보여줍니다.

- 모든 로그인에는 보안 인증이 필요합니다.
- 모든 로그인이 전체 Portal Server 로드의 10%를 차지합니다.
- 보안 트랜잭션의 성능 요구 사항과 비보안 트랜잭션의 성능 요구 사항이 동일합니다. 보안 트랜잭션을 처리할 추가 컴퓨팅 성능으로 인해 해당 트랜잭션을 처리할 CPU 수가 배수로 증가할 것입니다. 예의 다른 CPU 수치처럼 이 수치는 설명을 위한 임의의 숫자입니다.

표 5-5 보안 트랜잭션에 대한 CPU 예상 개수 수정

단계	설명	계산	결과
1	모든 Portal Server 트랜잭션에 대한 기본 예상으로 시작합니다.	71 페이지 “최고 부하 사용에 대한 사용 사례 연구”의 기본 예상치는 CPU 4개입니다.	-----

표 5-5 보안 트랜잭션에 대한 CPU 예상 개수 수정 (계속)

단계	설명	계산	결과
2	보안 트랜잭션의 추가 CPU 예상 개수를 계산합니다. 보안 트랜잭션이 비보안 트랜잭션보다 4배의 성능을 필요로 한다고 가정합니다.	보안 전송은 기본 예상치의 10배가 필요합니다. $0.10 \times 4 \text{ CPUs} = 0.4 \text{ CPUs}$ 보안 트랜잭션의 CPU 성능을 4배수로 증가합니다. $4 \times 0.4 = 1.6 \text{ CPUs}$	1.6개의 CPU
3	비보안 트랜잭션의 감소된 CPU 예상 개수를 계산합니다.	기본 예상치의 90%는 비보안입니다. $0.9 \times 4 \text{ CPUs} = 3.6 \text{ CPUs}$	3.6개의 CPU
4	보안 및 비보안 트랜잭션의 조정된 총 CPU 예상치를 계산합니다.	보안 예상치 + 비보안 예상치 = 전체. $1.6 \text{ CPUs} + 3.6 \text{ CPUs} = 5.2 \text{ CPUs}$	5.2개의 CPU
5	짝수로 반올림합니다.	$5.2 \text{ CPUs} \implies 6 \text{ CPUs}$	6개의 CPU

이 예의 보안 트랜잭션 계산에서 Portal Server의 다음과 같은 총 개수를 구하려면 추가로 두개의 CPU와 4GB의 메모리를 추가함으로써 74 페이지 “보안 트랜잭션의 CPU 예상 개수”의 총 CPU 예상 개수를 수정할 수 있습니다.

구성 요소	CPU	메모리
Portal Server	6	12GB

SSL 트랜잭션을 처리하는 특수 하드웨어

SSL 가속기 카드 및 기타 장치와 같은 특수한 하드웨어 장치를 사용하여 보안 세션 설정 및 데이터의 암호화 및 해독을 처리할 컴퓨팅 성능을 제공할 수 있습니다. SSL 작업을 위해 특수한 하드웨어를 사용할 경우 계산 성능은 SSL 계산의 일부, 일반적으로 보안 세션을 설정하는 “핸드셰이크” 작업에만 사용됩니다.

이 하드웨어는 최종 배포 구조에 좋은 영향을 줄 수 있습니다. 그러나 하드웨어의 특수성 때문에 먼저 CPU 성능 면에서 보안 트랜잭션 성능 요구 사항을 예상한 다음 특수 하드웨어를 사용하여 추가 로드를 처리할 때의 이점을 고려하는 것이 좋습니다.

특수 하드웨어를 사용할 경우 고려해야 할 요소로는 사용 사례에서 이 하드웨어 사용을 지원 하는지 여부(예를 들어 많은 수의 SSL 핸드셰이크 작업이 필요한 사용 사례)와 이 하드웨어 유형으로 인해 설계에 추가되는 복잡도 계층을 들 수 있습니다. 이러한 복잡도에는 이 장치의 설치 구성 테스트 및 관리가 포함됩니다.

가용성 전략 결정

가용성 요구 사항을 위한 전략을 개발하는 경우 고려할 가용성 솔루션이 어떤 것인지 결정하기 위한 구성 요소 상호 작용 및 사용 분석을 연구합니다. 가용성 및 페일오버 요구 사항을 위한 최적의 솔루션을 판별하여 구성 요소별로 분석합니다.

다음 항목은 가용성 전략을 결정하도록 도와주기 위해 수집한 정보 유형의 예입니다.

- 가용성에 대해 9를 몇 개 지정합니까?
- 페일오버 상황과 관련된 성능 사양은 무엇입니까(예:페일오버 동안 최소한 성능의 50%)?
- 사용 분석이 최고 및 최고가 아닌 사용 시간을 식별합니까?
- 지형적 고려 사항은 무엇입니까?

선택하는 가용성 전략에서는 87 페이지 “최적의 자원 사용을 위한 설계”에 설명되어 있는 서비스 가용성 요구 사항도 고려해야 합니다. 상당한 관리 및 유지 보수가 필요한 복잡한 솔루션은 피합니다.

가용성 전략

Java Enterprise System 배포의 가용성 전략은 다음과 같습니다.

- **로드 균형 조정.**프로세싱 로드를 공유하려면 중복 하드웨어 및 소프트웨어 구성 요소를 사용합니다. 로드 밸런서는 모든 서비스 요청을 서비스의 다중 대칭 인스턴스 중 하나에 전달합니다. 만약 한 인스턴스가 실패하는 경우 보다 과중한 로드로 가정되어 다른 인스턴스가 사용 가능합니다.
- **페일오버.**어떤 구성 요소가 실패하는 경우 서비스의 지속적인 액세스 및 중요한 데이터를 위한 보안을 제공하는 중복 하드웨어 및 소프트웨어 관리가 포함됩니다.
Sun Cluster 소프트웨어는 Messaging Server를 위한 메시지 저장소 및 Calendar Server를 위한 달력 데이터 등 백엔드 구성 요소에 의해 관리되는 주요 데이터를 위한 페일오버 솔루션을 제공합니다.
- **서비스 복제.**서비스 복제는 동일한 데이터에 대한 다중 소스를 제공합니다.
Directory Server는 LDAP 디렉토리 액세스를 위해 여러 개의 복제 및 동기화 전략을 제공합니다.

다음 절에서는 다양한 수준의 로드 균형 조정, 페일오버 및 서비스 복제를 제공하는 가용성 솔루션의 예를 설명합니다.

단일 서버 시스템

서비스를 위한 모든 컴퓨팅 자원을 단일 서버에 배치합니다. 서버에 오류가 발생할 경우 전체 서비스가 실패합니다.

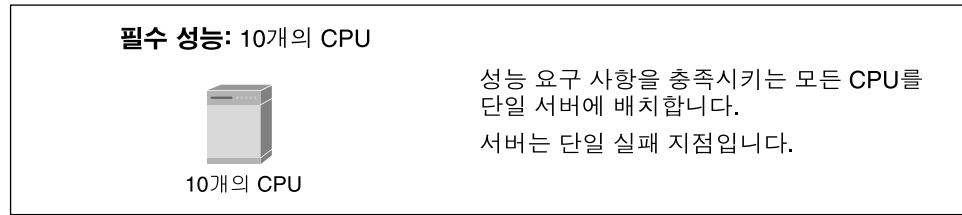


그림 5-2 단일 서버 시스템

Sun에서는 다음 이점을 제공하는 고성능 서버를 제공합니다.

- 시스템 실행 중에 하드웨어 구성 요소의 교체 및 재구성
- 서버의 무결함 도메인에서 다중 응용 프로그램을 실행하는 능력
- 시스템을 다시 부팅하지 않고도 용량, 성능 속도 및 I/O 구성을 업그레이드하는 능력

고성능 서버는 일반적으로 비슷한 다중 서버 시스템보다 비용이 더 많이 듭니다. 그러나 단일 서버인 경우 데이터 센터의 서버에 대한 관리, 모니터링 및 호스팅 비용을 절약할 수 있습니다. 로드 균형 조정, 페일오버 및 단일 실패 지점 제거는 다중 서버 시스템에서 더 유연합니다.

수평으로 중복된 시스템

로드 균형 조정과 페일오버를 제공하는 병렬로 중복된 서버의 가용성을 늘릴 수 있는 몇 가지 방법이 있습니다. 다음 그림에서는 N+1 페일오버 시스템을 제공하는 두 개의 복제 서버를 설명합니다. N+1 시스템에는 한 서버가 실패할 경우 100% 용량을 제공하는 추가 서버가 있습니다.

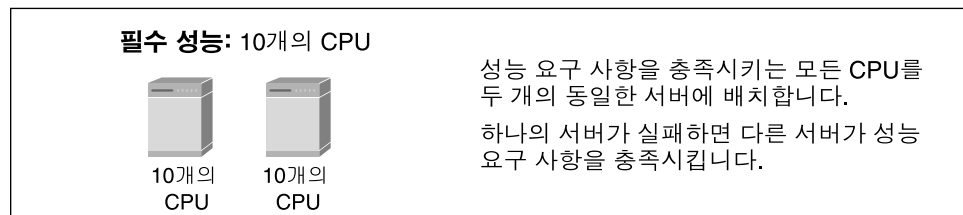


그림 5-3 두 서버를 가진 N+1 페일오버 시스템

위 77 페이지 “수평으로 중복된 시스템”에서 각 서버의 컴퓨팅 성능은 동일합니다. 하나의 서버만 성능 요구 사항을 처리합니다. 다른 서버는 백업으로 서비스에 호출된 경우 100%의 성능을 제공합니다.

N+1 페일오버 설계의 장점은 페일오버 상황에서 100%의 성능을 유지하는 것입니다. 단점은 하드웨어 비용은 증가하면서도 그에 따른 성능을 모두 얻지 못한다는 것입니다(왜냐하면 한 서버는 오직 페일오버 상황에서만 사용하기 위해 대기하고 있기 때문입니다).

다음 그림은 두 서버 간 성능을 분산하도록 로드 균형 조정에 페일오버를 더하여 구현한 시스템을 보여줍니다.

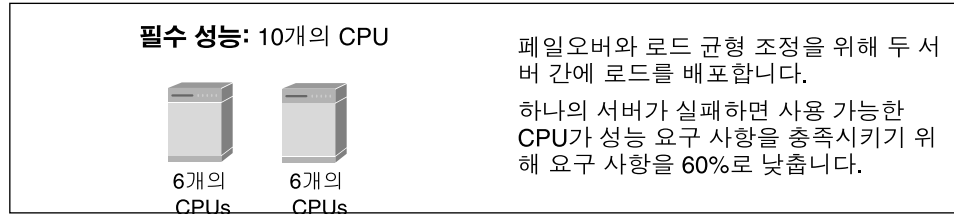


그림 5-4 두 서버 간 로드 균형 조정 및 페일오버

위 77 페이지 “수평으로 중복된 시스템”에서 설명한 시스템에서 한 서버가 실패하더라도 전체 용량의 일부이긴 하지만 모든 서비스가 여전히 사용 가능합니다. 남은 서버에서 10개 CPU 요구 사항의 60%인 6개 CPU의 컴퓨팅 성능을 제공합니다.

이 설계의 장점은 두 서버를 모두 사용할 수 있는 경우 추가로 2개의 CPU 잠재 용량이 있다는 점입니다.

다음 그림에서는 성능 및 로드 균형 조정을 위한 여러 서버 간의 배포를 보여줍니다.

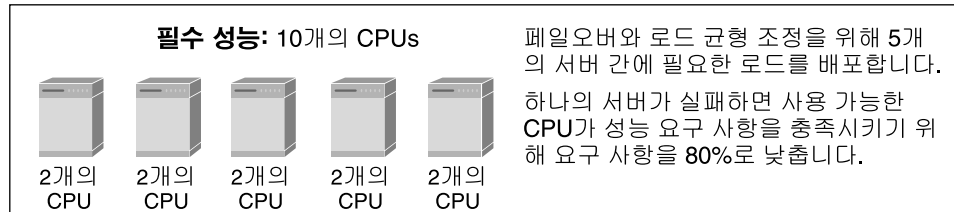


그림 5-5 n개의 서버 간 로드 배포

77 페이지 “수평으로 중복된 시스템”과 같이 이 설계에 다섯 개의 서버가 있기 때문에 한 서버가 실패할 경우 나머지 서버에서 10개 CPU 성능 요구 사항의 80%인 총 8개 CPU의 컴퓨팅 성능을 제공합니다. 설계에 2개 CPU 용량을 가진 서버를 추가할 경우 결과적으로 N+1 설계가 됩니다. 한 서버가 실패하면 나머지 서버에서 성능 요구 사항의 100%를 충족시킵니다.

이 설계에는 다음과 같은 장점이 있습니다.

- 단일 서버가 실패할 경우 성능 추가
- 둘 이상의 서버가 중단될 경우에도 가용성 제공
- 유지 보수 및 업그레이드를 위해 서버의 서비스를 교대할 수 있음
- 복수의 저성능 서버가 일반적으로 하나의 고성능 서버보다 비용이 덜 듭

그러나 서버를 추가할 경우 관리 및 유지 보수 비용이 상당히 늘어날 수 있습니다. 또한 데이터 센터에 서버를 호스팅하는 비용을 고려해야 합니다. 일정 시점이 되면 서버 추가에 따른 수익이 줄어들기 시작합니다.

Sun Cluster 소프트웨어

높은 수준의 가용성이 필요한 경우(예: 4개 또는 5개의 9) Sun Cluster 소프트웨어를 가용성 설계의 일부로 고려할 수 있습니다. 클러스터 시스템은 저장소 및 기타 네트워크 자원이 있는 중복 서버를 연결한 것입니다. 클러스터의 서버들은 서로 계속해서 통신합니다. 한 서버가 오프라인이 될 경우 클러스터의 나머지 장치에서 해당 서버를 격리하고 실패한 노드의 응용 프로그램이나 데이터를 다른 노드로 페일오버합니다. 이 페일오버 프로세스는 시스템 사용자의 서비스를 방해하지 않고 비교적 빠르게 수행됩니다.

Sun Cluster 소프트웨어는 추가 전용 하드웨어와 구성, 관리 및 유지 보수를 위한 특수 기술이 필요합니다.

가용성 설계 예

이 절에서는 59 페이지 “신원 기반 통신 예”에서 이미 설명한 대로 1,000명에서 5,000명의 직원이 있는 중소기업을 위한 신원 기반 통신 솔루션을 기반으로 하는 가용성 전략의 두 가지 예가 설명됩니다. 첫 번째 가용성 전략은 Messaging Server를 위한 로드 균형 조정을 보여줍니다. 두 번째는 Sun Cluster 소프트웨어를 사용하는 페일오버 솔루션을 보여줍니다.

Messaging Server를 위한 로드 균형 조정 예

다음 표에는 논리적 구조에 있는 각 논리적 Messaging Server 구성 요소의 CPU 성능에 대한 예상치가 나열되어 있습니다. 이 표에는 72 페이지 “CPU 예상 개수 업데이트” 절에서 계산한 최종 예상치가 반복되어 있습니다.

표 5-6 구성 요소 지원을 위한 CPU 예상 개수조정

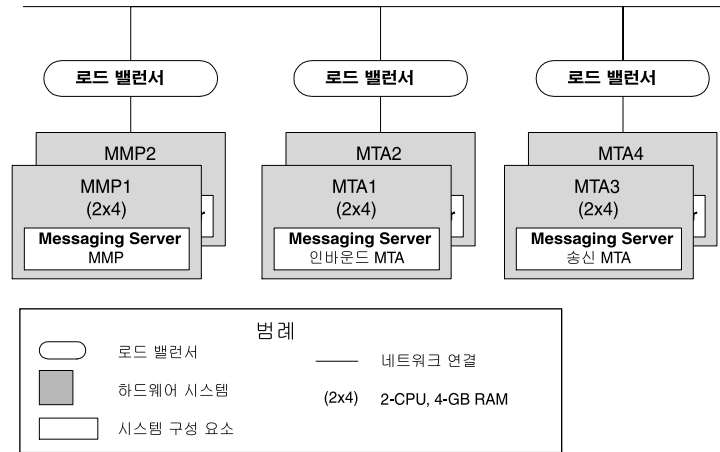
구성 요소	CPU	메모리
Messaging Server(MTA, 인바운드)	2	4GB
Messaging Server(MTA, 아웃바운드)	2	4GB
Messaging Server(MMP)	2	4GB
Messaging Server(메시지 저장소)	2	4GB

예를 들면 기술적 요구 사항 단계 중에 서비스 품질 요구 사항이 다음과 같이 지정되었다고 가정합니다.

- **가용성.** 전반적인 시스템 가용성이 99.99%(예정된 중단 시간을 포함하지 않음)가 되어야 합니다. 개별 컴퓨터 시스템 실패가 서비스 실패를 일으키지 않아야 합니다.

- **확장성.** 어떤 서버도 일별 최고 로드 하에서 80% 이상 사용될 수 없으며 시스템은 연간 10%의 장기적인 증가를 수용할 수 있어야 합니다.

가용성 요구 사항을 성취하기 위해 각 Messaging Server의 구성 요소가 각각 별개 하드웨어 서버에 있는 두 개의 인스턴스를 제공합니다. 한 구성 요소를 가진 서버가 실패하는 경우 다른 서버가 서비스를 제공합니다. 다음 그림은 이러한 가용성 전략을 위한 네트워크 다이어그램을 보여줍니다.



이전 그림에서 CPU 수가 원래 예상치의 두 배가 됩니다. CPU는 다음 이유로 두 배가 됩니다.

- 한 서버가 실패한 경우 남은 서버가 로드를 처리하기 위해 CPU 성능을 제공합니다.
- 최고 로드 하에서 어떤 단일 서버도 80% 이상 사용하지 않는 확장성 요구 사항의 경우 추가된 CPU 성능이 이러한 안전 여유분을 제공합니다.
- 연간 로드 10% 증가를 수용하기 위한 확장성 요구 사항의 경우 추가된 CPU 성능이 추가 확장이 필요할 때까지 증가하는 로드를 처리할 수 있는 잠재 용량을 가져옵니다.

Sun Cluster 소프트웨어를 사용하는 페일오버 예

다음 그림은 Calendar Server 백엔드 및 Messaging Server 메시징 저장소에 대한 페일오버 전략 예를 나타냅니다. Calendar Server 백엔드 및 메시징 저장소는 별개 하드웨어 서버에 복제되며 Sun Cluster 소프트웨어와 함께 페일오버를 위해 구성됩니다. CPU 수와 해당 메모리는 Sun Cluster의 각 서버에 복제됩니다.

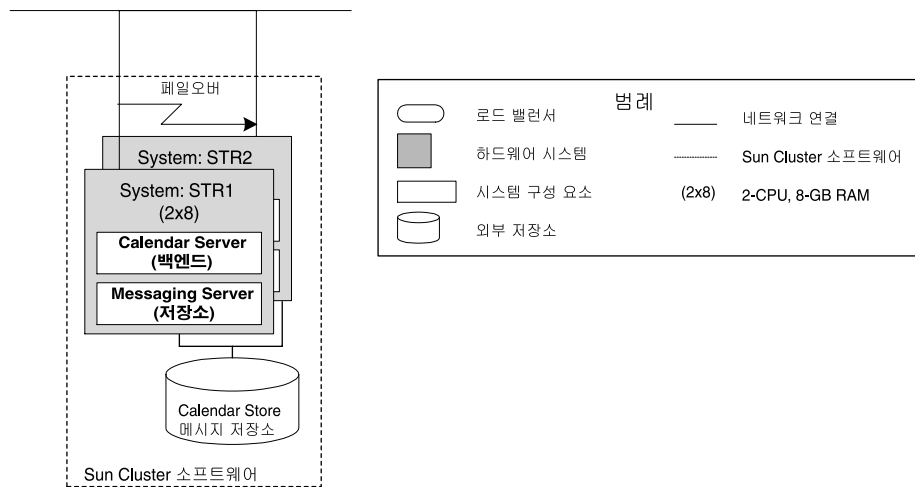


그림 5-6 Sun Cluster 소프트웨어를 사용하는 페일오버 예

디렉토리 서비스 복제 예

디렉토리 서비스를 고가용성을 제공하면서 다른 서버 간 트랜잭션을 분산하기 위해 복제할 수 있습니다. Directory Server는 다음을 포함하여 서비스 복제의 다양한 전략을 제공합니다.

- **다중 데이터베이스.** 별개의 데이터베이스에 디렉토리 트리의 각각 다른 부분을 저장합니다.
- **연쇄화 및 참조.** 분산 데이터를 단일 디렉토리 트리에 연결합니다.
- **단일 마스터 복제.** 마스터 데이터베이스에 대한 중앙 소스를 제공하며 이 중앙 소스는 사용자 복제본에 분산됩니다.
- **다중 마스터 복제.** 여러 서버 내에 마스터 데이터베이스를 분산합니다. 그 다음에 각각의 마스터는 사용자 복제본으로 해당 데이터베이스를 분산합니다.

Directory Server를 위한 가용성 전략은 복잡한 내용으로서 이 설명서 범위를 벗어납니다. 다음 절인 81 페이지 “단일 마스터 복제” 및 82 페이지 “다중 마스터 복제”에서는 기본 복제 전략에 대한 수준 높은 설명을 제공합니다. 자세한 내용은 **Sun Java System Directory Server 5 2005Q1 Deployment Planning Guide**를 참조하십시오.

단일 마스터 복제

다음 그림은 기본 복제 개념을 보여 주는 단일 마스터 복제 전략을 나타냅니다.

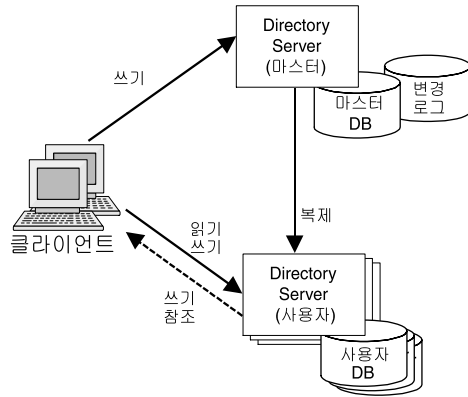


그림 5-7 단일 마스터 복제 예

단일 마스터 복제에서는 Directory Server의 한 인스턴스가 모든 변경 사항을 기록하면서 마스터 디렉토리 데이터베이스를 관리합니다. 마스터 데이터베이스는 모든 수의 사용자 데이터베이스에 복제됩니다. Directory Server의 사용자 인스턴스는 읽기 및 검색 작업을 위해 최적화되어 있습니다. 사용자가 수신한 모든 읽기 작업은 마스터에게 돌아가 참조됩니다. 마스터는 주기적으로 사용자 데이터베이스를 업데이트합니다.

단일 마스터 복제의 장점은 다음과 같습니다.

- 데이터베이스 읽기 및 쓰기 작업을 위해 최적화된 Directory Server의 단일 인스턴스
- 읽기 및 검색 작업을 위해 최적화된 Directory Server의 모든 사용자 인스턴스
- Directory Server의 사용자 인스턴스에 대한 수평 확장성

다중 마스터 복제

다음 그림은 디렉토리 액세스를 전세계적으로 분산하는데 사용할 수 있는 다중 마스터 복제 전략을 나타냅니다.

다중 마스터 복제에서는 Directory Server의 한 개 이상의 인스턴스가 마스터 디렉토리 데이터베이스를 관리합니다. 각 마스터에는 마스터 데이터베이스를 동기화하기 위한 절차를 지정하는 복제 계약이 있습니다. 각 마스터는 모든 사용자 데이터베이스에 복제합니다. 단일 마스터 복제와 같이 Directory Server의 사용자 인스턴스는 읽기 및 검색 액세스를 위해 최적화되어 있습니다. 사용자가 수신한 모든 읽기 작업은 마스터에게 돌아가 참조됩니다. 마스터는 주기적으로 사용자 데이터베이스를 업데이트합니다.

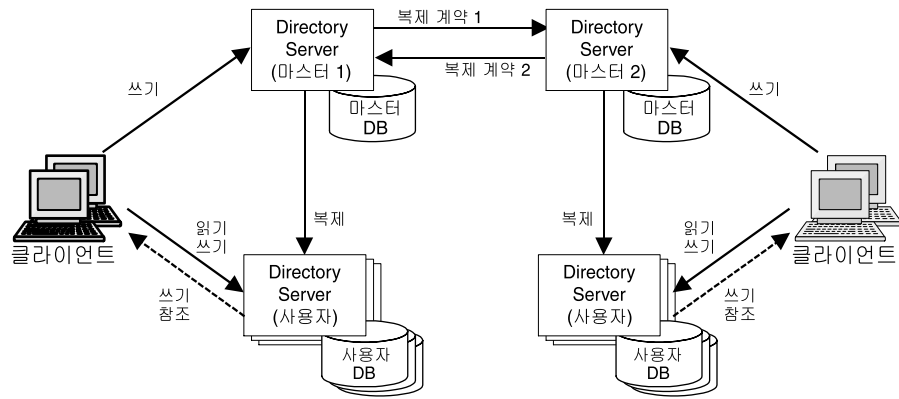


그림 5-8 다중 마스터 복제 예

다중 마스터 복제 전략은 단일 마스터 복제의 모든 장점에 더하여 마스터에 업데이트할 경우 로드 균형을 제어할 수 있는 가용성 전략을 제공합니다. 또한 전세계적으로 분산된 데이터 센터가 있는 기업에 있어서 중요한 고려 사항인 디렉토리 작업의 로컬 제어를 제공하는 가용성 전략도 구현할 수 있습니다.

확장성에 대한 전략 결정

확장성은 대개 시스템 자원을 추가하지만 배포 구조는 변경하지 않고 시스템에 용량을 추가할 수 있는 기능을 말합니다. 요구 사항 분석 시 일반적으로 비즈니스 요구 사항과 그 후의 사용 분석을 기준으로 예상되는 시스템 증가를 예상합니다. 시스템의 사용자 수에 대한 이 예측과 이들 사용자의 요구 사항을 충족시키기 위한 시스템의 용량은 배포된 시스템의 실제 수치와는 크게 다른 경우가 많습니다. 예측의 차이를 수용할 수 있도록 설계에 융통성이 있어야 합니다.

확장 가능한 설계는 시스템이 추가 자원과 함께 업그레이드할 수 있을 때까지 증가된 로드를 처리할 수 있는 충분한 잠재 용량을 포함합니다. 확장 가능 설계는 시스템을 다시 설계하지 않고도 증가하는 로드를 처리하도록 즉시 확장될 수 있습니다.

잠재 용량

잠재 용량은 추가 성능과 가용성 자원을 시스템에 포함시켜서 비정상적인 최고 로드를 쉽게 처리할 수 있도록 하는 확장성의 한 요소입니다. 또한 잠재 용량이 배포 시스템에서 사용되는 방법을 모니터링하여 언제 자원을 추가하여 시스템을 확장해야 할 지를 결정할 수 있습니다. 잠재 용량은 설계에 안전을 확립하는 한 가지 방법입니다.

사용 사례 분석으로 비정상적인 최고 로드를 만들 수 있는 시나리오를 식별할 수 있습니다. 예상치 않은 증가를 처리하고 시스템에 안전을 확립하는 잠재 용량을 설계하는 요소와 더불어 이와 같은 비정상적인 최고 로드를 분석합니다.

시스템 설계는 일반적으로 운영의 처음 6-12 개월 동안 합리적인 시간에 예상된 용량을 처리할 수 있어야 합니다. 유지 보수 주기를 사용하여 자원을 추가하거나 필요한 용량을 늘릴 수 있습니다. 시스템 업그레이드를 정기적으로 예약할 수 있는 것이 가장 좋지만 필요한 용량 증가를 예측하는 것은 종종 어렵습니다. 시스템 업그레이드 시기를 결정하기 위해서는 비즈니스 예측뿐만 아니라 자원에 대한 신중한 모니터링에 의존합니다.

솔루션을 증분 단계에 구현하려고 하는 경우 각 증분 단계에 대해 예약된 다른 향상과 일치하도록 증가하는 시스템 용량을 예약해야 합니다.

확장성 예

이 절의 예는 Messaging Server를 구현하는 솔루션을 위한 수평 및 수직 확장을 설명합니다. 수직 확장의 경우 서버에 CPU를 추가하여 증가하는 로드를 처리할 수 있게 합니다. 수평 확장의 경우 로드 분산을 위한 서버를 추가하여 증가하는 로드를 처리합니다.

이 예는 기본적으로 로드 균형 조정을 위해 분산된 메시지 저장소 인스턴스 2개가 50,000 사용자를 지원한다고 가정합니다. 각 서버가 CPU를 2개씩 가지고 있어 총 CPU는 4개가 됩니다. 다음 그림은 250,000 사용자 및 2,000,000 사용자에게 대해 증가하는 로드를 처리할 수 있도록 이 시스템을 확장할 수 있는 방법을 나타냅니다.

주 - 84 페이지 "확장성 예"는 수평 확장 및 수직 확장 간의 차이를 나타냅니다. 이 그림에서는 로드 균형 조정, 페일오버 및 사용 패턴 변경과 같이 확장 시 고려해야 할 기타 요소는 보여주지 않습니다.

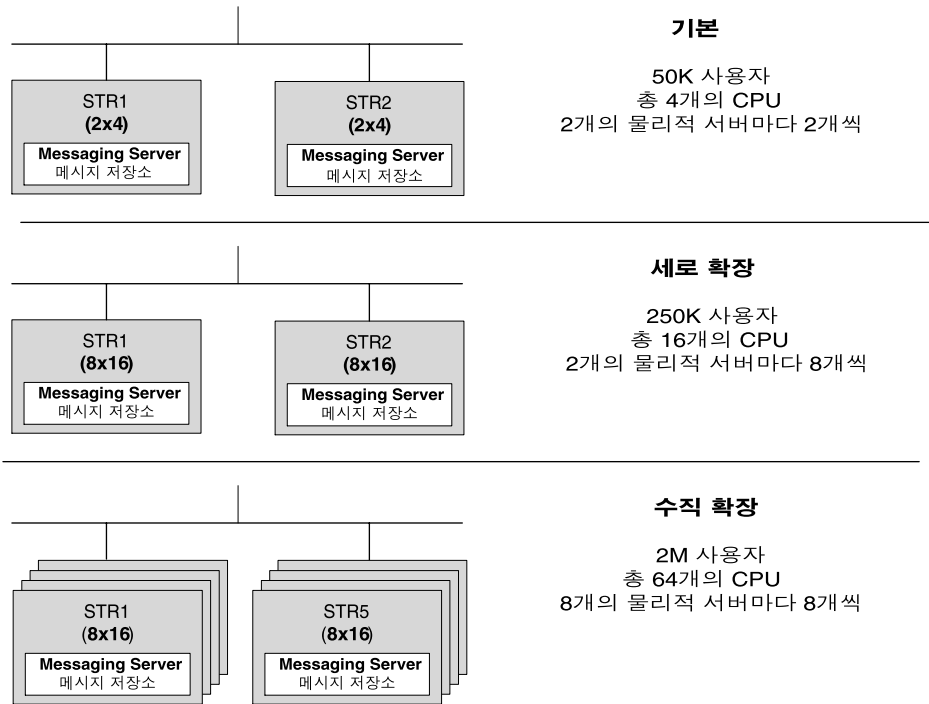


그림 5-9 수평 및 수직 확장 예

성능 병목 현상 식별

배포 설계를 성공으로 이끄는 열쇠 중 하나는 잠재적인 성능 병목 현상을 식별하여 그것을 피하는 전략을 개발하는 것입니다. 성능 병목 현상은 어떤 데이터를 액세스하는 비율이 지정된 시스템 요구 사항에 미치지 못할 때 발생합니다.

병목 현상은 다음 표의 시스템 간 데이터 액세스 지점에서 나열한 대로 다양한 하드웨어 클래스에 따라 범주화할 수 있습니다. 또한 이 표는 각 하드웨어 클래스의 병목 현상에 대한 가능한 해결 방법을 제시합니다.

표 5-7 데이터 액세스 지점

하드웨어 클래스	상대 액세스 속도	성능 향상을 위한 해결 방법
프로세서	10억분의 1초	수직 확장: 처리 능력 추가 및 프로세서 캐시 향상 수평 확장: 로드 균형 조정을 위한 병렬 처리 능력 추가
시스템 메모리(RAM)	100만분의 1초	시스템 메모리를 특정 작업 전용으로 사용 수직 확장: 메모리 추가 수평 확장: 병렬 처리 및 로드 균형 조정을 위한 추가 인스턴스 생성
디스크 읽기 및 쓰기	밀리초	디스크 배열(RAID)로 디스크 액세스 최적화 디스크 액세스를 읽기 및 쓰기 전용과 같이 특정 기능 전용으로 사용 자주 액세스하는 데이터를 시스템 메모리에 캐시
네트워크 인터페이스	네트워크 노드의 대역폭 및 액세스 속도에 따라 다양화	대역폭 증가 보안 데이터 전송 시 가속기 하드웨어 추가 데이터를 즉시 사용할 수 있도록 네트워크 간 노드의 성능 향상

주 - 85 페이지 “성능 병목 현상 식별”에서는 디스크와 같은 느린 액세스 지점이 병목 현상의 원인이 될 수 있음을 나타내면서 상대적인 액세스 속도에 따라 하드웨어 클래스를 나열합니다. 그러나 대형 로드를 처리하기에 성능이 떨어지는 프로세서 또한 병목 현상의 원인이 될 수 있습니다.

일반적으로 배포 및 해당 종속성의 각 구성 요소에 대한 기본 처리 능력 예상치로 배포 설계를 시작합니다. 그 다음에 시스템 메모리 및 디스크 액세스와 관련된 병목 현상을 피하는 방법을 결정합니다. 마지막으로 잠재적인 병목 현상을 결정할 수 있도록 네트워크 인터페이스를 검사하고 극복할 전략에 집중합니다.

디스크 액세스 최적화

배포 설계의 중요한 구성 요소는 LDAP 디렉토리와 같이 자주 액세스하는 데이터 집합에 대한 디스크 액세스 속도입니다. 디스크 액세스는 데이터로 가장 느리게 액세스하여 성능 병목 현상의 원인이 될 수 있습니다.

디스크 액세스를 최적화하는 한 방법은 읽기 작업과 쓰기 작업을 분리하는 것입니다. 쓰기 작업은 읽기 작업보다 비용이 많이 들 뿐 아니라 읽기 작업(LDAP 디렉토리 조회 작업)은 일반적으로 쓰기 작업(LDAP 디렉토리의 데이터 업데이트)보다 상당히 자주 발생합니다.

디스크 액세스를 최적화하는 또 다른 방법은 다른 유형의 I/O 작업에 대해 전용 디스크를 두는 것입니다. 예를 들면 트랜잭션 로그와 이벤트 로그, LDAP 읽기 및 쓰기 작업과 같은 Directory Server 로깅 작업을 위해 별개 디스크를 제공합니다.

또한 읽기 및 쓰기 작업 전용인 Directory Server에 한 개 이상의 인스턴스를 구현하고 로컬 서버로 분산된 복제 인스턴스를 읽기 및 검색 액세스를 위해 사용할 것을 고려합니다. 연쇄 및 연결 선택 사항 또한 디렉토리 서비스 액세스를 최적화하는데 사용할 수 있습니다.

Sun Java System Directory Server 5 2005Q1 Deployment Planning Guide에서는 디스크 액세스 계획의 다양한 요소를 설명합니다. 이 장의 내용은 다음과 같습니다.

- **최소 메모리 및 디스크 공간 요구 사항.** 다양한 크기의 디렉토리에 필요한 디스크 및 메모리 예상치를 제공합니다.
- **캐시 액세스를 위한 물리적 메모리 크기 조정.** Directory Server의 계획된 사용에 따른 캐시 크기 예상 및 총 메모리 사용 계획에 대한 지침을 제공합니다.
- **디스크 부속 시스템 크기 조정.** 디렉토리 접미어 및 디스크 사용에 영향을 미치는 Directory Server 요소에 따른 디스크 공간 요구 사항 계획 및 다양한 디스크 배열 대안을 포함하는 디스크 전반에 걸친 파일 분산에 대한 정보를 제공합니다.

최적의 자원 사용을 위한 설계

배포 설계가 단지 서비스 품질 요구 사항을 충족하는데 필요한 자원을 예상하는 것만은 아닙니다. 배포 설계 중에 모든 사용 가능한 선택 사항을 분석하고 비용을 최소화하면서도 서비스 품질 요구 사항을 계속 충족하는 최상의 솔루션을 선택합니다. 한 영역의 이점이 다른 영역의 비용으로 상쇄되지 않도록 각 설계 결정 간의 균형을 조절하기 위한 분석을 해야 합니다.

예를 들어 가용성을 가로로 확장할 경우 전체적인 가용성이 증가할 수 있지만 유지 보수 및 서비스 비용이 증가됩니다. 성능을 세로로 확장하면 컴퓨팅 성능을 저렴한 비용으로 확장할 수 있지만 일부 서비스에서 추가 성능을 비효율적으로 사용할 수 있습니다.

설계 전략을 완료하기 전에 결정을 검토하여 제안된 솔루션에 대해 전체적으로 이익이 되도록 자원 사용의 균형을 유지하는지 확인합니다. 이 분석은 일반적으로 한 영역의 시스템 품질이 다른 시스템 품질에 어떤 영향을 미치는지도 검토합니다. 다음 표는 자원 관리를 위한 시스템 품질 및 해당 고려 사항을 나열합니다.

표 5-8 자원 관리 고려 사항

시스템 품질	설명
성능	개별 서버의 CPU에 집중하는 성능 솔루션의 경우 서비스가 컴퓨팅 성능을 효율적으로 이용할 수 있습니까? (예를 들어 일부 서비스의 경우 효율적으로 사용할 수 있는 CPU 개수에 한계가 있습니다.)
잠재 용량	전략이 성능 예상치를 초과하는 로드를 처리합니까? 서버의 세로 확장, 다른 서버에 대한 로드 균형 조정 또는 둘 다를 사용하여 과도한 로드를 처리합니까? 배포 확장을 위한 다음 시점까지 비정상적인 최고 로드를 처리할 만큼 잠재 용량이 충분합니까?
보안	보안 트랜잭션을 처리하는 데 필요한 성능 오버헤드를 충분히 고려했습니까?
가용성	수평으로 중복된 솔루션의 경우 장기 유지 보수 비용을 충분히 예상했습니까? 시스템을 유지 보수하는 데 필요한 예약된 중단 시간을 고려했습니까? 고성능 서버와 저성능 서버 간의 비용을 공정하게 고려했습니다.
확장성	배포를 확장하기 위한 중요 시점을 예상했습니까? 배포 확장의 중요 시점까지 예상되는 로드 증가를 처리할 충분한 잠재적 용량을 제공하는 전략이 있습니까?
실용성	가용성 설계에서 관리, 모니터링 및 유지 보수 비용을 고려했습니까? 관리 비용을 줄이기 위해 사용자가 직접 일부 관리 작업을 수행할 수 있도록 하는 위임 관리 솔루션을 고려해 보았습니까?

위기 관리

서비스 품질 요구 사항 및 사용 분석 같은 배포 설계의 기반이 되는 정보 대부분은 경험적인 데이터가 아니라 궁극적으로 비즈니스 분석으로부터 이끌어낸 예측과 예상 기반인 데이터입니다. 이러한 예상은 비즈니스 운영 중 예기치 못한 상황, 데이터 수집의 잘못된 방법 또는 단순하게 사람의 실수 등의 여러 이유로 인해 정확하지 않을 수 있습니다. 배포 설계를 완료하기 전에 설계의 기반이 되는 분석을 재검토하고 예측 또는 예상으로부터 모든 합리적인 이탈을 고려하여 설계했는지 확인합니다.

예를 들면 사용 분석이 시스템의 실제 사용량에 못 미칠 경우 발생하는 트래픽 양을 처리할 수 없는 시스템을 구성할 위험이 있습니다. 제 성능을 내지 못하는 설계는 결단코 실패로 간주해야 합니다.

한편 필요한 것보다 몇 곱절 더 강력한 시스템을 구성하는 경우 다른 곳에서 사용할 수 있는 자원을 유용하는 것입니다. 요구 사항을 초과하는 여분의 안전을 포함시키면서도 자원을 지나치게 사용하는 것은 피하는 것이 열쇠입니다.

지나친 자원 사용은 설계를 실패하게 합니다. 왜냐하면 사용되지 않은 자원을 다른 영역에 적용할 수도 있기 때문입니다. 또한 지나친 솔루션은 이해 관계자에게 계약을 성실하게 이행하지 못한 것으로 간주될 수 있습니다.

예제 배포 구조

다음 그림에서는 백서 앞 부분에서 소개한 예제 배포의 전체 배포 구조를 보여줍니다. 이 그림은 배포 구조를 제공하는 방법에 대한 개념을 설명합니다.



주의 - 다음 그림의 배포 구조는 설명을 위한 것입니다. 실제로 설계, 구축 또는 테스트한 배포를 나타내지 않으며 배포 계획에 대한 유용한 정보로 간주해서는 안 됩니다.

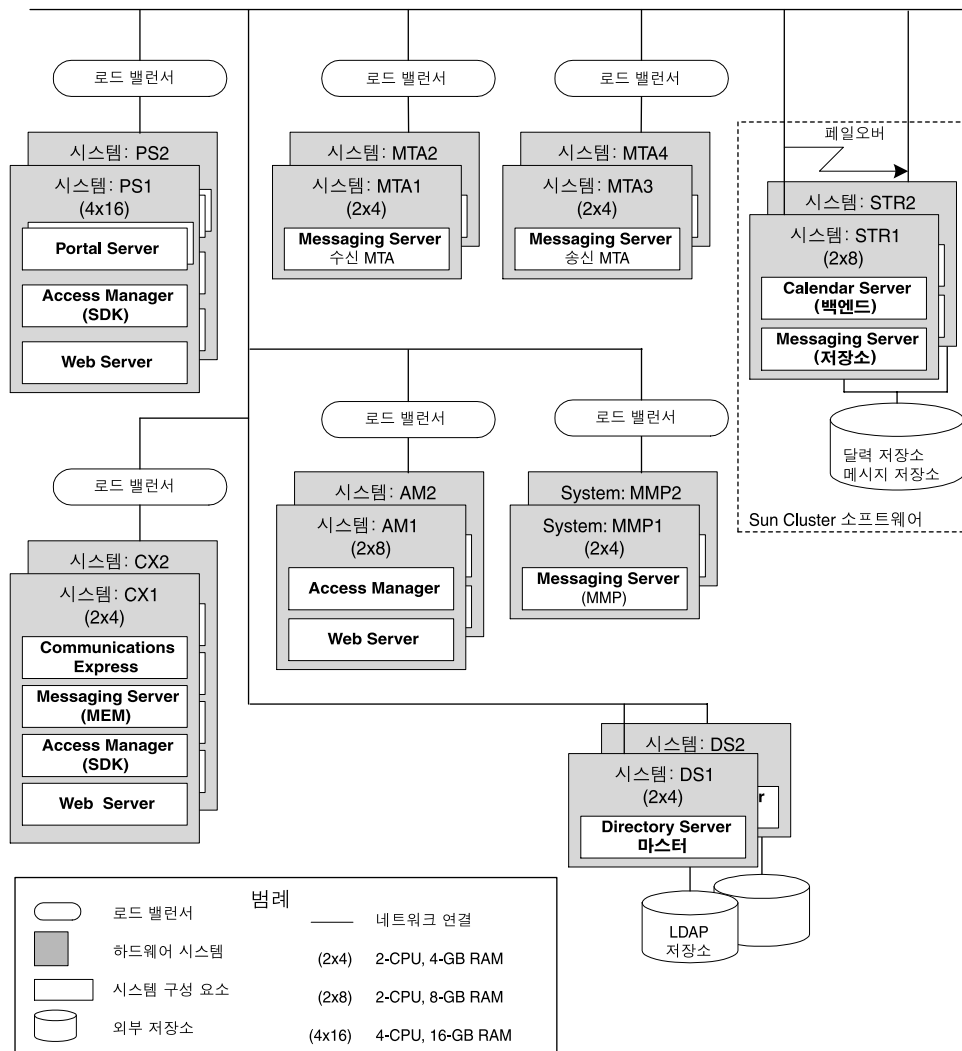


그림 5-10 예제 배포 구조

배포 설계 구현

솔루션 라이프 사이클의 구현 단계 중에 배포 설계 중 만들어진 사양 및 계획을 작업하여 궁극적으로 배포가 작업 환경으로 실행될 수 있도록 배포 구조를 구축하고 테스트합니다. 구현은 이 설명서 범위 밖에 있지만 이 장은 이러한 단계에 대한 높은 수준의 시각을 제공합니다.

이 장의 내용은 다음과 같습니다.

- 91 페이지 “배포 설계 구현 정보”
- 92 페이지 “소프트웨어 설치 및 구성”
- 92 페이지 “파일럿 및 프로토타입 개발”
- 93 페이지 “파일럿 및 프로토타입 배포 테스트”
- 94 페이지 “프로덕션 배포 실행”

배포 설계 구현 정보

배포 구조를 승인하고 구현 사양 및 계획을 완료하고 나면 솔루션 라이프 사이클의 구현 단계로 들어갑니다. 구현은 일련의 복잡한 과정 및 절차로서 성공하려면 신중하게 계획해야 합니다. 구현에는 다음과 같은 작업이 있습니다.

- 네트워크 및 하드웨어 인프라 구축
- 설치 계획에 따른 소프트웨어 설치 및 구성
- 기존 응용 프로그램에서 현재 솔루션으로 데이터 마이그레이션
- 사용자 관리 구현 계획
- 테스트 계획에 따라 테스트 환경의 파일럿 또는 프로토타입 설계 및 배포
- 테스트 계획에 따라 기능 테스트 및 스트레스 테스트 설계 및 실행
- 실행 계획에 따라 테스트 환경에서 작업 환경으로 솔루션 실행
- 교육 계획에 따라 배포 사용자 및 관리자 교육

구현에 대한 자세한 설명은 이 설명서 범위 밖에 있습니다. 그러나 다음 절에서 이러한 작업에 대한 개요 정보를 제공합니다.

소프트웨어 설치 및 구성

분산 엔터프라이즈 응용 프로그램을 위한 Sun Java™ Enterprise System의 설치 및 구성에는 많은 작업 및 절차 계획과 조정이 필요합니다. 배포 설계 단계 중에 Java Enterprise System 소프트웨어를 설치하는 데 필요한 설치 및 구성 정보를 제공하는 높은 수준의 배포 구조를 기반으로 설치 계획을 만듭니다.

이 설치 계획의 중요 부분은 다음과 같습니다.

- 설치 순서 및 유형 결정
- 이미 설치된 소프트웨어 및 설치 준비를 위한 호스트 검사
- 설치 중인 각 Java Enterprise System 구성 요소에 대한 구성 정보 수집

Sun Java Enterprise System 2005Q4 설치 계획 설명서는 설치 계획 정보를 수집하는 방법에 대해 자세히 설명합니다. **Sun Java Enterprise System 2005Q4 설치 참조 설명서**는 자세한 구성 정보와 이 정보를 문서화하는 데 사용할 수 있는 워크시트를 제공합니다. **Sun Java Enterprise System 2005Q4 UNIX용 설치 설명서**는 또한 여러 개의 Java Enterprise System 구성 요소를 포함하는 일반 설치 시나리오에 대한 지침을 제공합니다. 자세한 내용은 **Sun Java Enterprise System 2005Q4 UNIX용 설치 설명서**의 1 장, “설치 준비”을 참조하십시오.

파일럿 및 프로토타입 개발

Java Enterprise System 배포는 대개 두 가지 범주 즉, 주로 Java Enterprise System와 함께 제공되는 서비스를 기반으로 하는 배포와 Java Enterprise System 서비스와 통합된 많은 사용자 정의 서비스가 필요한 배포로 구분됩니다. 전자 유형의 배포를 80:20 배포(Java Enterprise System에서 서비스의 80%를 제공)로 생각할 수 있고, 마찬가지로 전자를 20:80 배포로 생각할 수 있습니다.

80:20 배포의 경우 구현 단계 중에 대개 테스트 목적으로 파일럿 배포를 개발합니다. 80:20 배포의 경우 “뛰어난”인 기능을 제공하는 완벽한 Java Enterprise System 서비스를 사용하기 때문에 파일럿 배포는 개발, 테스트 및 수정 단계에서 프로덕션 개발로 비교적 빠르게 진행됩니다. 파일럿 배포는 솔루션 기능을 검증하면서 또한 시스템이 얼마나 잘 수행하는지에 대한 정보도 제공합니다.

한편 20:80 배포에서는 80:20 배포와 함께 제공되는 상호 운용성 내역이 없는 새로운 사용자 정의 서비스를 도입합니다. 이런 이유 때문에 작업 환경으로 가기 전에 엄격한 개발, 테스트, 수정 주기가 필요한 POC(Proof-of-concept) 배포인 프로토타입을 작성합니다. 프로토타입은 제안된 솔루션이 테스트 환경에서 문제를 얼마나 잘 해결하는 지 판별할 수 있게 합니다. 기능이 충분함을 프로토타입이 보여주고 나면 보다 엄격한 테스트로 이동하고 그 다음에 파일럿 배포로 이동할 수 있습니다.

주 - 실제 엔터프라이즈 배포는 필요한 사용자 정의 서비스 개발 양에 있어서 크게 다를 수 있습니다. 테스트를 목적으로 파일럿 및 프로토타입 배포를 어떻게 사용할 지는 배포의 복잡도와 특성에 따라 다릅니다.

파일럿 및 프로토타입 배포 테스트

파일럿 및 프로토타입 배포를 테스트하는 목적은 가능한 한 테스트 조건에서 배포가 시스템 요구 사항을 충족시키고 비즈니스 목표도 충족시키는지 여부를 판별하는 것입니다.

기능 테스트에서 모든 식별된 사용 사례를 기반으로 시나리오를 모델링하고 호환성을 측정하기 위한 매트릭스 집합을 개발하는 것이 이상적입니다. 기능 테스트에는 비즈니스 요구 사항을 충족시키는지 확인하기 위해 베타 사용자 그룹을 선택하기 위한 제한된 배포가 포함될 수도 있습니다.

스트레스 테스트에서는 최고 로드 시 성능을 측정합니다. 이 테스트에서는 대개 시뮬레이션된 일련의 환경을 사용하고 생성기를 로드하여 데이터 처리량 및 성능을 측정합니다. 배포의 시스템 요구 사항은 대개 스트레스 테스트를 설계 및 통과하기 위한 기본입니다.

주 - 기능 및 스트레스 테스트는 시스템 요구 사항이 제대로 정의될 수 없고 추정된 기본이 되는 이전 구현이 없으며 배포에 많은 양의 새로운 개발이 필요한 대형 배포에 특히 중요합니다.

테스트에서 배포 설계 사양의 문제점을 표시할 수 있고 배포를 작업 환경으로 실행하기 전에 일부 설계, 구축, 테스트 반복이 생길 수 있습니다. 프로토타입 배포 테스트 시 배포 설계의 문제를 발견할 수도 있습니다. 이 경우 솔루션 라이프 사이클의 초기 단계로 돌아가서 문제를 해결할 수 있습니다.

파일럿 배포로 진행하기 전에 배포 설계를 완전히 테스트했는지 확인합니다. 파일럿 배포는 초기 테스트의 배포 설계를 이미 검증했음을 나타냅니다. 파일럿 배포 테스트 중에 해결하지 못한 문제는 일반적으로 배포 설계 매개 변수 내에서 처리되어야 합니다.

왜냐하면 테스트가 작업 환경을 완전하게 시뮬레이션할 수 없으며 배포된 솔루션의 특성이 전개 및 변화할 수 있기 때문에 배포 시스템을 계속 모니터링하여 조정, 유지 보수 또는 서비스가 필요한 모든 영역을 식별해야 합니다.

프로덕션 배포 실행

파일럿이나 POC 배포가 테스트 기준을 통과하면 배포를 작업 환경으로 실행할 준비가 된 것입니다. 일반적으로 프로덕션 환경 단계로 실행합니다. 준비된 실행은 많은 사용자에게 영향을 미치는 대형 배포에 특히 중요합니다.

준비된 배포는 적은 사용자로 시작할 수 있고 배포를 모든 사용자가 사용할 수 있을 때까지 사용자 기반을 확장할 수 있습니다. 준비된 배포를 제한된 서비스로 시작하여 나머지 서비스를 단계별로 실행할 수 있습니다. 서비스를 단계별로 실행하면 작업 환경에서 서비스에 발생할 수 있는 문제점을 격리, 식별 및 해결하는 데 도움이 될 수 있습니다.

색인

번호와 기호

- 20\\
 - 80 배포, 19
 - 구현 단계, 92
- 3차원 구조, 45
- 80\\
 - 20 배포, 19, 92

A

- Access Manager, 48, 71
- Application Server, 48

C

- Calendar Server, 48, 71
- Communications Express, 48

D

- Directory Proxy Server, 48, 52
- Directory Server, 48, 55, 71
 - 다중 마스터 복제, 81, 82-83
 - 단일 마스터 복제, 81
- DMZ, 외부 액세스 영역, 63

I

- Instant Messaging, 48

J

- Java Enterprise System
 - 20\\
 - 80 배포, 19
 - 3차원 구조, 45
 - 80\\
 - 20 배포, 19
 - 구성 요소, 47-53
 - 구성 요소 종속성, 48-51
 - 마이그레이션 문제, 19-20
 - 사용자 정의 서비스, 19
 - 서비스, 19
 - 설치, 92
 - 시스템 서비스, 17-19
 - 액세스 구성 요소, 52
 - 정보, 17
 - 프로덕션 배포 실행, 94
 - Java Enterprise System 설치, 92

M

- Message Queue, 48
- Messaging Server, 48
 - Message Multiplexor(MMP), 51, 52, 55, 71
 - Message Transfer Agent(MTA), 51
 - Messenger Express Multiplexor(MEM), 51
 - 논리적 구조 예, 54-59
 - 논리적으로 별개인 서비스, 51-52
 - 로드 균형 조정 예, 79-80
 - 메시지 저장소(STR), 51, 55, 71
 - 메시지 전송 에이전트(MTA), 55

N

N+1 페일오버 시스템, 77-78

P

Portal Server, 48, 52
Mobile Access, 52
보안 원격 액세스, 48, 52

Q

QoS(서비스 품질 요구 사항), 36-42

S

SLA, 30
Sun Cluster 소프트웨어, 79
페일오버 예, 80-81

W

Web Server, 48, 71

가

가용성
N+1 페일오버 시스템, 77-78
로드 균형 조정, 76
서비스 복제, 76
서비스 품질 요구 사항, 37-39
수평으로 중복된 시스템, 77-78
예, 79-83
우선 순위 지정, 38-39
자원 최적화, 88
페일오버, 76
가용성 전략, 결정, 76-83

고

고장 허용 시스템, 38

교

교육 계획, 67

구

구성 요소 종속성, 48-51
웹 컨테이너 지원, 51
구현 단계, 23-24, 92
정보, 91
파일럿 및 프로토타입 개발, 92-93
구현 사양, 66-67

규

규정 요구 사항, 29

기

기능 테스트, 93
기술 요구 사항
가용성, 37-39
보안, 40-41
서비스 수준 요구 사항, 43
성능, 37
실용성, 42
잠재 용량, 41
확장성, 39-40
기술 요구 사항 단계, 22
사용 분석, 34-35
사용 사례, 35-36
서비스 품질 요구 사항, 36-42
정보, 33-34
기업 문화, 28-29

내

내부 액세스 영역(인트라넷), 63

논

논리적 계층, 다중 계층 구조 모델, 53
논리적 구조, 45-46
배포 설계에 영향을 미침, 67

논리적 구조 (계속)
설계, 46-47
신원 기반 통신 예, 59
예, 54-62
논리적 설계, 정보, 45-46
논리적 설계 단계, 22-23

다

다중 계층 구조 설계, 52-53
다중 마스터 복제, 81
예, 82-83

단

단일 마스터 복제, 81
예, 81-82

데

데이터 계층, 다중 계층 구조 모델, 53

로

로드 균형 조정, 76
로드 균형 조정 예, 예, 78

롤

롤아웃 계획, 66

마

마이그레이션 계획, 66
마이그레이션 문제, 19-20
비즈니스 제약 조건, 30

메

메시징 서버, 사용 사례, 56-59

배

배포 계획
솔루션 라이프사이클, 20-21
정보, 20-24
증분 방법, 29
배포 구조, 66-67
예, 89-90
배포 설계
방법, 68
요소, 67
정보, 65-67
출력, 66-67
프로젝트 승인, 66
배포 설계 단계, 23
배포 시나리오, 45, 64, 65

병

병목 현상 식별, 배포 설계, 68

보

보안
서비스 품질 요구 사항, 40-41
자원 최적화, 88
프로세서 요구 사항 예상, 68
보안 액세스 영역, 64

비

비즈니스 목표
배포 설계에 영향을 미침, 67
정의, 26-27
비즈니스 분석 단계, 22
정보, 25
비즈니스 서비스 계층, 다중 계층 구조 모델, 53
비즈니스 요구 사항
규정 요구 사항, 29
기업 문화, 28-29
보안 목표, 29
비즈니스 목표, 26-27
사용 패턴, 28
사용자 이해, 27-28
서비스 수준 계약, 30
운영 상 요구 사항, 27-28

비즈니스 요구 사항 (계속)
정의, 26-30
비즈니스 제약 조건, 30-31
마이그레이션 문제, 30
소유 비용, 31
예산 제한, 31
일정 요구 사항, 30

사

사양 계획, 66-67
사용 분석, 34-35
 배포 설계에 영향을 미침, 67
사용 사례, 35-36
 메시징 서버 예, 56-59
 배포 설계에 영향을 미침, 67
 신원 기반 통신 예, 60-62
 프로세서 요구 사항 예상, 71-72
사용 패턴, 28
사용자 관리 계획, 66

서

서비스 복제, 68
 Directory Server 예, 81
 가용성 전략, 76
서비스 수준 계약, 30
 배포 설계에 영향을 미침, 67
 요구 사항, 43
서비스 수준 요구 사항, 41, 43
서비스 품질 요구 사항, 36-42
 배포 설계에 영향을 미침, 67
 배포 설계에서의 역할, 65

설

설명서
 기술 개요, 19, 46, 47
 설치 설명서, 49, 92
설치 계획, 66

성

성능
 병목 현상 식별, 85-87
 서비스 품질 요구 사항, 37
 자원 최적화, 88

소

소유 비용, 31
 배포 설계에 영향을 미침, 67

솔

솔루션 라이프사이클, 20-21
 구현 단계, 23-24, 91
 기술 요구 사항 단계, 22, 33-34
 논리적 설계 단계, 22-23, 45-46
 배포 설계 단계, 23, 65-67
 비즈니스 분석 단계, 22, 25
 운영 단계, 24

수

수평으로 중복된 시스템, 77-78

스

스트레스 테스트, 93

신

신원 기반 통신 예, 59-62
 사용 사례, 60-62
 프로세서 요구 사항 예상, 69

실

실용성
 서비스 수준 요구 사항, 42
 자원 최적화, 88
실행서, 67

액

액세스 영역, 62-64

예

예

Directory Server, 81
Messaging Server 논리적 구조, 54-59
가용성 설계, 79-83
논리적 구조, 54-62
다중 마스터 복제, 82-83
단일 마스터 복제, 81-82
로드 균형 조정, 79-80
로드 균형 조정 예, 78
배포 구조, 89-90
보안 트랜잭션을 위한 프로세서 요구 사항
 예상, 74-75
서비스 복제, 81
신원 기반 통신, 59-62
액세스 영역, 62
페일오버, 80-81
프로세서 요구 사항 예상, 69-73
확장성, 84-85
예산 제한, 31

외

외부 액세스 영역(DMZ), 63

용

용어집, 링크, 13

운

운영 계획(실행서), 67
운영 단계, 24
운영 상 요구 사항, 27-28

위

위기 관리, 88-89
 배포 설계, 68

일

일정 요구 사항, 30

자

자원 최적화, 배포 설계, 68

잡

잠재 용량, 41
 확장성 고려 사항, 83-84

재

재난 복구 계획, 66

최

최적화
 디스크 액세스, 86-87
 자원 사용, 87-88

클

클라이언트 계층, 다중 계층 구조 모델, 53

테

테스트
 기능 테스트, 93
 스트레스 테스트, 93
 파일럿 및 프로토타입, 93
테스트 계획, 66

파

파일럿, 92-93
 테스트, 93

페

- 페일오버, 76
- Sun Cluster 소프트웨어, 79
- 예, 80-81

프

- 프로세서 요구 사항, 예상, 68-73
- 프로세서 요구 사항 예상, 68
 - 보안 트랜잭션, 73-75
 - 사용 사례, 71-72
 - 예, 69-73
- 프로젝트 승인, 66
- 프로토타입, 92-93
 - 테스트, 93
- 프리젠테이션 계층, 다중 계층 구조 모델, 53

확

- 확장성
 - 서비스 품질 요구 사항, 39-40
 - 예, 84-85
 - 자원 최적화, 88
 - 전략, 83-85
 - 증가 예상, 39-40