



Sun Java™ System

Sun Java Enterprise System

배포 계획 백서

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

부품 번호 : 817-7593

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 모든 권리는 저작권자의 소유입니다 . Sun Microsystems, Inc. 는 본 설명서에 설명된 제품에 포함된 기술 관련 지적 재산을 소유합니다 . 특히 이 지적 재산권에는 <http://www.sun.com/patents>에 나열된 하나 이상의 미국 특허권이 포함될 수 있으며 , 미국 및 다른 국가에서 하나 이상의 추가 특허권 또는 출원 중인 특허권이 제한 없이 포함될 수 있습니다 .

이 제품에는 SUN MICROSYSTEMS, INC. 의 기업 기밀 정보가 포함되어 있습니다 . SUN MICROSYSTEMS, INC. 의 명시적인 사전 서면 승인 없이 사용 , 공개 또는 복제하는 것을 금합니다 .

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

배포본 중에는 타사에서 개발한 자료가 포함되어 있을 수 있습니다 .

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

Sun, Sun Microsystems, Sun 로고 , Java, Solaris, JDK, Java Naming 및 Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, Duke 로고 , Java Coffee Cup logo, Solaris 로고 , SunTone Certified 로고 및 Sun ONE 로고는 미국 및 다른 국가에서 Sun Microsystems, Inc. 의 상표 또는 등록 상표입니다 .

모든 SPARC 상표는 사용 허가를 받았으며 미국 및 다른 국가에서 SPARC International, Inc. 의 상표 또는 등록 상표입니다 . SPARC 상표를 사용하는 제품은 Sun Microsystems, Inc. 에서 개발한 구조에 기반을 두고 있습니다 .

Legato 및 Legato 로고는 Legato Systems, Inc. 의 등록 상표이며 Legato NetWorker 는 Legato Systems, Inc. 의 상표 또는 등록 상표입니다 .

Netscape Communications Corp 로고는 Netscape Communications Corporation 의 상표 또는 등록 상표입니다 .

OPEN LOOK 및 Sun(TM) GUI (그래픽 사용자 인터페이스) 는 Sun Microsystems, Inc. 가 자사의 사용자 및 정식 사용자로 개발했습니다 . Sun 은 컴퓨터 업계를 위한 시각적 또는 그래픽 사용자 인터페이스의 개념을 연구 개발한 Xerox 사의 선구적인 노력을 높이 평가하고 있습니다 . Sun 은 Xerox 와 Xerox Graphical User Interface 에 대한 비독점적 사용권을 보유하고 있습니다 . 이 사용권은 OPEN LOOK GUI 를 구현하는 Sun 의 정식 사용자에게도 적용되며 그렇지 않은 경우에는 Sun 의 서면 사용권 계약을 준수해야 합니다 .

이 서비스 설명서에서 다루는 제품과 수록된 정보는 미국 수출 관리법에 의해 규제되며 다른 국가의 수출 또는 수입 관리법의 적용을 받을 수도 있습니다 . 이 제품과 정보를 직간접적으로 핵무기 , 미사일 또는 생화학 무기에 사용하거나 핵과 관련하여 해상에서 사용하는 것은 엄격하게 금지합니다 . 미국 수출 금지 국가 또는 금지된 개인과 특별히 지정된 국민 목록을 포함하여 미국 수출 금지 목록에 지정된 대상으로의 수출이나 재수출은 엄격하게 금지됩니다 .

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 .

목차

그림 목차	7
표 목차	9
1 장 배포 계획 소개	11
Java Enterprise System 정보	12
Java Enterprise System 서비스 제품군	13
Java Enterprise System 의 이점	15
배포 계획 정보	17
비즈니스 분석 단계	18
기술 요구 사항 단계	18
논리적 설계 단계	18
배포 설계 단계	19
구현 단계	19
2 장 비즈니스 분석	21
비즈니스 요구 사항	21
비즈니스 제약 조건	24
배포에 대한 증분 방법	25
3 장 기술 요구 사항	27
사용 분석	28
사용 사례	30
시스템 요구 사항	31
가용성	32
내결합성이 있는 시스템	32
Sun Cluster 3.1 4/04	33
서비스 가용성 우선 순위 지정	33
잠재 용량	34
성능	34

확장성	34
보안 요구 사항	36
인증	37
권한 부여	37
아이디 관리	37
서비스 가능성 요구 사항	37
서비스 수준 요구 사항	39
4 장 논리 구조 설계	41
배포 계획 예	42
Java Enterprise System 서비스	43
예제 배포의 논리 구조	47
예제 배포의 데이터 흐름	48
배포 시나리오	49
5 장 배포 구조 설계	51
계획된 배포의 크기 조정	52
성능을 위한 크기 조정	52
사용자 시작점에 대한 기본 CPU 예상치 결정	53
서비스 종속성에 대해 CPU 예상 개수 조정	54
잠재 용량, 확장성 및 가용성에 대한 CPU 예상 개수 조정	56
보안을 위한 크기 조정	58
보안 트랜잭션에 대한 성능 계산	58
SSL 트랜잭션을 처리하는 특수 하드웨어	60
가용성을 위한 크기 조정	60
복잡한 시스템의 디렉토리 설계	61
하드웨어 및 소프트웨어 오류	61
가용성에 대한 일반적인 접근 방법	61
샘플 배포의 가용성 설계	64
서비스 가능성 문제점	65
확장성을 위한 크기 조정	66
잠재 용량	66
시스템의 용량 업그레이드	66
자원 최적화	67
위험 관리	67
자원 관리	67
예제 배포 구조	69
세부 설계 지정	71
6 장 배포 설계 구현	73
파일럿 및 프로토타입 개발	73
파일럿 및 프로토타입 배포 테스트	74

프로덕션 배포 실행	75
색인	77

그림 목차

그림 1-1	배포 계획 단계	17
그림 3-1	기술 요구 사항 단계와 다른 배포 계획 단계	28
그림 4-1	논리 설계와 다른 배포 계획 단계와의 관계	42
그림 4-2	Java Enterprise System 구성 요소	44
그림 4-3	논리 구조의 Java Enterprise System 구성 요소	47
그림 4-4	예제 배포에 대한 데이터의 논리적 흐름	48
그림 5-1	사용자 시작점을 제공하는 구성 요소에 대한 기본 CPU 예상 개수	54
그림 5-2	지원 서비스에 대해 조정된 CPU 예상 개수	56
그림 5-3	메모리 요구 사항을 포함한 성능 수치	57
그림 5-4	보안 트랜잭션의 CPU 예상 개수를 계산하기 위한 워크시트	59
그림 5-5	단일 서버	61
그림 5-6	2 개의 복제 서버	62
그림 5-7	두 서버 간의 로드 배포	62
그림 5-8	n 개의 서버 간 로드 배포	63
그림 5-9	예제 배포에서 Calendar Server 에 대한 가용성 설계	65
그림 5-10	예제 배포 구조	70

표 목차

표 1-1	Java Enterprise System 구성 요소	12
표 1-2	Java Enterprise System 서비스 제품군	14
표 1-3	Java Enterprise System 이점	15
표 2-1	비즈니스 요구 사항 분석 항목	22
표 2-2	비즈니스 제약 조건 분석 항목	24
표 3-1	사용 분석 항목	29
표 3-2	배포 설계에 영향을 미치는 시스템 품질	31
표 3-3	1 년 내내 (8,760 시간) 실행되는 시스템의 중단 시간	32
표 3-4	서비스 가용성 우선 순위 지정	33
표 3-5	확장성 고려 사항	35
표 3-6	서비스 가능성 요구 사항 항목	38
표 4-1	예제 배포의 사용 사례	43
표 4-2	Java Enterprise System 구성 요소 상호 종속성	45
표 4-3	예제 사용 사례를 지원하기 위한 Java Enterprise System 구성 요소	46
표 4-4	예제 사용 사례를 지원하기 위한 추가 구성 요소	46
표 5-1	지원 서비스에 대한 CPU 예상 개수	55
표 5-2	자원 관리 항목	68

배포 계획 소개

이 백서에서는 Sun Java™ Enterprise System 을 기반으로 한 대형 배포를 소개합니다. 배포 계획의 기본적인 개념과 원칙을 제공하고 엔터프라이즈 차원의 배포를 설계할 때 기준으로 삼을 수 있는 여러 프로세스를 소개합니다.

Java Enterprise System 을 평가하거나 Java Enterprise System 을 기반으로 대형 응용 프로그램을 작성 및 배포할 계획을 할 경우 이 백서를 기준으로 배포 계획을 세우십시오.

이 장에서는 Java Enterprise System 의 간단한 개요를 제공하고 나중 장에서 설명하는 배포 계획에 대한 개념을 소개합니다. 이 장의 내용은 다음과 같습니다.

- 12 페이지의 “Java Enterprise System 정보 ”
- 17 페이지의 “배포 계획 정보 ”

Java Enterprise System 정보

Java Enterprise System 은 네트워크나 인터넷 환경을 통해 배포된 엔터프라이즈급 응용 프로그램을 지원하는 서비스를 제공하는 소프트웨어 인프라입니다. 다음 표에서는 Java Enterprise System 의 구성 요소와 구성 요소에서 제공하는 인프라 서비스를 나열합니다.

표 1-1 Java Enterprise System 구성 요소

시스템 구성 요소	제공 서비스
Application Server	Session Bean, Entity Bean 및 Message-Driven Bean 같은 Enterprise JavaBeans™ (EJB) 구성 요소용 Java 2 Platform, Enterprise Edition (J2EE™ 플랫폼) 컨테이너 서비스를 제공합니다. 컨테이너는 밀접하게 연결된 분산 구성 요소 간 상호 작용에 필요한 인프라 서비스를 제공하여 전자 상거래 응용 프로그램과 웹 서비스의 개발 및 실행을 위한 플랫폼이 됩니다. Application Server 에서도 웹 컨테이너 서비스를 제공합니다.
Calendar Server	최종 사용자와 최종 사용자 그룹에게 달력 및 일정 서비스를 제공합니다. Calendar Server 는 서버와 상호 작용하는 브라우저 기반 클라이언트를 포함합니다.
Directory Proxy Server	회사 외부의 방화벽에서 Directory Server 용 보안 서비스를 제공합니다. Directory Proxy Server 는 여러 Directory Server 인스턴스에 대해 향상된 디렉토리 액세스 제어, 스키마 호환성, 라우팅 및 로드 균형 조정을 제공합니다.
Directory Server	아이디 프로필 (직원, 고객, 공급자 등), 사용자 자격 증명 (공개 키 인증서, 비밀번호, PIN 번호), 액세스 권한, 응용 프로그램 자원 정보 및 네트워크 자원 정보 등과 같은 인터넷 및 인터넷 정보를 저장 및 관리하기 위한 중앙 저장소를 제공합니다.
Identity Server	액세스 관리 및 디지털 아이디 관리 서비스를 제공합니다. 액세스 관리 서비스는 응용 프로그램 또는 서비스에 대한 액세스를 위한 인증 (단일 사인 온 포함) 및 역할 기반 권한 부여를 포함하고 관리 서비스는 개별 사용자 프로필, 역할, 그룹 및 정책의 중앙 집중화된 관리를 포함합니다.
Instant Messaging	인스턴트 메시징 (채팅), 회의, 경고, 뉴스, 폴링 및 파일 전송 등의 최종 사용자 간에 실시간 보안 통신을 제공합니다. 서비스에는 현재 온라인 상태인 사용자를 알려 주는 존재 관리자 그리고 서버와 상호 작용하는 브라우저 기반 클라이언트가 포함되어 있습니다.
Message Queue	느슨하게 연결된 분산된 구성 요소와 응용 프로그램 간에 신뢰할 수 있는 비동기 메시지를 제공합니다. Message Queue 는 Java Message Service (JMS) API 사양을 구현하고 보안, 확장성 및 원격 관리 같은 엔터프라이즈 기능을 추가합니다.

표 1-1 Java Enterprise System 구성 요소 (계속)

시스템 구성 요소	제공 서비스
Messaging Server	전자 메일, 팩스, 호출기, 음성 및 비디오를 지원하는 안전하고 신뢰할 수 있는 대용량의 저장 후 전달 (store-and-forward) 메시징을 제공합니다. 여러 메일 저장소를 동시에 액세스하고 임의의 전자 메일을 거부하고 바이러스 공격을 방지하는 데 도움을 주는 내용 필터링을 제공합니다.
Portal Server	비즈니스 응용 프로그램이나 서비스를 액세스하는 브라우저 기반 클라이언트에 내용 집계 및 개인화 같은 주요 포털 서비스를 제공합니다. Portal Server 에서는 구성 가능한 검색 엔진도 제공합니다.
Secure Remote Access	회사 외부의 방화벽에서 내부 포털 및 인터넷 응용 프로그램을 포함한 Portal Server 콘텐츠와 서비스에 대해 보안 인터넷 액세스를 제공합니다.
Web Server	Java Servlet 및 JavaServer Pages™ (JSP™) 구성 요소 같은 Java 웹 구성 요소용 J2EE 플랫폼 웹 컨테이너 서비스를 제공합니다. Web Server 에서는 CGI 스크립트 및 Active Server Pages 같은 정적 및 동적 웹 콘텐츠를 제공하기 위한 다른 웹 응용 프로그램 기술도 지원합니다.
Sun Cluster	Java Enterprise System 용고가용성 및 확장성 서비스, Java Enterprise System 인프라 위에서 실행되는 응용 프로그램 및 이 둘이 배포되는 하드웨어 환경을 제공합니다.

Java Enterprise System 서비스 제품군

Java Enterprise System 배포는 일반적으로 Java Enterprise System에서 제공하는 서비스로 주로 구성되는 범주와 상당수의 사용자 정의 개발 서비스 및 타사 응용 프로그램을 통합하는 범주의 두 가지 일반 범주에 해당합니다. 전자의 배포 유형은 80:20 배포 (80%의 Java Enterprise System 제공 서비스)로, 후자는 20:80 배포로 주할 수 있습니다.

주 실제 엔터프라이즈 배포는 필요로 하는 사용자 정의 개발 서비스의 양에 따라 매우 다양할 수 있습니다.

Java Enterprise System 은 서비스 세트가 풍부하기 때문에 80:20 배포에 특히 적합합니다. 예를 들어, 엔터프라이즈 전체의 통신 시스템이나 엔터프라이즈 전체의 포털 시스템을 배포하는 것이 비교적 쉽습니다.

그러나 사용자 정의 개발이 필요한 배포의 경우 Java Enterprise System 은 사용자 개발 서비스와 응용 프로그램을 작성 및 통합할 수 있는 기능을 제공합니다.

다음 표에서는 Java Enterprise System 구성 요소를 엔터프라이즈 배포를 제공할 수 있는 제품군으로 그룹화합니다. 일부 구성 요소는 둘 이상의 제품군에 속합니다.

표 1-2 Java Enterprise System 서비스 제품군

제품군	Java Enterprise System 구성 요소
네트워크 Identity 서비스	Identity Server Directory Server Web Server
엔터프라이즈 포털 서비스	Portal Server Secure Remote Access Identity Server Directory Server Application Server 또는 Web Server
엔터프라이즈 통신 및 공동 작업 서비스	Messaging Server Calendar Server Instant Messaging Identity Server Directory Server Application Server 또는 Web Server
웹 및 응용 프로그램 서비스	Application Server Message Queue Web Server
가용성 서비스	Sun Cluster 3.1 4/04 Sun Cluster Agents

위 표 1-2의 제품군은 대부분 80:20 배포 유형을 제공할 수 있습니다. 예를 들어, 엔터프라이즈 통신 및 공동 작업 서비스는 최종 사용자에게 전자 메일, 달력 및 인스턴트 메시징 서비스를 제공하여 콘텐츠를 집계 및 개인화할 수 있도록 하는 배포를 작성하는 데 사용할 수 있습니다. 마찬가지로 네트워크 Identity 및 엔터프라이즈 포털 제품군을 사용하면 사용자 정의 서비스를 개발하거나 통합하지 않고도 엔터프라이즈 차원의 응용 프로그램을 설치 및 구성할 수 있습니다.

가용성 서비스 제품군은 엔터프라이즈 응용 프로그램의 대형 배포에 대해 고가용성을 제공합니다. 엔터프라이즈 응용 프로그램이 응용 프로그램 서버나 웹 서버에서 실행되는 J2EE 플랫폼 서비스의 사용자 정의 개발을 필요로 하는 경우 웹 및 응용 프로그램 서비스 제품군을 사용합니다.

Java Enterprise System 서비스 간에 상호 운용성이 있기 때문에 사용자의 특정 엔터프라이즈 요구 사항에 맞게 고유한 서비스 제품군을 작성할 수 있습니다.

Java Enterprise System 의 이점

엔터프라이즈 배포 성공의 핵심 요소는 다음 세 가지입니다.

- 제공 시기
- 제공 비용
- 기능

다음 표에 표시된 대로 Java Enterprise System 은 이러한 각 성공 요소를 해결하기 위한 도구를 제공합니다.

표 1-3 Java Enterprise System 이점

이점	설명
간편한 사용	<p>Java Enterprise System 에서는 쉽게 설치, 구성 및 업그레이드할 수 있게 해주는 공통된 설치 프로그램을 제공합니다.</p> <p>Java Enterprise System 에서는 독립적으로 개발된 단일 제품과 미들웨어의 통합으로부터 별다른 사용자 정의 없이 배포 및 구성할 수 있는 통합된 플랫폼 서비스 시스템으로 전환할 수 있습니다.</p>
예측 가능	<p>Java Enterprise System 릴리스 주기는 Java Enterprise System 구성 요소 간의 호환성을 고려해 결정됩니다. 새 릴리스로 업그레이드할 때 구성 요소 간의 비호환성과 잘못 맞춤을 방지할 수 있습니다.</p> <p>Java Enterprise System 구성 요소에서는 서비스 간의 상호 운용을 용이하게 하는 공유 플랫폼 구성 요소 집합을 사용합니다.</p> <p>Java Enterprise System 의 예약된 릴리스 제공 모델은 배포 계획을 예측 가능하게 해줍니다.</p>

표 1-3 Java Enterprise System 이점 (계속)

이점	설명
저렴함	상용 라이선스에 대한 Java Enterprise System 단일 단위 가격 모델은 배포 설치 및 업그레이드의 복잡함과 비용을 줄입니다. 단일 단위 가격에는 지원, 유지 보수 및 컨설팅 서비스가 포함됩니다. OEM 및 교육 라이선스용의 다른 가격 모델도 사용할 수 있습니다.

배포 계획 정보

성공적인 배포 계획은 다음 그림에서 설명하는 일련의 단계를 통한 세밀한 준비, 분석 및 설계로 만들 수 있습니다.

그림 1-1 배포 계획 단계

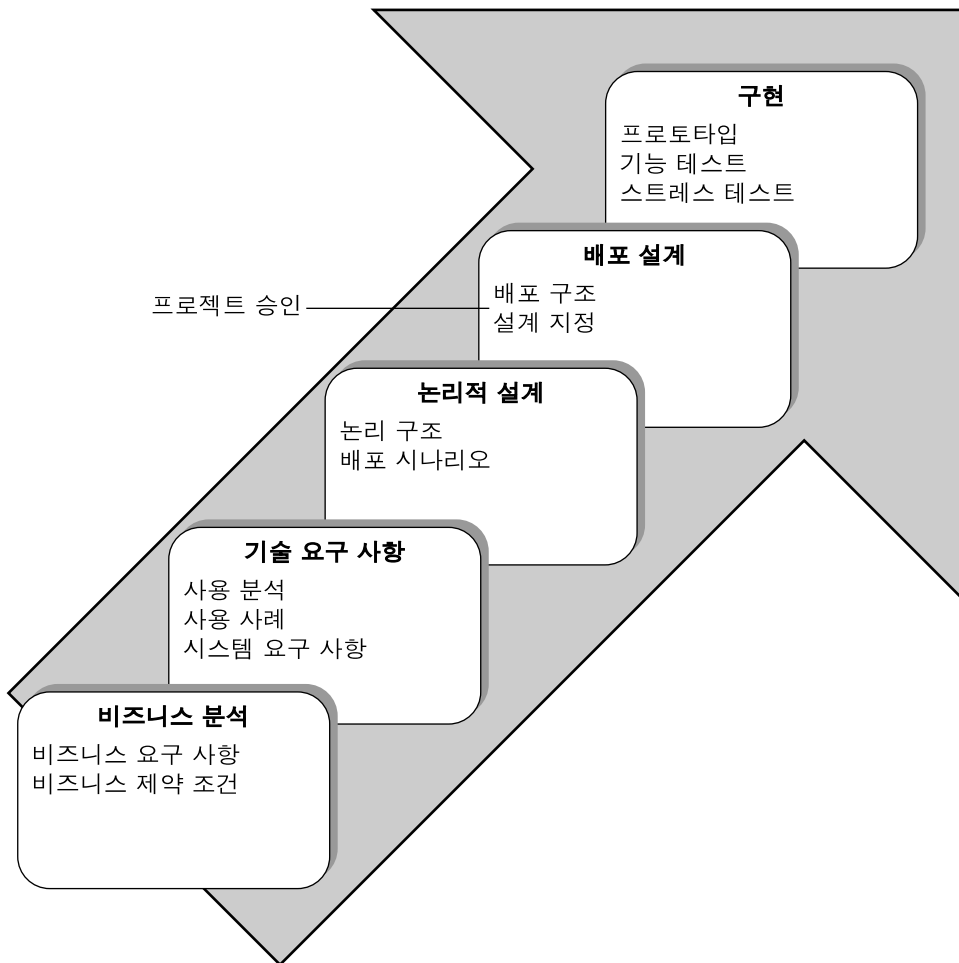


그림 1-1에서 설명하는 각 단계에는 후속 단계로 전달되는 사양 및 설계를 만드는 자체의 분석 및 절차 집합이 포함되어 있습니다. 이 장의 다음 절에서는 각 배포 계획 단계에 대한 요약 설명을 제공합니다.

비즈니스 분석 단계

비즈니스 분석 단계 중에는 배포 프로젝트의 비즈니스 목표를 정의하고 해당 목표를 달성하려면 충족시켜야 하는 비즈니스 요구 사항을 기술합니다. 비즈니스 요구 사항을 기술할 때 비즈니스 목표를 달성할 수 있는 기능에 영향을 줄 수 있는 비즈니스 제약 조건을 고려합니다. 비즈니스 분석 단계에서 *비즈니스 요구 사항 문서*가 만들어지며 이 문서는 나중에 기술 요구 사항 단계에서 사용되고 배포 설계의 성공 여부를 측정하는 기준으로 사용됩니다.

비즈니스 분석 단계에 대한 자세한 내용은 [21 페이지의 2장](#), “비즈니스 분석”을 참조하십시오.

기술 요구 사항 단계

기술 요구 사항 단계는 비즈니스 분석 단계 중에 작성한 비즈니스 요구 사항으로 시작되며, 이 요구 사항을 배포 구조를 설계하는 데 사용할 수 있는 기술 사양으로 변환합니다. 기술 요구 사항 단계 동안 다음 정보를 준비합니다.

- 사용자 작업 및 사용 패턴 분석
- 사용자와 계획된 배포의 상호 작용을 모델링하는 사용 사례
- 사용자 작업 및 사용 패턴 분석을 고려하여 비즈니스 요구 사항에서 추출한 시스템 요구 사항

사용 분석, *사용 사례* 및 *시스템 요구 사항 문서*의 결과 집합이 논리적 설계 단계로 입력됩니다.

기술 요구 사항 분석 중에 배포된 시스템이 시스템 요구 사항을 충족시키지 못하는 문제를 해결하기 제공해야 하는 고객 지원의 계약 조건인 *서비스 수준 요구 사항*도 지정할 수 있습니다. 서비스 수준 요구 사항은 프로젝트 승인 시 서명하는 *서비스 수준 계약*의 기준입니다.

기술 요구 사항 단계에 대한 자세한 내용은 [27 페이지의 3장](#), “기술 요구 사항”을 참조하십시오.

논리적 설계 단계

배포 설계는 논리적 설계 단계에서 시작됩니다. 이 단계에서는 기술 요구 사항 단계 중에 식별한 사용 사례를 충족시키는 Java Enterprise System 서비스와 종속성을 나타내는 논리적 구조를 설계합니다.

논리적 구조는 시스템 요구 사항 문서와 함께 *배포 시나리오*를 결정합니다. 논리적 구조에서는 배포 시나리오를 구현하는 데 필요한 실제 하드웨어를 지정하지 않습니다.

논리적 설계 단계에 대한 자세한 내용은 41 페이지의 4장, “논리 구조 설계”를 참조하십시오.

배포 설계 단계

배포 설계 단계에서는 배포 시나리오를 물리적 환경에 매핑하는 것을 나타내는 *배포 구조*를 작성합니다. 물리적 환경은 배포에 대한 네트워크 인프라이며 컴퓨팅 노드, 각 노드에 대한 하드웨어 요구 사항, 방화벽 및 기타 네트워크 장치 등을 포함합니다.

매핑 프로세스는 시스템 요구 사항을 이행하는 데 필요한 실제 하드웨어를 지정하기 위한 배포의 크기 조정과 예산 고려 사항에 맞게 배포 구조를 최적화하기 위한 전략 결정으로 구성됩니다.

일반적으로 배포 프로젝트 승인은 배포 구조 작성 다음에 이루어집니다. *프로젝트 승인* 동안에는 배포 비용을 평가하고 승인된 경우 배포 구현을 위한 계약에 서명하고 프로젝트를 구성하는 자원을 구입합니다.

자세한 *설계 사양*도 배포 설계 단계의 일부입니다. 설계 사양에서는 실제 하드웨어, 운영 체제, 네트워크 설계 및 물리적 환경의 기타 요소와 같은 배포 구조를 구현하는 데 필요한 세부 사항을 제공합니다. 또한 자세한 설계 사양에서는 시스템 서비스에 액세스할 *사용자*를 지정하는 데 필요한 디렉토리 서비스 데이터 구조도 지정합니다. 배포 프로젝트의 프로세스 및 정책에 따라 설계 사양은 프로젝트 승인 전이나 후에 수행됩니다.

배포 설계 단계에 대한 자세한 내용은 51 페이지의 5장, “배포 구조 설계”를 참조하십시오.

구현 단계

구현 단계에서는 배포 구조를 작성합니다. 배포 프로젝트의 특성에 따라 이 단계에는 다음 단계 중 일부 또는 전부가 포함됩니다.

- 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

- 단계적으로 생산에 들어갈 수 있는 생산 배포 작성

배포가 생산에 들어가면 비즈니스 목표를 이행할 수 있도록 배포를 계속 모니터, 테스트 및 조정해야 합니다.

구현 단계에 대한 자세한 내용은 [73 페이지의 6 장](#), “[배포 설계 구현](#)”을 참조하십시오.

비즈니스 분석

이 장에서는 비즈니스 문제를 분석하고, 비즈니스 요구 사항 및 제약 조건을 식별하며, 비즈니스 목표를 표현하는 방법에 대한 지침을 제공합니다.

비즈니스 분석은 배포 프로젝트의 비즈니스 목표를 기술하는 것으로 시작됩니다. 그런 다음 해결해야 할 비즈니스 문제를 분석하고, 비즈니스 목표를 달성하기 위해 충족시켜야 할 *비즈니스 요구 사항*을 식별합니다. 목표를 달성할 수 있는 능력을 제한하는 *비즈니스 제약 조건*도 고려합니다. 식별하는 비즈니스 요구 사항 및 제약 조건은 나중에 기술 요구 사항 단계에서 시스템 요구 사항을 추출하는 데 사용하는 *비즈니스 요구 사항 문서*의 기준이 됩니다.

비즈니스 요구 사항을 식별하는 간단한 공식은 없습니다. 비즈니스 영역에 대한 지식을 기반으로 고객과 공동 작업을 통해 요구 사항을 결정합니다. 여기서 제공하는 지침은 비즈니스 분석을 시작할 수 있는 한 가지 방법일 따름입니다.

이 장의 내용은 다음과 같습니다.

- “비즈니스 요구 사항”
- 24 페이지의 “비즈니스 제약 조건”
- 25 페이지의 “배포에 대한 증분 방법”

비즈니스 요구 사항

비즈니스 문제 설명은 프로젝트의 최종 목표를 설명하는 프로젝트의 실행 요약과 유사합니다. 비즈니스 문제 설명에서는 프로젝트의 비즈니스 사례 (프로젝트가 필요한 이유 또는 하고자 하는 것)를 작성하고 프로젝트의 범위 (프로젝트의 경계 안팎에 있는 것)를 정의해야 합니다. 또한 프로젝트의 성공에 중요한 프로젝트 기능도 결정해야 합니다.

비즈니스 요구 사항 분석 결과는 배포를 통해 비즈니스 목표를 충족시키는 방법을 정의하는 문서가 됩니다. 다음 표에서는 비즈니스 요구 사항 분석에서 일반적으로 다루는 항목을 나열합니다.

표 2-1 비즈니스 요구 사항 분석 항목

항목	설명
비즈니스 목표	<p>프로젝트의 목표를 명확하게 표현합니다. 목표를 명확하게 알고 있으면 설계 결정에 집중할 수 있습니다.</p> <p>다음은 목표의 몇 가지 예입니다.</p> <ul style="list-style-type: none"> • 메시징, 주소록, 인스턴트 메시징 및 달력 서비스와 같은 기능을 포함한 엔터프라이즈 공동 작업 • 내용을 집계 및 개인화할 수 있도록 하고, 전자 메일, 달력, 인스턴트 메시징 및 기타 엔터프라이즈 서비스에 대한 액세스를 제공할 수 있는 엔터프라이즈 포털 • 회의실, 사무실 및 기타 공유하는 물리적 자원을 예약하는 엔터프라이즈 자원 스케줄러 • 온라인 전자 상거래 사용 가능 <p>계획한 배포 목표와 현재 운영을 비교하면 나중에 설계 결정을 내리는 데 도움이 될 수 있습니다.</p>
배포 유형	<p>그려보고 있는 배포의 유형을 다음 중에서 식별합니다.</p> <ul style="list-style-type: none"> • B2C (기업 - 사용자) • B2E (기업 - 직원) • B2B (기업 - 기업) • 엔터프라이즈 직원 대 직원 통신 • 이 유형의 조합 <p>배포 유형을 알고 있으면 해당 유형에 내재된 특정 설계 문제에 집중할 수 있습니다.</p>
범위	<p>프로젝트 범위를 명확하게 기술합니다. 해결할 수 있는 영역을 식별하고 목표를 불명확하거나 도달할 수 없게 만드는 "제한 없는" 설명을 방지하도록 합니다.</p> <p>범위를 제대로 정의하지 못하면 비즈니스 요구 사항을 불충분하게 처리하는 배포 설계가 만들어질 수 있습니다.</p>
이해 관계자	<p>배포 성공에 따라 수익을 얻는 개인이나 조직을 식별합니다.</p> <p>모든 이해 관계자가 비즈니스 목표와 요구 사항을 정의하는 데 적극적으로 참여해야 합니다.</p>
중요한 품질	<p>성공에 결정적인 역할을 하는 영역을 식별합니다. 이를 통해 가장 중요한 기준에 따라 설계를 분석할 수 있습니다.</p>

표 2-1 비즈니스 요구 사항 분석 항목 (계속)

항목	설명
대상 사용자	<p>배포 대상인 사용자의 유형을 식별합니다. 예를 들면 다음과 같습니다.</p> <ul style="list-style-type: none"> • 현재 및 이전 직원 • 적극적인 고객 • 구성원 사이트 • 일반 대중 • 관리자
사용자에 대한 혜택	<p>배포 사용자에게 예상되는 혜택을 기술합니다. 예를 들면 다음과 같습니다.</p> <ul style="list-style-type: none"> • 회사 자원에 대한 원격 액세스 • 엔터프라이즈 공동 작업 • 응답 시간 감소 • 오류 비율 감소 • 일일 작업 단순화 • 원격 팀 간 자원 공유 • 생산성 향상 <p>예상되는 혜택을 명확하게 기술하면 설계 결정을 추진하는 데 도움이 됩니다.</p>
서비스 수준 계약	<p>배포가 특정 시스템 요구 사항을 충족시키지 못할 경우 제공해야 할 고객 지원의 수준과 범위를 정의합니다.</p> <p>일반적으로 서비스 수준 계약은 기술 요구 사항 분석에서 정의되는 서비스 수준 요구 사항을 기준으로 프로젝트 승인 중에 서명됩니다.</p>
보안 문제	<p>앞에서 식별한 목표에는 문제 설명에서 나열할 필요가 없는 암시적인 보안 문제가 있을 수 있습니다. 그러나 배포에 필수적인 특정 보안 목표를 호명하는 것은 유용할 수 있습니다. 예를 들면 다음과 같습니다.</p> <ul style="list-style-type: none"> • 권한 있는 사용자에 대한 소유권 정보 액세스 • 기밀 정보에 대한 역할 기반 액세스 • 원격 위치 간의 보안 통신 • 로컬 시스템에서 원격 응용 프로그램 호출 • 타사 비즈니스와의 보안 트랜잭션
우선 순위	<p>목표의 우선 순위를 기술합니다.</p> <p>크고 복잡한 배포에는 단계적인 구현이 필요할 수 있습니다. 자원이 제한될 경우 일부 목표를 제거하거나 수정해야 할 수 있습니다. 우선 순위를 명확하게 기술하면 승인을 받기 위해 배포 설계에서 결정해야 할 사항에 대한 지침을 제공할 수 있습니다.</p>

비즈니스 제약 조건

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.

다음 표에서는 배포 설계에 영향을 미칠 수 있는 일반적인 비즈니스 제약 조건을 나열합니다. 개별 배포 프로젝트에는 각 상황에 관련된 특정한 비즈니스 제약 조건이 있을 수 있습니다.

표 2-2 비즈니스 제약 조건 분석 항목

항목	설명
시간 단위 또는 일정	<p>배포 일정은 설계 결정에 영향을 미칠 수 있습니다. 빡빡한 일정으로 인해 목표가 축소되거나, 우선 순위가 변경되거나, 증분 해결 방법이 채택될 수 있습니다.</p> <p>일정 내에는 고려할 만한 중요한 이정표가 있을 수도 있습니다.</p>
예산 고려 사항	<p>대부분의 배포는 특정 예산을 지켜야 합니다. 비용 초과를 방지하기 위해 설계 과정 중에 항상 이 예산을 고려해야 합니다.</p> <p>예산을 고려할 때 프로젝트 완료 비용뿐만 아니라 특정 수명 동안 프로젝트를 유지 보수하는 데 필요한 자원도 고려해야 합니다.</p>
자원	<p>자본 지출뿐 아니라 성공적인 배포에 필요한 모든 자원을 고려합니다. 여기에는 다음이 포함됩니다.</p> <ul style="list-style-type: none"> 기존 하드웨어 및 네트워크 인프라 기존 인프라에 대한 의존이 시스템 설계에 영향을 미칠 수 있습니다. 배포 설계를 구현하는 데 필요한 개발 자원 하드웨어, 소프트웨어 및 인적 자원 등을 포함한 개발 자원이 제한된 경우 증분 배포를 제안할 수 있습니다. 각 증분 단계마다 동일한 자원이나 개발 팀을 다시 사용해야 할 수도 있습니다. 유지 보수, 관리 및 지원 시스템의 사용자를 관리, 유지 보수 및 지원하는 데 사용할 수 있는 자원을 분석합니다. 여기의 제한된 자원은 설계 결정에 영향을 미칠 수 있습니다.
소유 비용	<p>유지 보수, 관리 및 지원 외에 소유 비용에 영향을 미치는 다른 요소도 분석해야 합니다.</p> <p>예를 들어, 필요한 하드웨어 및 소프트웨어 업그레이드, 전선망 유지비, 텔레커뮤니케이션 비용 및 현금 지출에 영향을 미치는 기타 요소 등입니다.</p>
회사 표준 및 정책	<p>배포를 요청한 조직의 표준과 정책을 이해합니다.</p> <p>이러한 표준과 정책이 설계, 제품 선택 및 배포 방법의 기술적인 요소에 영향을 미칠 수 있습니다.</p>

표 2-2 비즈니스 제약 조건 분석 항목 (계속)

항목	설명
회사 변경 사항 관리	회사의 변경 사항 관리 절차가 배포 방법 및 시간표에 상당한 영향을 미칠 수 있습니다 .
투자 수익	모든 배포는 고객에게 투자에 따른 수익을 제공해야 합니다 . 일반적으로 투자 수익 분석에서는 자본 지출에 따른 재무 이익을 측정합니다 . 배포의 재무 이익을 평가할 때는 배포에 따라 달성될 목표를 해당 목표를 달성하는 다른 방법과 비교하거나 아무 것도 수행하지 않을 경우의 비용과 비교하여 세밀하게 분석합니다 .
규정 요구 사항	규정 요구 사항은 배포 특성에 따라 크게 다릅니다 .

배포에 대한 증분 방법

일반적으로 배포는 전체적이고 포괄적인 시스템으로 간주됩니다 . 그러나 종종 이 포괄적인 시스템을 세밀한 단계에 따라 증분적으로 달성하기도 합니다 .

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 .
- 제휴관계 가능성을 고려할 수 있습니다 .

증분 방법을 적용할 경우 일반적으로 최종의 포괄적인 솔루션에 이르는 이정표를 제공하는 로드맵을 설계합니다 . 또한 계획에서 나중에 구현될 단계에 대해 단기 솔루션을 고려해야 할 수 있습니다 .

어떤 방법을 사용하든 항상 변경 및 성장에 대한 여지를 허용하도록 배포를 설계해야 합니다 .

기술 요구 사항

이 장에서는 기술 요구 사항 분석 중에 발생하는 일부 프로세스와 절차를 설명합니다.

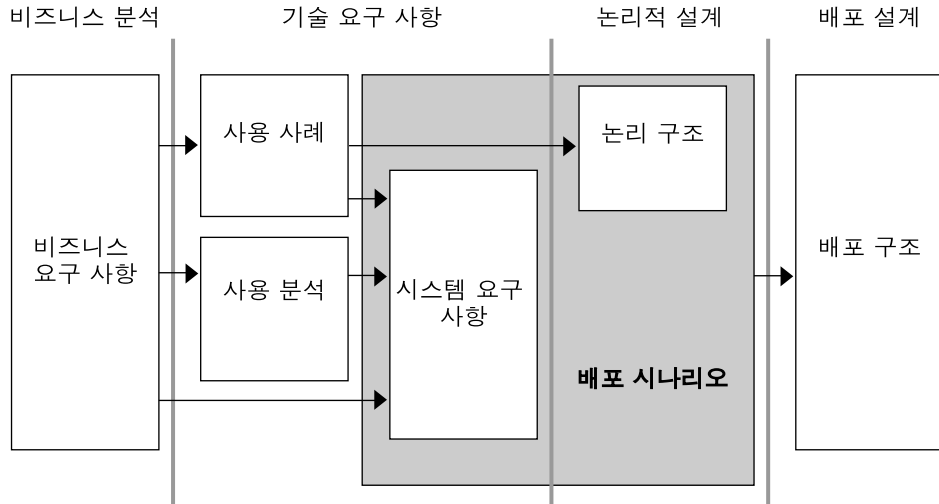
기술 요구 사항 분석은 비즈니스 분석 단계에서 작성된 비즈니스 요구 사항 문서를 기준으로 시작됩니다. 비즈니스 요구 사항을 기반으로 사용하여 다음 단계를 수행합니다.

- 배포의 예상 로드를 결정하는 데 도움을 주기 위해 *사용 분석* 수행
- 사용자와 배포 간의 일반적인 상호 작용을 모델링하는 *사용 사례* 집합 작성
- 비즈니스 요구 사항, 사용 사례 및 사용 분석에서 추출하는 *시스템 요구 사항* 집합 작성

사용 사례는 설계 단계에서 *논리 구조* 설계의 기준으로도 사용됩니다. 논리 구조와 시스템 요구 사항은 함께 *배포 시나리오*를 구성하며, 이 배포 시나리오는 나중에 배포 설계 단계의 입력으로 사용됩니다.

다음 그림에서는 기술 요구 사항 단계를 비즈니스 분석, 논리 설계 및 배포 설계 단계와 비교하여 보여 줍니다.

그림 3-1 기술 요구 사항 단계와 다른 배포 계획 단계



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

이 장의 내용은 다음과 같습니다.

- “ 사용 분석 ”
- 30 페이지의 “ 사용 사례 ”
- 31 페이지의 “ 시스템 요구 사항 ”

사용 분석

사용 분석에서는 설계하려는 배포의 여러 사용자를 식별하고 이러한 사용자의 사용 패턴 결정합니다. 수집한 정보는 예상 로드 조건에 대한 개념을 제공하며 뒤에서 성능 요구사항 및 기타 시스템 요구 사항을 결정하는 데 사용됩니다. 또한 사용 분석 정보는 30 페이지의 “ 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 항목을 나열합니다.

표 3-1 사용 분석 항목

항목	설명
사용자 수 및 유형	<p>배포에서 지원해야 할 사용자 수를 식별하고 필요하다면 해당 사용자를 범주화합니다.</p> <p>예를 들면 다음과 같습니다.</p> <ul style="list-style-type: none"> • B2C (기업 - 사용자) 배포에는 방문자가 많을 수 있지만, 비즈니스 트랜잭션에 등록 및 관여하는 사용자 수는 적습니다. • B2E (기업 - 직원) 배포에서는 일반적으로 모든 직원을 수용해야 하지만, 일부만 회사 외부 네트워크에서의 액세스가 필요할 수 있습니다. • B2E (기업 - 직원) 배포에서 관리자는 일반 직원이 액세스할 수 없는 영역에 대한 권한이 필요할 수 있습니다.
활성 및 비활성 사용자	<p>활성 및 비활성 사용자의 사용 패턴과 비용을 식별합니다.</p> <p>활성 사용자는 시스템에 로그인하여 시스템의 구성 요소와 상호 작용하는 사용자입니다. 비활성 사용자는 로그인하지 않았거나 로그인했지만 시스템의 구성 요소와 상호 작용하지 않는 사용자일 수 있습니다.</p>
관리 권한이 있는 사용자	<p>배포된 시스템을 액세스하는 배포를 모니터, 업데이트 및 지원하는 사용자를 식별합니다.</p> <p>시스템 요구 사항에 영향을 미칠 수 있는 특정 관리 사용 패턴을 결정합니다. 방화벽 외부에서 배포 관리를 예로 들 수 있습니다.</p>
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>

표 3-1 사용 분석 항목 (계속)

항목	설명
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? 또는 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>중종 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 배포와 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.

각 사용 사례에는 사용자 동작에 대한 수량 예측이 포함될 수 있으며, 나중에 성능, 가용성 및 기타 서비스 품질에 대한 시스템 요구 사항을 결정하는 데 이 수치를 사용할 수 있습니다. 또한 사용 사례는 **41 페이지의 4 장, “논리 구조 설계”**에서 설명하는 논리 구조 설계를 위한 출발점이기도 합니다.

중종 사용 사례에 상대적인 가중치를 할당하기도 하는데 가중치가 가장 높은 사용 사례는 가장 일반적인 사용자 작업을 나타냅니다. 사용 사례의 가중치는 시스템 요구 사항을 결정하는 데 도움이 됩니다.

다음 두 가지 수준으로 사용 사례를 설명할 수 있습니다.

- 사용 사례 다이어그램
사용자와 사용 사례 간의 관계를 그래픽으로 표시합니다.
- 사용 사례 보고서
기본 및 대체 이벤트 흐름을 포함하는 개별 사용 사례에 대한 설명입니다.

시스템 요구 사항

시스템 요구 사항에서는 비즈니스 분석을 통해 얻은 비즈니스 요구 사항을 충족시키기 위해 배포된 시스템에서 제공해야 하는 서비스 품질을 설명합니다. 대개 사용 분석과 사용 사례를 비즈니스 요구 사항과 같이 사용하여 시스템 요구 사항을 도출합니다.

다음 표에서는 시스템 요구 사항을 지정하기 위해 주로 사용하는 시스템 품질을 나열합니다.

표 3-2 배포 설계에 영향을 미치는 시스템 품질

시스템 품질	설명
가용성	최종 사용자가 시스템의 자원 및 서비스를 액세스할 수 있는 빈도에 대한 측정이며 대개 시스템의 <i>가동 시간</i> 으로 표시됩니다.
잠재 용량	추가 자원 없이 비정상적인 최고 로드 사용을 처리할 수 있는 시스템의 기능입니다.
성능	사용자 로드 조건에 따라 응답 시간 및 대기 시간을 측정한 값입니다.
확장성	시간에 따라 배포된 시스템에 용량 및 사용자를 추가할 수 있는 기능입니다. 확장성은 일반적으로 시스템에 자원을 추가하는 것을 포함하지만 배포 구조 변경을 요구해서는 안 됩니다.
보안	시스템과 그 사용자의 무결성을 설명하는 요소들의 복잡한 조합입니다. 보안에는 정보의 보안 전송뿐 아니라 사용자의 인증 및 권한 부여도 포함됩니다.
서비스 가능성	시스템 모니터링, 발생한 문제 복구, 하드웨어 및 소프트웨어 구성 요소 업그레이드 등을 포함하여 배포된 시스템의 관리 용이성입니다.

배포 설계에 영향을 미치는 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 가용성에 영향을 미칠 수 있습니다. 가용성 문제를 처리하기 위해 서버를 추가하면 유지 보수 비용(서비스 가능성)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 배포 설계에 영향을 미치는 시스템 품질을 자세히 조사하여 시스템 요구 사항을 공식화할 때 고려해야 할 요소에 대한 지침을 제공합니다. 서비스 수준 계약을 작성할 때 사용되는 특별한 시스템 요구 사항 집합인 서비스 수준 요구 사항에 대한 절도 있습니다.

가용성

가용성은 배포된 시스템의 *가동 시간*을 지정하는 방법입니다. 대개는 사용자가 시스템을 액세스할 수 있는 시간의 백분율로 측정합니다. 시스템을 액세스할 수 없는 시간 (*중단 시간*)은 하드웨어, 소프트웨어 또는 네트워크의 오류나 시스템을 중단시키는 기타 요소 (예: 정전)로 인한 것일 수 있습니다. 대부분의 경우 서비스 (유지 보수 및 업그레이드)에 예약된 시간은 중단 시간으로 간주되지 않습니다.

대개 사용자가 달성할 수 있는 "9"의 개수로 가용성을 측정합니다. 예를 들어, 99% 가용성은 9가 두 개입니다. 9를 추가로 지정하면 가용성에 대한 배포 설계에 상당한 영향을 미칩니다. 다음 표는 하루 24시간 365일 (8,760시간) 실행되는 시스템의 가용성에 9를 추가한 결과를 표시합니다.

표 3-3 1년 내내 (8,760시간) 실행되는 시스템의 중단 시간

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

내결합성이 있는 시스템

9가 네 개 또는 다섯 개인 가용성 요구 사항에서는 일반적으로 *내결합성*이 있는 시스템을 요구합니다. 내결합성이 있는 시스템은 하드웨어나 소프트웨어 오류 중에도 서비스를 계속할 수 있어야 합니다. 일반적으로 내결합성은 주요 서비스를 제공하는 하드웨어 (예: CPU, 메모리 및 네트워크 장치)와 소프트웨어 모두의 중복을 통해 달성됩니다.

*단일 오류 지점*은 중복 구성 요소에서 백업하지 않는 하드웨어나 소프트웨어 구성 요소입니다. 이 구성 요소의 실패는 시스템의 서비스가 손실을 일으킵니다. 내결합성이 있는 시스템을 설계할 때는 잠재적인 단일 오류 지점을 식별하여 제거해야 합니다.

내결합성이 있는 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.

Sun Cluster 3.1 4/04

Sun Cluster 3.1 4/04 소프트웨어는 고가용성의 내결합성이 있는 시스템을 요구하는 배포를 위한 고가용성 솔루션을 제공합니다. Sun Cluster 3.1 4/04에서는 서버, 저장소 및 기타 네트워크 자원을 결합하여 시스템의 사용자 서비스를 방해하지 않고 신속하게 수행되는 페일오버 프로세스를 제공합니다.

서비스 가용성 우선 순위 지정

사용자 관점에서 가용성은 전체 시스템의 가용성보다는 배포된 시스템에서 제공하는 각 서비스에 적용되는 경우가 많습니다. 예를 들어, 인스턴트 메시징 서비스를 사용할 수 없더라도 대개 다른 서비스의 가용성에 거의 또는 전혀 영향을 미치지 않습니다. 그러나 많은 다른 서비스가 종속된 서비스(예: Directory Server)의 가용성은 시스템에서 폭 넓은 영향을 미칩니다.

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

표 3-4 서비스 가용성 우선 순위 지정

우선 순위	서비스 유형	설명
1	전략적	작업에 필수적인 서비스. 예를 들어, 많은 서비스가 Directory Server 에 종속됩니다.
2	임무 결정적	최고 로드 시 사용할 수 있어야 하는 서비스. 임무 결정적으로 정의된 응용 프로그램의 데이터베이스 서비스를 예로 들 수 있습니다.
3	사용 가능해야 함	사용 가능해야 하지만 성능은 떨어져도 관계 없는 서비스. 예를 들어, Messaging Server 가용성은 일부 비즈니스 환경에서는 중요하지 않을 수도 있습니다.
4	연기할 수 있음	지정한 기간 내에 사용 가능해야 하는 서비스. 예를 들어, Instant Messaging 가용성은 일부 비즈니스 환경에서 필수적이지 않을 수도 있습니다.
5	선택 사항	무기한 연기할 수 있는 서비스.

가용성 요구 사항을 구현하는 여러 설계 전략에 대한 자세한 내용은 60 페이지의 “가용성을 위한 크기 조정”을 참조하십시오.

잠재 용량

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

성능

성능 요구 사항 결정은 성능에 대한 예상 비즈니스 요구 사항을 시스템 요구 사항으로 변환하는 과정입니다. 일반적으로 비즈니스 요구 사항에서는 응답 시간을 지정하는 비기술적 용어로 성능을 표현합니다. 예를 들어, 웹 기반 액세스에 대한 비즈니스 요구 사항에서 다음과 같이 기술할 수 있습니다.

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

이 비즈니스 요구 사항을 출발점으로 하여 모든 사용 사례를 조사하고 이 요구 사항을 시스템 수준으로 표현하는 방법을 결정합니다. 사용 분석에서 결정된 사용자 로드 조건을 고려합니다. 각 사용 사례의 성능 요구 사항을 *지정한 로드 조건에 따른 응답 시간*이나 *응답 시간 및 처리 능력*으로 표현합니다. 허용 가능한 오류 수를 지정할 수도 있습니다.

다음은 성능에 대한 시스템 요구 사항을 지정하는 방법의 예입니다.

사용자 로그인에 대한 응답은 하루종일 4 초보다 길어서는 안 되며 15 분 간격으로 측정되고 오류가 백만 개의 트랜잭션 당 3.4 개보다 적어야 합니다.

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

확장성

확장성은 시간에 따라 시스템에 용량과 사용자를 추가할 수 있는 기능을 설명합니다. 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 시스템 요구 사항에서 확장성 요구 사항을 지정할 필요는 없습니다. 배포 설계 단계의 배포 구조에서는 확장성 요구 사항을 지정하지 않은 경우에도 시스템 확장을 고려해야 합니다.

확장성 요구 사항 결정은 그리 과학적이지 않습니다. 시스템 증가를 예측하는 데에는 달성할 수 없는 예상, 예측 및 추측 등이 포함됩니다. 확장 가능한 시스템을 구축하기 위한 세 가지 비결은 다음과 같습니다.

- 고성능 설계 전략을 채택합니다.
 성능 요구 사항의 지정 및 설계 동안 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.
- 배포를 단계적으로 구현합니다.
 증분 구현은 자원 추가 예약에 도움이 될 수 있습니다.
- 광범위한 성능 모니터링을 구현합니다.
 배포의 성능을 모니터링하면 배포에 자원을 추가할 시기를 결정하는 데 도움이 될 수 있습니다.

다음 표에서는 확장성에 대해 고려할 일부 항목을 나열합니다.

표 3-5 확장성 고려 사항

항목	설명
사용 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 로드에 대한 성능 평가를 기준으로 24 개월에 대해 예측하는 양입니다. 예측 기간은 예상의 신뢰성에 따라 상당히 다릅니다.

표 3-5 확장성 고려 사항 (계속)

항목	설명
적절한 중요 시점 설정	<p>예상치 못한 증가를 수용하는 버퍼가 포함된 단기간 요구 사항을 충족시키기 위한 증분 배포 설계를 구현합니다. 시스템 자원을 추가할 중요 시점을 설정합니다.</p> <p>예를 들면 다음과 같습니다.</p> <ul style="list-style-type: none"> • 자본 취득 예 : 분기별 또는 년별 • 하드웨어 선행 시간 예 : 1-6 주 • 버퍼 (증가 예상에 따라 10% - 100%)
최신 기술 통합	<p>최신 기술 (예 : 더 빠른 CPU 와 웹 서버) 과 이 기술이 기본 구조의 성능에 미칠 수 있는 영향을 이해합니다.</p>

보안 요구 사항

보안은 사용자 트랜잭션 및 관련 데이터의 무결성을 포함하여 시스템과 해당 사용자의 무결성에 영향을 미치는 시스템 품질입니다. 다른 시스템 요구 사항처럼 보안 요구 사항 분석은 비즈니스 요구 사항 , 사용량 분석 및 사용 사례에 의해 추진됩니다.

보안 요구 사항 분석은 다음 범주로 나뉩니다.

- 인증
- 권한 부여
- 아이디 관리

인증 , 권한 부여 및 아이디 관리는 견고한 보안 방법의 적용을 위한 기업 차원의 정책 과 함께 트랜잭션을 보안하고 사이트에 저장된 데이터가 손상되지 않도록 할 수 있습니다.

주 인프라 (예 : 방화벽 소프트웨어 및 네트워크 설계) 의 무결성에 영향을 미치는 보안 요구 사항은 일반적으로 시스템 요구 사항 분석에서 고려되지 않습니다. 대신 이러한 보안 문제는 배포 설계 중에 영향을 미치기 시작합니다.

인증

인증은 사용자가 자신을 시스템에 식별시키는 방법이기도 하고 시스템이 자신을 사용자에게 식별시키는 방법이기도 합니다. 인증은 무단 액세스로부터 시스템을 보호하는 시스템 무결성의 핵심 부분입니다.

배포를 위한 최상의 인증 방법을 선택하려면 사용자 요구 사항에 대해 알고 있어야 합니다. 예를 들어, B2C (기업 - 사용자) 배포의 경우 사용자가 사용자 이름 / 비밀번호 조합을 사용하여 등록할 수 있도록 할 수 있습니다. 이 사용자는 신뢰할 수 있는 인증 기관 (예: VeriSign) 에서 발행한 서버 인증서를 사용하여 보안 전송을 통해 판매 시스템을 인증합니다.

B2E (기업 - 직원) 배포에서는 대신 직원에게 기존 사용자 기반에서 제공할 수 있습니다. 회사 방화벽 내에서는 알려진 보안 위치에 대한 액세스가 허용됩니다. 방화벽 외부에서는 보안 위치에 대한 액세스가 회사 방화벽 내에서 인증 및 리디렉션을 수행하는 프록시를 통해 이루어집니다.

권한 부여

권한 부여는 인증된 사용자에 대해 특정 권한을 인정하는 것입니다. 예를 들어, 관리자 권한을 가진 사용자는 배포된 시스템에서 일반 사용자가 액세스할 수 없는 부분을 액세스할 수 있습니다.

권한 부여는 단일 사인 온 (SSO) 를 구현하는 배포에서도 사용됩니다. 배포에 대해 인증된 사용자는 두 번 이상 로그인하지 않고도 여러 서비스에 대한 액세스 권한을 가질 수 있습니다.

아이디 관리

배포된 시스템에서는 시스템 서비스를 액세스할 사용자를 추가, 수정 또는 삭제할 수 있어야 합니다. 사용자 요구 사항에 따라 아이디 관리는 권한이 있는 관리자가 수행하거나 *위임된 관리* 인터페이스를 통해 사용자 자신이 수행할 수 있습니다. 중소기업이나 대기업의 배포에서는 위임된 관리 설계를 고려해야 합니다. 위임된 관리는 고객 만족도를 개선하고 시스템 관리 비용을 줄입니다.

서비스 가능성 요구 사항

서비스 가능성은 시스템 모니터링, 발생한 문제 복구, 하드웨어 및 소프트웨어 구성 요소 업그레이드 등을 포함하여 배포된 시스템을 얼마나 쉽게 관리할 수 있는지를 나타냅니다.

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>배포의 사용 패턴을 식별하여 유지 보수의 기회 시간을 결정합니다 .</p> <p>예를 들어 , 대개 일반적으로 업무 시간 중에 사용량이 최고인 시스템의 경우 기회 시간은 밤이나 주말이 됩니다 . 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 위한 시간이 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>이렇게 하면 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 요구 사항*은 고객 지원을 제공해야 하는 조건을 지정하는 시스템 요구 사항 집합입니다. 서비스 수준 요구 사항은 대개 프로젝트 승인 중에 서명하는 서비스 수준 계약의 기준이 됩니다.

시스템 요구 사항처럼 서비스 수준 요구 사항은 비즈니스 요구 사항에서 추출되며 배포에서 충족시켜야 할 전체적인 시스템 품질에 대해 고객에게 하는 일종의 보증을 나타냅니다. 서비스 수준 계약은 사용자와 고객 간의 계약이기 때문에 서비스 수준 요구 사항을 모호하게 지정해서는 안 됩니다. 서비스 수준 요구 사항에서는 요구 사항을 검사하는 기준과 요구 사항을 충족시키는데 실패한 것을 간주되는 경우 정확하게 정의해야 합니다.

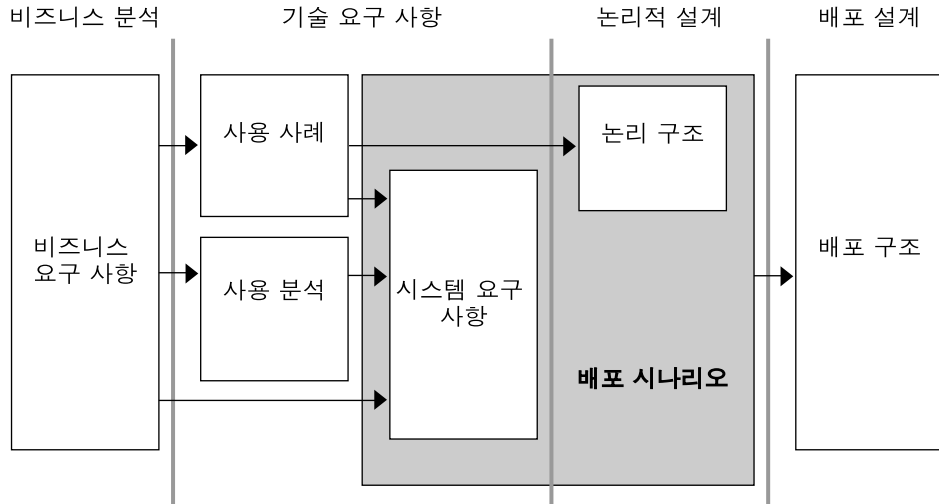
논리 구조 설계

이 장에서는 논리 구조 작성 과정을 설명하고 중소기업의 통신 배포에 있는 사용 사례의 전형인 사용 사례 세트를 사용하여 과정 예를 제공합니다.

논리 구조는 배포의 비즈니스 목표를 충족시키는 데 필요한 소프트웨어 서비스를 제공하는 Java Enterprise System 구성 요소 및 그 종속성을 식별합니다. 일반적으로 기술 요구 사항 단계에서 개발된 사용 사례에서는 필요한 소프트웨어 서비스를 표시합니다. 그러나 소프트웨어 서비스 정보는 주로 비즈니스 분석 단계에서 파생된 비즈니스 요구 사항에서 직접 가져올 수 있습니다.

논리 구조는 요구 사항 분석에서 결정된 시스템 요구 사항과 함께 배포 시나리오를 나타냅니다. 배포 시나리오는 배포 구조를 설계하기 위한 기준입니다. 다음 그림에서는 논리 설계 단계와 비즈니스 분석, 기술 요구 사항 및 배포 설계 단계의 관계를 보여 줍니다.

그림 4-1 논리 설계와 다른 배포 계획 단계와의 관계



이 장의 내용은 다음과 같습니다.

- “ 배포 계획 예 ”
- 43 페이지의 “Java Enterprise System 서비스 ”
- 47 페이지의 “ 예제 배포의 논리 구조 ”

배포 계획 예

배포 계획 과정을 설명하기 위해 이 절에서는 전형적인 중소기업의 통신 요구를 기준으로 한 예제 배포의 사용 사례를 소개합니다. 이 예제 배포는 여러 배포 계획 단계를 설명하기 위해 이 백서의 나중 장에서 계속됩니다.

주의 예제 배포의 사용 사례, 논리 구조, 배포 구조 및 설계 지정은 배포 계획 과정의 단계를 단순화한 것입니다.

설명을 위해 예를 단순화했습니다. 예의 설계는 불완전하고 빌드하거나 테스트하지 않았습니다. 예를 계획 중인 배포의 청사진으로 사용하지 마십시오.

예제 배포에서는 통신 배포의 일반적인 비즈니스 요구 사항에서 파생된 사용 사례 세트를 출발점으로 삼습니다. 다음 표에서 이 사용 사례를 요약합니다.

표 4-1 예제 배포의 사용 사례

사용 사례	설명
#1 단일 사인 온	<p>웹 브라우저에서 시스템에 로그인하여 (사용자 이름 / 비밀번호) 다음 엔터프라이즈 서비스에 액세스합니다.</p> <ul style="list-style-type: none"> • 사용자 정의 포털 웹 페이지 • 웹 기반 전자 메일 페이지 • 달력 인터페이스 • 보안 웹 페이지
#2 개인 포털 화면 열기	<p>웹 브라우저에서 개인 포털 화면으로 이동합니다.</p>
#3 포털을 통해 전자 메일 확인	<p>포털 인터페이스에서 새로운 전자 메일 메시지를 확인합니다.</p>
#4 포털을 통해 보안 웹 페이지 확인	<p>개인 포털 인터페이스에서 보안 프로젝트 상태 페이지를 확인합니다.</p>
#5 포털을 통해 달력 확인	<p>포털 인터페이스에서 매일 약속을 확인합니다.</p>
#6 달력 관리	<p>웹 기반 달력 클라이언트에서 약속을 예약합니다.</p>
#7 전자 메일 관리	<p>전자 메일 클라이언트에서 전자 메일을 읽고 보냅니다.</p>

이 사용 사례에서 다음 절에서 설명하는 대로 논리 구조에 필요한 서비스를 추출할 수 있습니다.

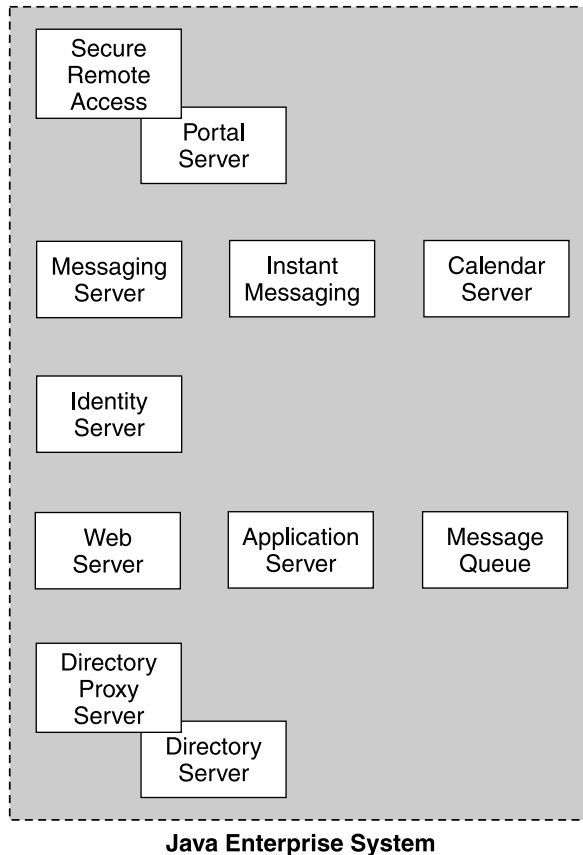
Java Enterprise System 서비스

논리 구조의 설계에서는 우선 배포에 필요한 서비스를 결정할 수 있게 도와주는 사용 사례를 분석합니다. Java Enterprise System 에 대한 지식과 이전 설계 경험을 이용하여 사용 사례에서 식별한 서비스를 제공하는 Java Enterprise System 구성 요소에 대한 초기 논리 설계의 레이아웃을 만듭니다.

구성 요소를 배치할 때 시스템 내에서 데이터의 논리적 흐름과 서비스를 제공하는 구성 요소 간 종속성을 고려합니다. 설계의 구성 요소 간 데이터 흐름에 영향을 미치는 이러한 종속성을 논리 설계에서 반영해야 합니다.

다음 그림에서는 Java Enterprise System 에서 제공하는 구성 요소를 보여 줍니다. Java Enterprise System 구성 요소의 상호 종속성에 대해 이해하려면 이 그림과 [표 4-2 페이지 45](#) 를 참조하십시오. 일반적으로 그림에서 아래에 있는 구성 요소가 그 위의 구성 요소에 대한 지원을 제공합니다.

그림 4-2 Java Enterprise System 구성 요소



다음 표에서는 Java Enterprise System 구성 요소 간 실제 상호 종속성을 나열합니다.

표 4-2 Java Enterprise System 구성 요소 상호 종속성

Java Enterprise System 구성 요소	지원 대상	종속 대상
Application Server	Identity Server Portal Server	Message Queue
Calendar Server	Portal Server (달력 채널용)	Directory Server Identity Server(단일 사인 운용) Messaging Server (Calendar Server 전자 메일 알림 서비스용)
Directory Proxy Server	없음	Directory Server
Directory Server	Administration Server Calendar Server Directory Proxy Server Identity Server Instant Messaging Messaging Server Portal Server	없음
Identity Server	Portal Server 단일 사인 온을 위해 구성하는 경우 : Calendar Server Instant Messaging Messaging Server	Directory Server Application Server 또는 Web Server
Instant Messaging	Portal Server	Directory Server
Message Queue	Application Server	Directory Server (선택 사항)
Messaging Server	Calendar Server Portal Server (메시징 채널용)	Directory Server Web Server Identity Server
Portal Server	Secure Remote Access	Directory Server Application Server 또는 Web Server Portal Server 채널을 사용하도록 구성된 경우 : Calendar Server Messaging Server Instant Messaging
Secure Remote Access	없음	Portal Server
Web Server	Identity Server Portal Server	없음

예를 들어, 예제 통신 배포의 Java Enterprise System 구성 요소를 배치하려면 표 4-1 페이지 43에 나열된 사용 사례를 분석합니다. 다음 표에서는 사용 사례에서 표시한 대로 배포에 직접 필요한 구성 요소를 나열합니다.

표 4-3 예제 사용 사례를 지원하기 위한 Java Enterprise System 구성 요소

Java Enterprise System 구성 요소	사용 사례
Portal Server	#1 단일 사인 온 #2 개인 포털 화면 열기 #3 포털을 통해 전자 메일 확인 #4 포털을 통해 보안 웹 페이지 확인 #5 포털을 통해 달력 확인
Calendar Server	#1 단일 사인 온 #5 포털을 통해 달력 확인 #6 달력 관리
Messaging Server	#1 단일 사인 온 #3 포털을 통해 전자 메일 확인 #7 전자 메일 관리

또한 위 표 4-3에 나열된 구성 요소를 지원하는 데 필요한 Java Enterprise System 구성 요소도 결정해야 합니다. 다음 표에서 이러한 추가 구성 요소를 나열합니다.

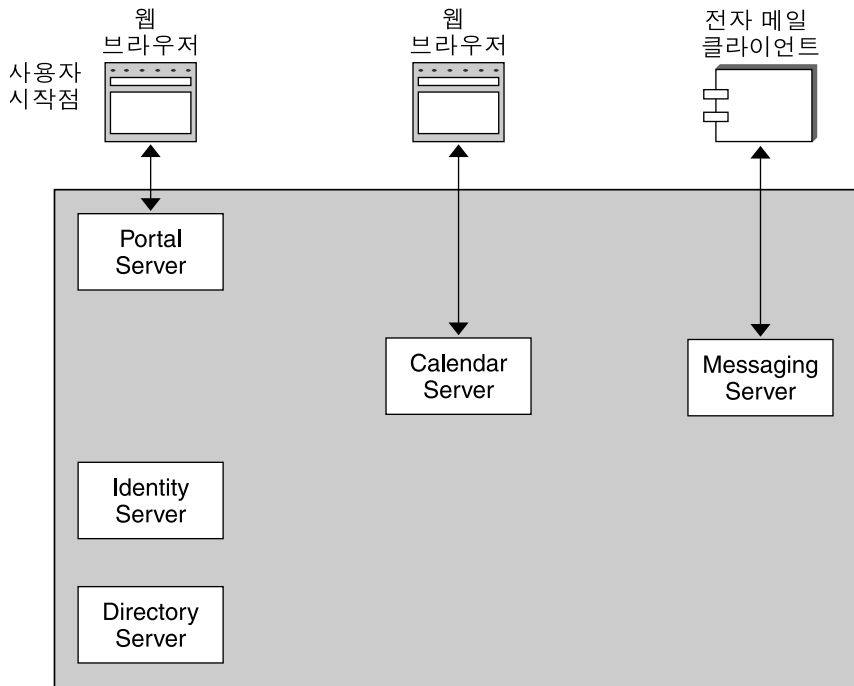
표 4-4 예제 사용 사례를 지원하기 위한 추가 구성 요소

Java Enterprise System 구성 요소	제공되는 지원
Identity Server	Portal Server 를 지원합니다. Calendar Server 및 Messaging Server 에 대해 단일 사인 온을 지원합니다.
Directory Server	Identity Server 및 Portal Server 를 지원합니다.
Application Server 또는 Web Server	Identity Server 및 Portal Server 를 지원합니다. (Identity Server 및 Portal Server 는 웹 컨테이너 안에서 실행해야 합니다.)

예제 배포의 논리 구조

아래 **그림 4-3**에서는 배포에 대한 사용자 시작점을 나타내는 예제 배포의 구성 요소 레이아웃을 보여 줍니다. 대부분의 지원이 필요한 서비스 (Portal Server)를 맨 위에 놓고 그 아래에 지원하는 구성 요소를 나열하여 **표 4-2 페이지 45**에서 설명한 구성 요소 간 종속성을 대략적으로 반영합니다. 이 그림에서는 Portal Server 및 Identity Server를 지원하기 위해 웹 컨테이너를 제공하는 구성 요소는 해당 종속성이 배포의 데이터 흐름을 반영하지 않기 때문에 표시하지 않습니다.

그림 4-3 논리 구조의 Java Enterprise System 구성 요소



예제 배포의 논리 구조

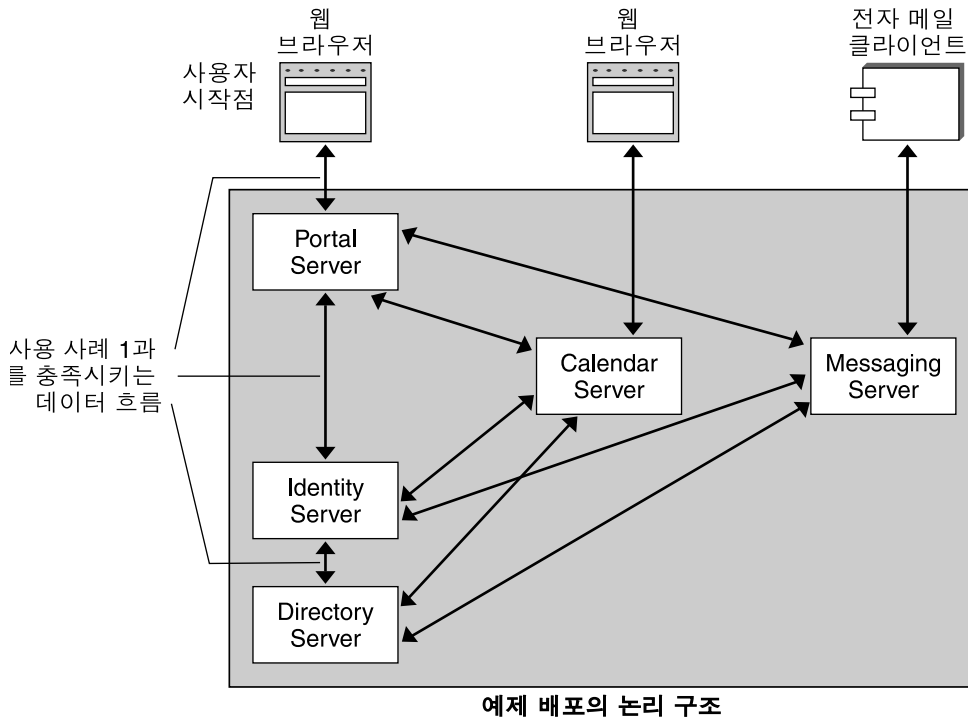
- Portal Server 및 Identity Server를 지원하는 웹 컨테이너가 표시되지 않습니다.

예제 배포의 데이터 흐름

사용 사례를 검토하여 논리 구조의 서비스 간 데이터의 논리적 흐름을 결정하고, 레이아웃에 이 흐름을 표시합니다. 시스템의 서비스 간 데이터 흐름은 [52 페이지의 “계획된 배포의 크기 조정”](#)에서 설명한 대로 성능 및 가용성을 위한 크기를 조정할 때 중요한 역할을 수행합니다.

다음 그림은 예제 배포의 데이터 흐름을 표시합니다. 데이터 흐름은 배포의 사용 사례 및 Java Enterprise System 서비스 종속성으로 결정됩니다.

그림 4-4 예제 배포에 대한 데이터의 논리적 흐름



[그림 4-4](#)에서는 사용 사례 1과 2를 모두 충족시키는 데이터 흐름을 호출합니다. 이 데이터 흐름은 다음을 나타냅니다.

- 웹 기반 클라이언트의 사용자 로그인 요청
- 인증 서비스를 제공하기 위한 Identity Server의 Portal Server에 대한 종속성

- Directory Server 에서 Identity Server 에 제공한 LDAP 정보

그림 4-4 의 나머지 데이터 흐름도 마찬가지로 사용 사례와 서버 종속성에서 추출됩니다.

배포 시나리오

완료된 논리 구조 설계와 요구 사항 분석에서 추출된 시스템 요구 사항이 배포 시나리오를 구성합니다. 배포 시나리오는 5 장, “배포 구조 설계”에서 설명하는 배포 구조 설계의 출발점입니다.

배포 시나리오

배포 구조 설계

이 장에서는 성능, 보안, 가용성 및 기타 시스템 품질에 대해 배포를 설계하는 방법에 대해 설명합니다. 또한 배포 설계를 최적화하는 방법도 설명합니다.

배포 구조는 논리적 구조를 물리적 환경에 매핑하는 것을 설명합니다. 물리적 환경에는 인트라넷이나 인터넷 환경의 컴퓨팅 노드, CPU, 메모리, 저장 장치 및 기타 하드웨어 및 네트워크 장치가 포함됩니다.

배포 구조 설계에서는 기술 요구 사항 단계에서 지정한 시스템 요구 사항을 충족시키는데 필요한 물리적 자원을 결정하기 위해 *배포 크기*를 조정합니다. 또한 배포 크기 조정 결과의 분석을 통해 *자원을 최적화*하여 비즈니스 제약 조건 내에서 자원을 가장 잘 이용하는 설계를 작성합니다.

배포 구조 설계를 완료한 후 *프로젝트 승인* 동안에 배포의 실제 비용을 평가합니다. 프로젝트가 승인되면 배포 완료를 위한 계약에 서명하고 프로젝트를 구현하기 위한 자원을 구입합니다.

*세부 설계 지정*은 프로젝트 승인 이후에 수행됩니다. 세부 설계 지정은 구현 단계에서 설계를 작성하는 데 사용됩니다.

이 장에서는 4 장의 예제 배포를 계속 사용하여 배포 구조 설계 과정의 여러 단계를 설명합니다.

이 장의 내용은 다음과 같습니다.

- 52 페이지의 “ 계획된 배포의 크기 조정 ”
- 67 페이지의 “ 자원 최적화 ”
- 69 페이지의 “ 예제 배포 구조 ”
- 71 페이지의 “ 세부 설계 지정 ”

계획된 배포의 크기 조정

계획된 배포의 크기 조정은 시스템 요구 사항을 이행하고 궁극적으로 비즈니스 목표를 충족시키는 데 필요한 하드웨어 자원 집합을 결정하는 과정입니다. 배포의 계획 및 설계 등의 다른 요소와 마찬가지로 크기 조정은 그리 과학적이지 않으며 공식이나 방법으로 규정할 수 없습니다. 성공적인 크기 조정은 과거의 설계 경험, 시스템 구조에 대한 지식, 도메인 지식 및 응용된 창조적 생각 등을 조합한 결과입니다.

크기 조정은 31 페이지의 “시스템 요구 사항”에서 설명한 대로 다음과 같은 서비스 품질에 대해 기존에 결정한 시스템 요구 사항을 기준으로 수행됩니다. 배포 설계 이전 단계의 비즈니스 요구 사항, 사용 분석, 사용 사례도 시스템 크기 조정에서 중요한 역할을 수행합니다.

크기 조정 연습을 수행할 때 사용 사례와 사용 분석을 통해 사용 사례를 지원하는 데 필요한 자원을 결정할 수 있습니다. 일반적으로 가중치가 가장 높은 사용 사례 (가장 일반적인 트랜잭션을 나타냄) 부터 시작하여 가중치가 가장 낮은 사용 사례로 진행합니다. 이와 같이 가중치가 적용된 사용 사례를 사용하면 시스템에 예상되는 스트레스에 따라 자원을 할당할 수 있습니다.

다음 절에서는 다음과 같은 시스템 품질에 대해 배포의 크기를 조정하는 방법에 대한 일반적인 지침을 제공합니다.

- 성능
- 보안
- 가용성
- 서비스 가능성

성능을 위한 크기 조정

성능 및 로드 요구 사항을 위한 크기 조정은 배포된 시스템에서 서비스를 지원하는 데 필요한 CPU 수와 해당하는 메모리를 예상하는 반복적인 과정입니다. 서비스 지원에 필요한 CPU 수를 예상할 때 다음을 고려하십시오.

- 서비스에 적용하는 사용 사례 및 해당 사용 분석
- 기술 요구 사항 분석에서 결정된 시스템 요구 사항
- 서비스를 제공하는 Java Enterprise System 구성 요소에 대한 과거의 경험
- 다양한 배포 시나리오 유형을 크기 조정 경험에 있는 Sun 전문 서비스의 자문

성능을 위한 크기 조정 과정은 대개 다음 단계로 구성됩니다. 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.
2. 구성 요소 간의 종속성을 고려하여 CPU 예상 개수를 조정합니다.
3. 보안, 가용성, 확장성 및 잠재 용량 요구 사항을 반영하여 CPU 예상 개수를 조정합니다.

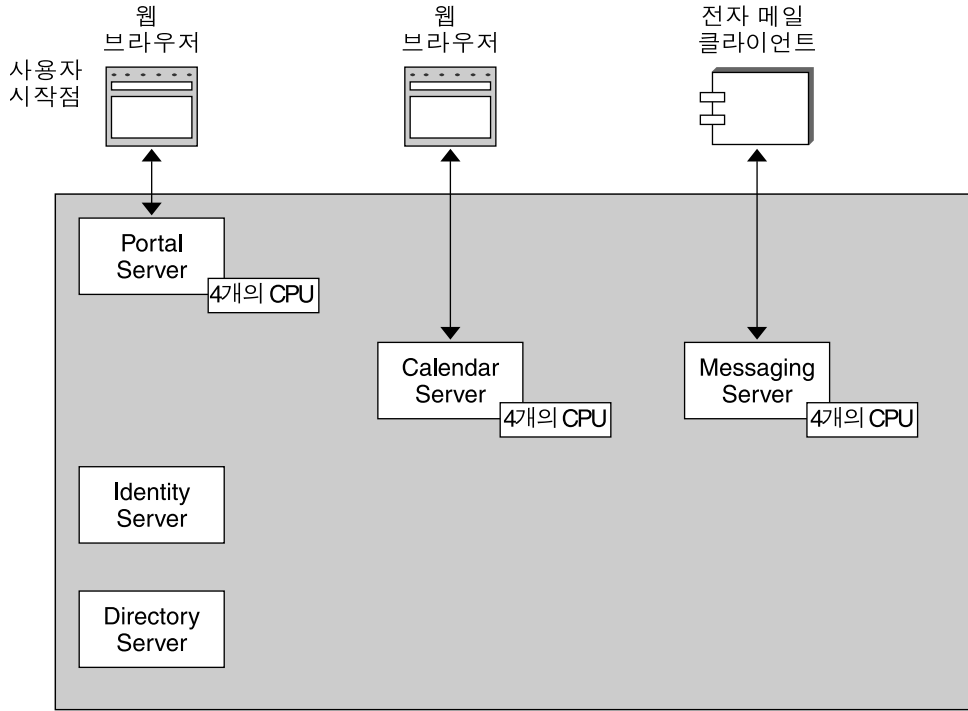
사용자 시작점에 대한 기본 CPU 예상치 결정

먼저 사용자 시작점인 각 구성 요소에서 예상되는 로드를 처리하는 데 필요한 CPU 개수를 예상합니다. 논리적 구조에 대한 레이아웃 설계에 이 예상 개수를 기록합니다.

다음 그림에서는 [42페이지의 “배포 계획 예”](#)에서 소개한 예제 배포를 사용하여 사용자 시작점인 구성 요소에 대한 초기 CPU 예상 개수를 보여 줍니다. 이 예상 개수는 시스템 요구 사항, 사용 사례 및 사용 분석에서 얻을 수 있는 수치를 나타냅니다.

주의 이 백서에서는 성능 크기 조정에 대해 자세히 설명하지 않습니다. 이 설명서에서 사용한 CPU 및 메모리 수치는 설명을 위한 임의적인 예상치입니다. 이 수치는 시스템을 설계할 때 사용할 수 있는 과정을 보여주는 것 이외에 특정 구현 정보를 나타내지는 않습니다.

그림 5-1 사용자 시작점을 제공하는 구성 요소에 대한 기본 CPU 예상 개수



예제 배포의 논리 구조

서비스 종속성에 대해 CPU 예상 개수 조정

사용자 시작점을 제공하는 구성 요소는 다른 Java Enterprise System 서비스에서 지원해야 합니다. 성능 요구 사항을 계속 지정하려면 다른 구성 요소에서 필요로 하는 지원을 고려하여 성능 예상치를 조정합니다.

예에서는 [그림 4-4 페이지 48](#)에서 설명한 대로 데이터의 논리적 흐름을 검토하고 다른 구성 요소에 지원을 제공하는 구성 요소를 조정합니다. 다음 표에서는 CPU 예상 개수에 대한 조정을 요약합니다. 예상에서 CPU 개수를 분수로 지정할 수 있습니다. 성능 예측이 완료되면 CPU 개수가 합산되어 반올림됩니다.

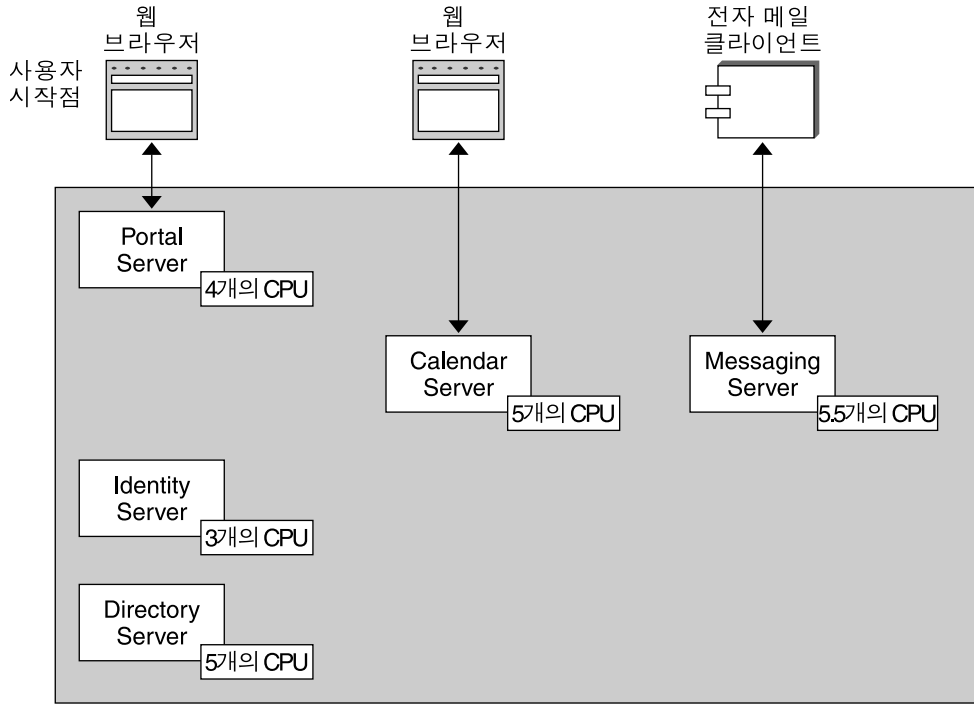
앞 절의 예상치와 같이 다음 표의 성능 예상치는 설명하기 위한 임의의 값입니다.

표 5-1 지원 서비스에 대한 CPU 예상 개수

서비스	예상 개수	설명
Portal Server	없음	다른 서비스를 지원하지 않습니다 .
Calendar Server	1 개의 CPU	다음을 지원합니다 . <ul style="list-style-type: none"> • Portal Server 의 달력 채널
Messaging Server	1.5 개의 CPU	다음을 지원합니다 . <ul style="list-style-type: none"> • Portal Server 의 메시징 채널 • Calendar Server 의 전자 메일 알림 서비스
Identity Server	3 개의 CPU	다음을 지원합니다 . <ul style="list-style-type: none"> • Portal Server • Calendar Server • Messaging Server
Directory Server	5 개의 CPU	다음을 지원합니다 . <ul style="list-style-type: none"> • Identity Server • Calendar Server • Messaging Server

다음 그림에서는 표 5-1 의 정보를 기준으로 성능에 대한 예상 개수를 업데이트합니다 .

그림 5-2 지원 서비스에 대해 조정된 CPU 예상 개수



예제 배포의 논리 구조

잠재 용량 , 확장성 및 가용성에 대한 CPU 예상 개수 조정

성능에 대한 예상 개수의 크기 조정을 완료하면 CPU의 수치를 반올림합니다. 일반적으로 CPU 개수를 다음 짝수로 반올림합니다. CPU 예상 개수를 반올림할 때 다음 요소를 고려합니다.

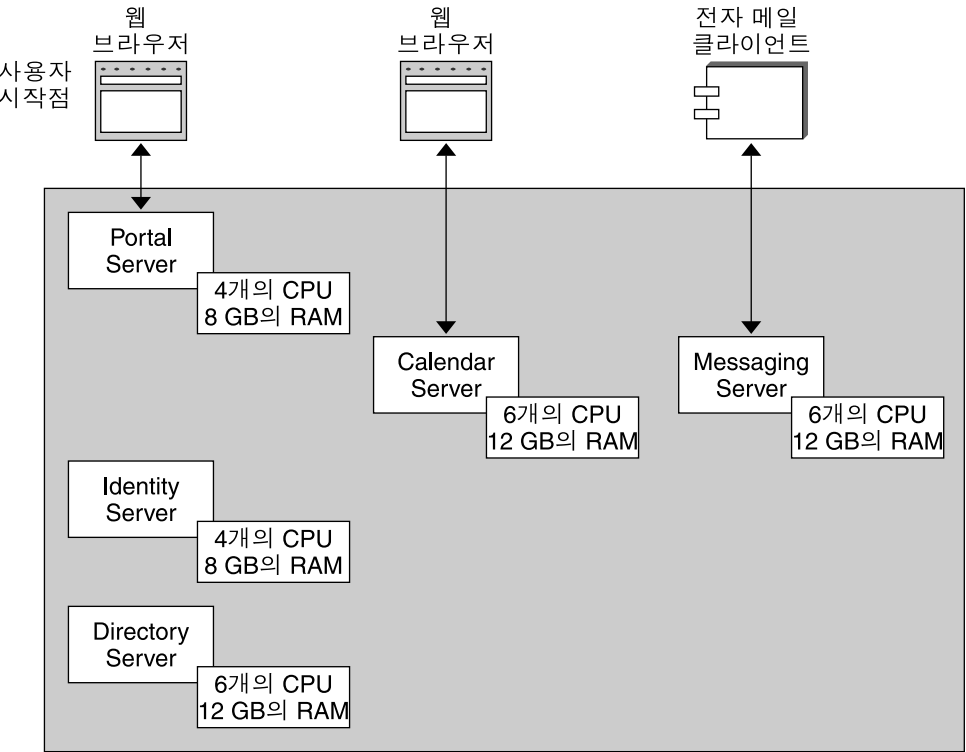
- 잠재 용량
최고 로드를 처리할 수 있는 기능을 향상시키려면 CPU 예상 개수를 늘립니다.
- 확장성
배포를 도중에 확장하지 않아도 되게 하려면 CPU 예상 개수를 늘립니다. 잠재 용량이 중요 시점에 도달할 수 있게 하려면 예상되는 확장 시점과 시간에 따라 예상되는 로드 증가를 확인합니다.

- 가용성

가용성이나 페일오버에 필요할 수 있는 서비스 복제를 고려하여 CPU 예상 개수를 조정합니다.

다음 수치는 예제 배포의 CPU 예상 개수를 조정한 것입니다. 이 수치는 각 CPU의 메모리 요구 사항도 지정합니다. 이 예에서는 각 CPU에 2GB 메모리가 필요한 것으로 가정합니다. 예의 이 메모리 사양은 설명을 위한 임의의 숫자입니다. 각 CPU에 필요한 메모리 계산은 이 백서 범위를 벗어납니다.

그림 5-3 메모리 요구 사항을 포함한 성능 수치



예제 배포의 논리 구조

보안을 위한 크기 조정

배포의 크기를 조정할 때 보안 문제는 다음과 같은 요소로 영향을 미칩니다.

- 데이터의 보안 전송
- 사용자 인증

데이터의 보안 전송에는 SSL(Secure Sockets Layer) 과 같은 보안 전송 프로토콜을 통한 트랜잭션 처리가 포함됩니다. 사용자 인증에서도 보안 전송을 통해 트랜잭션을 처리해야 합니다.

보안 전송을 통해 처리된 트랜잭션은 일반적으로 첫째 보안 세션 (*핸드셰이크*라고 함) 을 설정하고 , 둘째 전송된 데이터를 암호화 및 해독하기 위한 추가 컴퓨팅 성능을 필요로 합니다. 사용된 암호화 알고리즘 (예 : 40 비트 또는 128 비트 암호화 알고리즘) 에 따라 추가 컴퓨팅 성능이 상당히 많이 필요할 수 있습니다.

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

보안 트랜잭션을 처리하기 위한 성능 요구 사항을 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. 대부분의 경우 보안 전송이 필요한 트랜잭션의 5 내지 10 퍼센트의 예상치가 적절합니다.

예를 들어 , 온라인 전자 상거래 사이트의 제품 카탈로그를 찾을 때 고객이 선택을 마치고 " 지불 " 할 준비가 될 때까지 모든 트랜잭션이 비보안일 수 있습니다. 또한 이러한 전자 상거래 사이트에서는 대부분 보안 트랜잭션에 대한 잠재적인 응답 요구 사항을 완화합니다. 그러나 은행이나 중개업의 배포 같은 일부 사용 시나리오에서는 모든 또는 대부분의 트랜잭션이 보안되어야 하고 보안 및 비보안 트랜잭션 모두에 동일한 성능 표준을 적용합니다.

보안 트랜잭션에 대한 성능 계산

이 절에서는 예제 배포를 계속 사용하여 보안 및 비보안 트랜잭션을 모두 포함하는 사용 사례의 CPU 요구 사항을 계산하기 위한 워크시트를 설명합니다.

CPU 요구 사항을 계산하려면 워크시트에 다음 계산을 만듭니다.

1. 앞 절 52 페이지의 “ 성능을 위한 크기 조정 ”에서 계산한 CPU 요구 사항에 대한 기본 수치로 시작합니다 .
2. SSL 이 필요한 트랜잭션의 백분율을 계산하고 , SSL 트랜잭션의 CPU 요구 사항을 계산합니다 .
3. 비보안 트랜잭션에 대해 CPU 계산을 조정합니다 .
4. 보안 및 비보안 요구 사항을 합산하여 총 CPU 요구 사항을 계산합니다 .

그림 5-4 의 워크시트는 Portal Server 에 대한 추가 사용 사례와 사용 분석을 기반으로 합니다 . 추가 사용 사례와 사용 분석에서는 다음을 가정합니다 .

- 모든 로그인에는 보안 인증이 필요합니다 .
- 모든 로그인이 전체 Portal Server 로드의 10% 를 형성합니다 .
- 보안 트랜잭션의 성능 요구 사항과 비보안 트랜잭션의 성능 요구 사항이 동일합니다 .

이 예의 목적인 SSL 트랜잭션을 처리하기 위한 추가 컴퓨팅 성능을 고려하기 위해 이러한 트랜잭션을 처리하는 CPU 개수를 5 배씩 늘립니다 . 예의 다른 CPU 수치처럼 이 수치는 설명을 위한 임의의 숫자입니다 .

그림 5-4 보안 트랜잭션의 CPU 예상 개수를 계산하기 위한 워크시트

모든 Portal Server 트랜잭션에 대한 기본 예상치: 4개의 CPU

<p>1. SSL 트랜잭션에 대한 CPU 예상 개수 계산합니다.</p> <p>SSL에 10% 곱하기 $.10 \times 4 = 4$</p> <p>5x개의 CPU에 SSL 트랜잭션 곱하기 $5 \times .4 = 2\text{개의 CPU}$</p>	<p>2개의 CPU</p>
<p>2. 비SSL 트랜잭션에 대한 CPU 예상 개수를 조정합니다.</p> <p>90%가 비보안인 경우 $.9 \times 4 = 3.6\text{개의 CPU가 필요함}$</p>	<p>3.6개의 CPU</p>
<p>3. 총 CPU 예상 개수입니다.</p> <p>SSL 2 비SSL <u>3.6</u> 5.6개의 CPU</p>	<p>5.6개의 CPU</p>

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 보안 세션을 설정하는 SSL 계산의 일부 (일반적으로 "핸드셰이크" 작업)에만 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 지원하는지 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.

가용성을 위한 크기 조정

성능을 위한 크기 조정을 완료한 후 가용성을 위해 시스템 크기를 조정할 수 있습니다. 여기서 논리 구조의 구성 요소를 호스트하는 특정 서버를 지정하고 다양한 Java Enterprise System 구성 요소에 대한 로드 균형 조정, 중복 및 페일오버 전략을 설계합니다.

사용 사례 및 사용 분석을 연구하여 고려할 가용성 솔루션을 결정합니다. 다음 항목은 가용성 전략을 결정하도록 도와주기 위해 수집한 정보 유형의 예입니다.

- 가용성에 대해 9 를 몇 개 지정합니까?
- 페일오버 상황과 관련된 성능 사양이 있습니까 (예: 페일오버 동안 최소한 성능의 50%)?
- 사용 사례와 사용 분석에서 최고 및 최고가 아닌 시간을 식별합니까?
- 잠재적인 성능 요구 사항은 무엇입니까?
- 지리적인 고려 사항이 있습니까?

각 구성 요소에 대해 사용 사례를 분석하여 페일오버와 로드 균형 조정 요구 사항에 가장 잘 맞는 솔루션을 결정합니다. 또한 사용 사례와 사용 분석을 고려하여 로드 균형 서비스에 가장 적합한 방법을 결정합니다.

선택하는 가용성 전략에서는 65페이지의 "[서비스 가능성 문제점](#)"에 설명되어 있는 서비스 가능성 요구 사항도 고려해야 합니다. 관리하기 쉬운 시스템을 위하여 상당한 관리 및 유지 보수가 필요한 복잡한 솔루션은 피합니다.

복잡한 시스템의 디렉토리 설계

많은 수의 사용자에 대한 복잡한 배포의 경우 가용성 전략에 영향을 미칠 수 있는 Directory Server의 디렉토리 설계가 필요할 수 있습니다. 이는 LDAP 디렉토리 설계가 Identity Server 및 Messaging Server의 가용성 전략에 영향을 미치고, 차례로 다른 시스템 품질에도 영향을 미칠 수 있기 때문입니다.

복잡한 배포를 설계할 경우 가용성 설계를 지원하는 임시 디렉토리 설계를 작성하는 것을 고려해 봅니다. 나중에 세부 설계 지정이나 개발 단계에서 완전한 디렉토리 설계를 제공합니다.

하드웨어 및 소프트웨어 오류

가용성 설계에서는 하드웨어와 소프트웨어 오류를 방지해야 합니다. 일반적으로 소프트웨어 오류가 하드웨어 오류보다 비용이 높습니다. 소프트웨어 오류의 평균 시간이 하드웨어 오류보다 많습니다. 또한 소프트웨어 오류가 진단 및 복구가 더 어렵고 이를 방지하기 위한 관리 및 유지 보수 비용이 더 많이 듭니다.

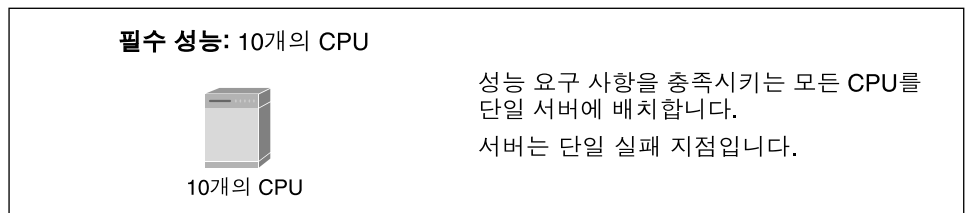
가용성에 대한 일반적인 접근 방법

이 절에서는 가용성 요구 사항을 설계할 수 있는 일반적인 방법을 설명합니다. 특정 가용성 설계는 이 백서의 범위를 벗어납니다.

단일 서버 시스템

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

그림 5-5 단일 서버



Sun에서는 다음 이점을 제공하는 고성능 서버를 제공합니다.

- 시스템 실행 중에 하드웨어 구성 요소의 교체 및 재구성
- 서버의 무결함 도메인에서 복수 응용 프로그램 실행

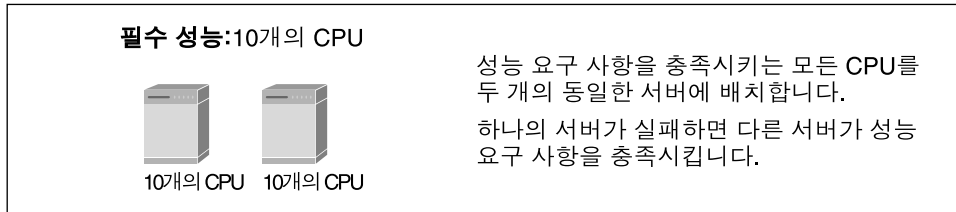
- 시스템을 다시 부팅하지 않고도 용량, 성능 속도 및 I/O 구성 업그레이드

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. 그리고 다중 서버 시스템인 경우 로드 균형 조정, 페일오버 및 단일 실패 지점 제거에서 더 유연합니다.

수평으로 중복된 시스템

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

그림 5-6 2 개의 복제 서버

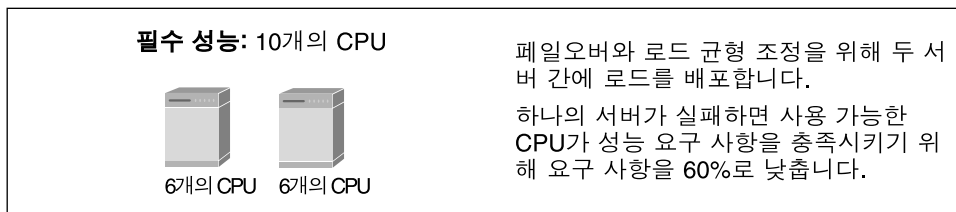


위 그림 5-6에서 각 서버의 컴퓨팅 성능은 동일합니다. 하나의 서버만 성능 요구 사항을 처리합니다. 다른 서버는 백업으로 서비스에 호출된 경우 100%의 성능을 제공합니다.

복제 서버 설계의 장점은 페일오버 상황에서 100% 성능을 제공한다는 점입니다. 단점은 하드웨어 비용 증가에 해당하는 전체적인 성능 증가가 없는 것입니다.

다음 그림에서는 로드 균형 조정 및 페일오버를 위해 두 서버 간에 성능을 배포하는 시나리오를 설명합니다.

그림 5-7 두 서버 간의 로드 배포



위 **그림 5-7**에서는 한 서버가 실패하더라도 전체 용량의 일부이긴 하지만 모든 서비스가 여전히 사용 가능합니다. 남은 서버에서 10 개 CPU 요구 사항의 60% 인 6 개 CPU 의 컴퓨팅 성능을 제공합니다.

이 설계의 장점은 두 서버를 모두 사용할 수 있는 경우 추가로 2 개의 CPU 잠재 용량이 있다는 점입니다. 또한 한 서버가 실패할 경우 비록 감소된 성능으로라도 모든 서비스를 사용할 수 있습니다.

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

그림 5-8 n 개의 서버 간 로드 배포

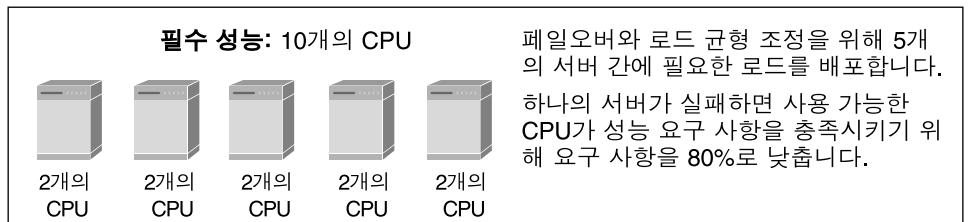


그림 5-8 과 같이 이 설계에 다섯 개의 서버가 있기 때문에 한 서버가 실패할 경우 나머지 서버에서 10 개 CPU 성능 요구 사항의 80% 인 총 8 개 CPU 의 컴퓨팅 성능을 제공합니다. 설계에 2 개 CPU 용량을 가진 서버를 추가할 경우 결과적으로 N+1 설계가 됩니다. 한 서버가 실패하면 나머지 서버에서 성능 요구 사항의 100% 를 충족시킵니다.

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 듬

그러나 서버를 추가할 경우 관리 및 유지 보수 비용이 상당히 늘어날 수 있습니다. 또한 데이터 센터의 서버에 대한 호스팅 비용도 있습니다. 일정 시점이 되면 서버 추가에 따른 수익이 줄어들기 시작합니다.

Sun Cluster

높은 수준의 가용성이 필요한 경우 (예 : 4 개 또는 5 개의 9) Sun Cluster 를 가용성 설계의 일부로 고려할 수 있습니다. 클러스터 시스템은 서버, 저장소 및 기타 네트워크 자원을 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 시스템의 사용자 서비스를 방해하지 않고 비교적 빠르게 수행됩니다.

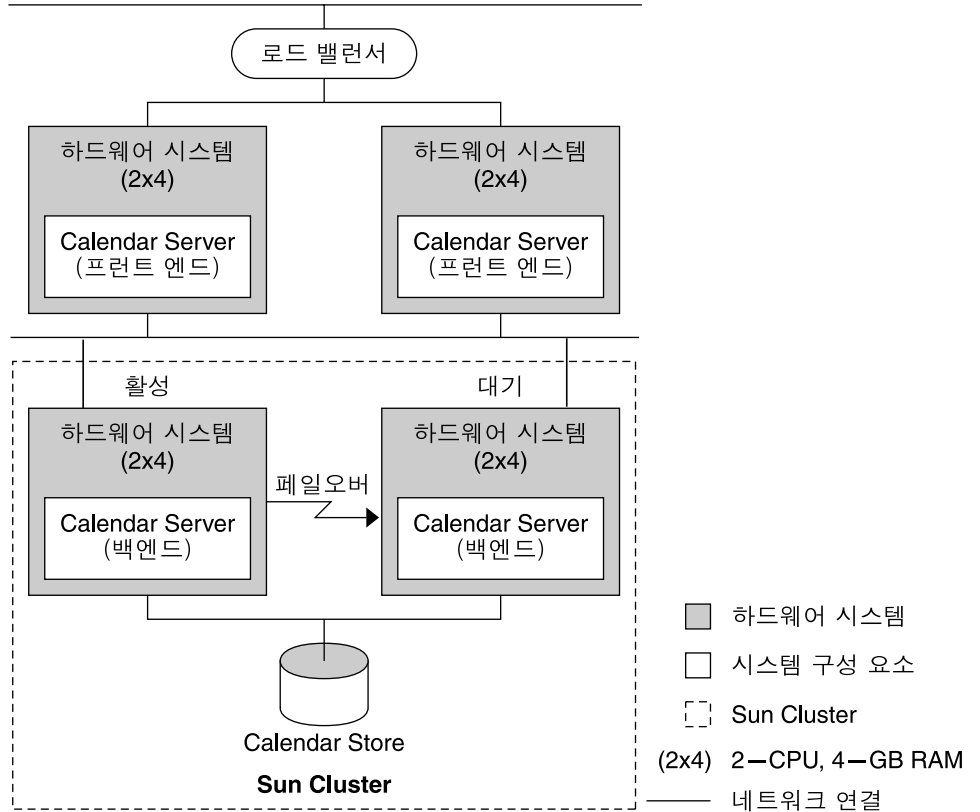
Sun Cluster 를 사용하려면 추가 전용 하드웨어와 구성, 관리 및 유지 보수를 위한 특수 기술이 필요합니다.

샘플 배포의 가용성 설계

다음 그림에서는 4 장, “논리 구조 설계”에서 소개한 예제 배포의 달력 서비스 부분에 대한 가용성 설계를 보여 줍니다. 이 그림은 예제 배포에 대한 논리 구조의 Calendar Server 부분을 위한 가용성 솔루션을 설명합니다. 예제 배포의 전체 가용성 솔루션에 대한 분석은 이 백서의 범위를 벗어납니다.

이 장 앞 부분의 크기 조정 연습에서는 그림 5-3 페이지 57에서 설명한 것처럼 Calendar Server 에 6 개의 CPU 와 12GB 의 메모리가 필요한 것으로 결정했습니다. 다음 그림에서는 들어오는 요청과 나가는 요청의 로드 균형 조정을 위해 두 서버에 배포한 Calendar Server 의 프런트 엔드를 보여 줍니다. Calendar Server 의 백 엔드는 별도의 서버에 배포되며, 페일오버를 위해 Sun Cluster 3.1 4/04 에 복제됩니다. 페일오버를 위해 Calendar Server 백 엔드에 필요한 CPU 및 메모리가 Sun Cluster 3.1 4/04 에 복제됩니다.

그림 5-9 예제 배포에서 Calendar Server 에 대한 가용성 설계



서비스 가능성 문제점

가용성을 설계할 때 솔루션의 관리 및 유지 보수 비용도 고려해야 합니다. 이 비용은 하드웨어 구입과 특별히 관련이 없기 때문에 설계에서 이 비용을 빠뜨리는 경우가 종종 있습니다. 오히려 설계의 복잡도를 반영하는 진행 중인 숨겨진 비용일 수 있습니다.

예를 들어, 설계에 높은 가용성을 제공하는 수평으로 중복된 여러 서버를 포함시킬 수 있습니다. 그러나 서버 설정 및 구성 비용을 고려하지 않고 소프트웨어를 계속 업그레이드하며 시스템 상태를 모니터링할 경우 가용성 증가가 손상될 수 있습니다.

서비스 가능성을 설계할 경우 다음과 같은 관리 및 유지 보수 비용을 고려합니다.

- 설정 및 구성
- 모니터링

- 서버 하드웨어 업그레이드
- 서버 소프트웨어 업그레이드
- 파일오버 자동화

확장성을 위한 크기 조정

확장성은 대개 시스템 자원을 추가하지만 배포 구조는 변경하지 않고 시스템에 용량을 추가할 수 있는 기능을 나타냅니다. 이 절에서는 확장성을 설계할 때 고려해야 할 항목을 설명합니다.

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.

잠재 용량

잠재 용량은 추가 성능과 가용성 자원을 시스템에 포함시켜서 비정상적인 최고 로드를 쉽게 처리할 수 있도록 하는 확장성의 한 요소입니다. 잠재 용량은 설계에 안전을 확립하는 한 가지 방법입니다.

사용 사례를 신중하게 분석하면 비정상적인 최고 로드를 작성할 수 있는 시나리오 (예를 들어, 필수 웹 캐스팅을 예약하는 B2E 배포) 를 식별하는 데 도움이 될 수 있습니다. 예상치 않은 증가를 처리하고 시스템에 안전을 확립하는 잠재 용량을 설계하는 요소와 더불어 이와 같은 비정상적인 최고 로드를 분석합니다.

또한 잠재 용량이 배포 시스템에서 사용되는 방법을 모니터링하여 언제 자원을 추가하여 시스템을 확장해야 할지를 결정할 수 있습니다.

시스템의 용량 업그레이드

시스템 설계는 운영의 처음 6-12 개월 동안 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.

비즈니스 또는 기술적인 이유로 시스템 일부의 배포를 지연하는 증분 배포를 수행할 경우 시스템의 새로운 기능을 포함하는 다른 업그레이드와 일치하도록 시스템의 용량 업그레이드를 예약할 수 있습니다.

자원 최적화

배포 크기 조정은 시스템 요구 사항을 충족시키기 위한 자원에 대한 예측일 뿐 아니라 위험 관리 및 자원 관리 모두에 대한 연습이기도 합니다. 설계에서 위험 관리와 자원 관리를 처리하는 방법이 비즈니스 목표를 충족시키는 열쇠가 되기도 합니다.

위험 관리

비즈니스 요구 사항과 사용량 분석 같은 크기 조정의 기반이 되는 정보 대부분은 경험적인 데이터가 아니라 예측과 예상을 기반으로 하는 데이터입니다. 계획된 배포의 크기 조정을 완료하기 전에 데이터를 다시 고려하여 크기 조정 설계에서 예측 또는 예상들 간의 적절한 편차를 고려하고 있는지 확인합니다.

예를 들어, 비즈니스 요구 사항의 예측이 시스템의 실제 사용량에 못 미칠 경우 발생하는 트래픽 양을 처리할 수 없는 시스템을 구성할 수 있는 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.

덜 사용한 자원을 성공에 필수적인 다른 영역에 적용할 수 있는 기회를 잃기 때문에 자원을 지나치게 사용하면 설계가 실패할 수도 있습니다. 또한 지나친 솔루션도 계약을 성실하게 이행하지 못한 것으로 간주할 수 있습니다.

자원 관리

자원 관리는 사용 가능한 모든 크기 조정 옵션을 분석하고 비용을 최소화하면서도 시스템 요구 사항을 이행하는 가장 적합한 솔루션을 선택하는 과정입니다. 여기에는 한 영역의 이점이 다른 영역의 비용으로 상쇄되지 않도록 각 설계 결정 간의 균형을 조절하는 방법을 이해하는 것이 포함됩니다.

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.

크기 조정 전략을 완료하기 전에 결정을 검토하여 자원 사용과 설계의 전체적인 이점 간의 균형이 맞는지 확인합니다. 여기에서는 일반적으로 한 영역의 시스템 품질이 다른 시스템 품질에 어떤 영향을 미치는지도 검토합니다. 다음 표에서는 자원 관리를 위해 고려할 수 있는 일부 항목을 나열합니다.

표 5-2 자원 관리 항목

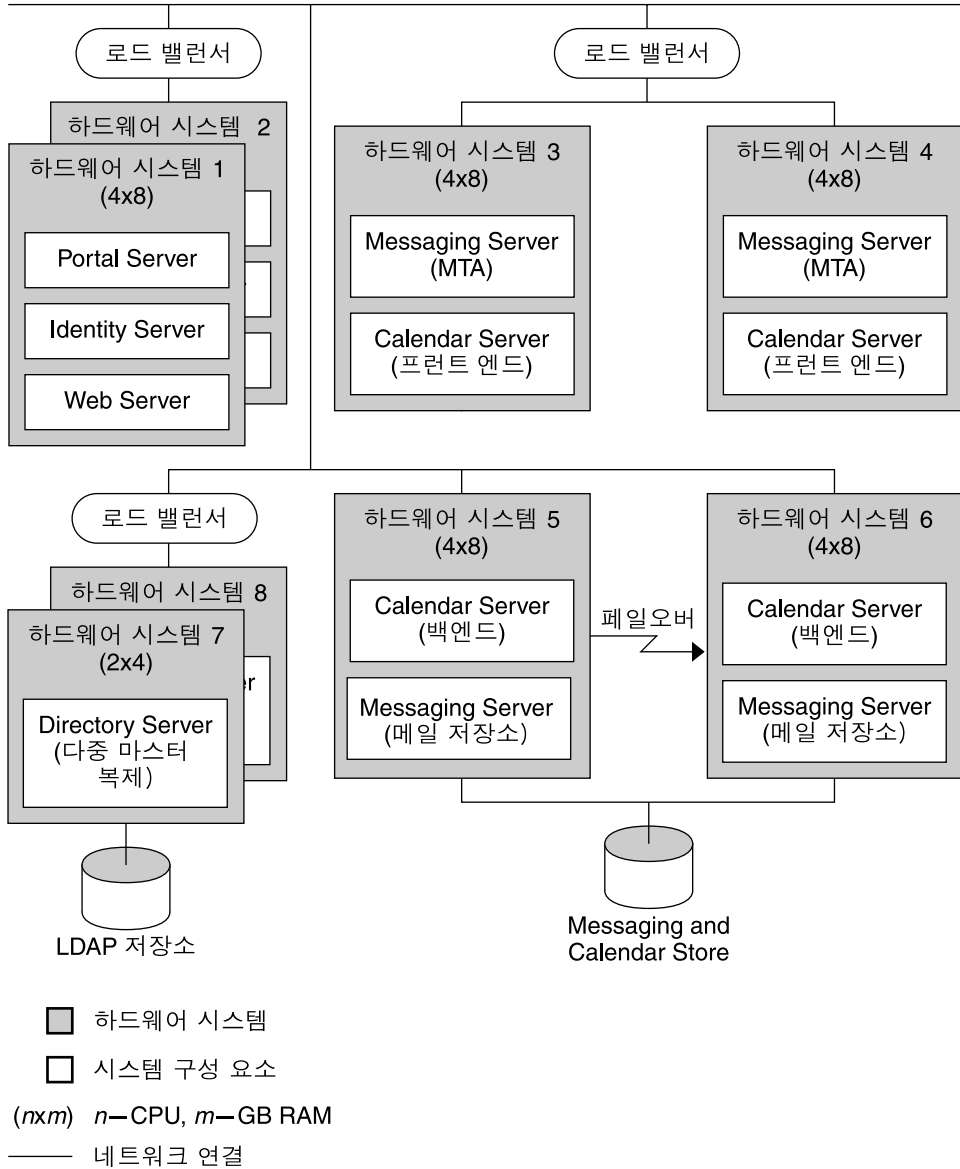
항목	설명
성능	개별 서버의 CPU 에 집중하는 성능 솔루션의 경우 서비스에서 컴퓨팅 성능을 효율적으로 이용할 수 있습니다. 예를 들어, 일부 서비스의 경우 효율적으로 사용할 수 있는 CPU 개수에 한계가 있습니다.
잠재 용량	성능 예상치를 초과하는 로드를 처리하기 위한 전략이 있습니까? 서버의 세로 확장, 다른 서버에 대한 로드 균형 조정 또는 둘 다를 사용하여 과도한 로드를 처리합니까? 배포 확장을 위한 다음 시점까지 비정상적인 최고 로드를 처리할 만큼 잠재 용량이 충분합니까?
보안	보안 트랜잭션을 처리하는 데 필요한 성능 오버헤드를 충분히 고려했습니까?
가용성	가로로 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 고려했습니까?
확장성	배포를 확장하기 위한 중요 시점을 예상했습니까? 배포 확장의 중요 시점까지 예상되는 로드 증가를 처리할 충분한 잠재 용량을 제공하는 전략이 있습니까?
서비스 가능성	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?

예제 배포 구조

다음 그림에서는 백서 앞 부분에서 소개한 예제 배포의 전체 배포 구조를 보여 줍니다. 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



세부 설계 지정

배포 구조를 완료한 후 고객이 검토하는 기간이 있고 프로젝트 승인이 뒤를 잇습니다. 일부 경우 고객이 승인하기 전에 배포 구조를 변경하라고 다시 지시할 수도 있습니다.

프로젝트 승인 후 배포 구현의 출발점이 되는 세부 설계 지정을 작성합니다. 설계 지정에는 자세한 LDAP 디렉토리 지정뿐만 아니라 특정 하드웨어 자원 및 네트워크 장치에 대한 세부 사항도 포함됩니다.

세부 설계 지정

배포 설계 구현

이 장에서는 배포 설계를 구현하는 데 필요한 단계의 개요를 제공합니다.

배포 구조를 승인하고 세부 설계 지정을 완료한 후 배포 계획의 구현 단계에 들어갑니다. 구현 단계 중에는 배포 구조를 작성합니다. 배포 프로젝트의 특성에 따라 배포 설계 구현에는 다음 단계 중 일부 또는 전부가 포함될 수 있습니다.

- 테스트 환경에서 파일럿이나 프로토타입 작성 및 배포
- 시스템 요구 사항 호환성을 측정하기 위한 기능 테스트 설계 및 실행
- 최고 로드 시 성능을 측정하기 위한 스트레스 테스트 설계 및 실행
- 단계적으로 생산에 들어갈 수 있는 생산 배포 작성

이 장의 내용은 다음과 같습니다.

- 73 페이지의 “ 파일럿 및 프로토타입 개발 ”
- 74 페이지의 “ 파일럿 및 프로토타입 배포 테스트 ”
- 75 페이지의 “ 프로덕션 배포 실행 ”

파일럿 및 프로토타입 개발

Java Enterprise System 배포는 대개 두 가지 범주에 해당하는데, 주로 Java Enterprise System 와 함께 제공된 서비스를 기반으로 한 범주와 Java Enterprise System 서비스와 통합된 많은 사용자 정의 서비스가 필요한 범주입니다. 전자 유형의 배포를 80:20 *비율* (Java Enterprise System 에서 제공하는 서비스의 80%) 라고 간주할 수 있고, 마찬가지로 전자를 20:80 *비율*라고 간주할 수도 있습니다.

80:20 배포의 경우 구현 단계 중에 대개 테스트를 위해 *파일럿 배포*를 개발합니다. 80:20 배포에서 "최신" 기능을 제공하는 성숙한 Java Enterprise System 서비스를 사용하기 때문에 파일럿 배포는 개발, 테스트 및 수정 단계에서 프로덕션 개발로 비교적 빠르게 이동합니다.

한편 20:80 배포에서는 80:20 배포와 함께 제공되는 상호 운용성 기록을 갖고 있지 않은 새로운 사용자 정의 서비스를 소개합니다. 이런 이유 때문에 프로덕션하기 전에 엄격한 개발, 테스트, 수정 주기가 필요한 POC (Proof-of-concept) 배포인 *프로토타입 배포*를 작성합니다.

주 실제 엔터프라이즈 배포는 필요한 사용자 정의 서비스 개발 양에 있어서 크게 다를 수 있습니다. 배포의 복잡도와 특성에 따라 테스트를 위해 파일럿이나 프로토타입 배포를 사용할지 여부가 달라집니다.

파일럿 및 프로토타입 배포 테스트

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

이상적으로 *기능 테스트*에서는 모든 식별된 사용 사례를 기반으로 시나리오의 모형을 만듭니다. 호환성을 측정하기 위해 메트릭스 세트를 개발해야 합니다. 기능 테스트에는 비즈니스 요구 사항을 충족시키는지 확인하기 위해 베타 사용자 그룹을 선택하기 위한 제한된 배포가 포함될 수도 있습니다.

*스트레스 테스트*에서는 최고 로드 시 성능을 측정합니다. 이 테스트에서는 대개 시뮬레이트된 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 지정의 문제점을 표시할 수 있고, 배포를 프로덕션 환경으로 실행하려면 일부 설계, 빌드, 테스트 반복이 포함될 수 있습니다. 그러나 테스트가 배포 구조의 문제점을 발견할 수 있는 위치가 될 수 없습니다. 배포 구조의 테스트 단계에서 설계 문제를 발견한 경우 분석, 계획 및 설계를 실패로 간주할 수 있습니다.

프로덕션 배포 실행

파일럿이나 POC 배포가 테스트 기준을 통과하면 배포를 프로덕션 환경으로 실행할 준비가 됩니다. 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 서비스 세트로 시작하여 나머지 서비스를 단계별로 실행할 수 있습니다. 서비스를 단계별로 실행하면 프로덕션 환경에서 서비스에 발생할 수 있는 문제점을 격리, 식별 및 해결하는 데 도움이 될 수 있습니다.

테스트에서 프로덕션 환경을 완벽하게 시뮬레이트하지 못하기 때문에 배포된 시스템을 계속 모니터링하여 조정, 유지 보수 또는 서비스가 필요한 영역을 식별해야 합니다.

A

Application Server 12

C

Calendar Server 12

D

Directory Proxy Server 12

Directory Server 12

I

Identity Server 12

Instant Messaging 12

J

J2EE

컨테이너 서비스 12

플랫폼 12

JMS (Java Message Service) 12

M

Message Queue 12

Messaging Server 13

P

Portal Server 13

Portal Server, Secure Remote Access 13

S

Sun Cluster 13

Sun ONE 제품

Application Server, Application Server 참조

Calendar Server, Calendar Server 참조

Directory Proxy Server, Directory Proxy Server [참조](#)

Directory Server, Directory Server [참조](#)

Identity Server, Identity Server [참조](#)

Instant Messaging, Instant Messaging [참조](#)

Message Queue, Message Queue [참조](#)

Messaging Server, Messaging Server [참조](#)

Portal Server, Portal Server [참조](#)

Portal Server, Secure Remote Access, Portal Server, Secure Remote Access [참조](#)

Web Server, Web Server [참조](#)

W

Web Server 13

ㄱ

구성 요소 제품, 설명 12

ㄴ

시스템 구성 요소, 구성 요소 제품 12