

WebSphere Application Server



Edge Components용 개념, 계획 및 설치

버전 6.1

WebSphere Application Server



Edge Components용 개념, 계획 및 설치

버전 6.1

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 91 페이지의 『주의사항』의 일반 정보를 읽으십시오.

초판(2006년 5월)

이 책은 다음에 적용됩니다.

WebSphere Application Server, 버전 6.1

새 개정판에 달리 명시되지 않는 한 후속 릴리스 및 수정판에 적용됩니다.

IBM 담당자 또는 해당 지역의 IBM 지사를 통해 책을 주문할 수 있습니다.

© Copyright International Business Machines Corporation 2006. All rights reserved.

목차

그림	v
이 책에 대한 정보	vii
이 책의 사용자	vii
내게 필요한 옵션	vii
이 책에 사용된 규약 및 용어	vii

제 1 부 개요 1

제 1 장 WebSphere Application Server Edge Components 소개	3
Caching Proxy	4
Load Balancer	5
Dispatcher	6
Content Based Routing	6
Site Selector	7
Cisco CSS Controller	7
Nortel Alteon Controller	7
Metric Server	7

제 2 장 Edge Components 및 WebSphere 그룹	9
Tivoli Access Manager	9
WebSphere Portal Server	9
WebSphere Site Analyzer	10
WebSphere Transcoding Publisher	10

제 3 장 Application Server 및 Edge Components에 대한 자세한 정보	11
---	----

제 2 부 Edge Components 개념 및 설명 13

제 4 장 캐싱	15
기본 Caching Proxy 구성	15
역방향 Caching Proxy(기본 구성)	15
전방향 Caching Proxy	16
고급 캐싱	19
로드 밸런싱된 Caching Proxy 클러스터	19
동적 콘텐츠 캐싱	20
추가 캐싱 기능	21

제 5 장 네트워크 성능	23
네트워크 하드웨어	23
메모리 고려사항	23

하드 디스크 고려사항	24
네트워크 고려사항	24
CPU 고려사항	24
네트워크 구조	24
웹 사이트 관심 및 프록시 서버 로드 고려사항	25
통신량 유형 고려사항	26

제 6 장 가용성	27
로드 밸런싱	27
다중 콘텐츠 호스트의 로드 밸런싱	27
다중 역방향 프록시 서버의 로드 밸런싱	29
다중 전방향 프록시 서버를 사용하는 Load Balancer	30
실패복구 지원	33

제 7 장 Content Based Routing 37

제 3 부 시나리오 41

제 8 장 B2C 네트워크	43
1단계	43
2단계	45
3단계	46

제 9 장 B2C banking 솔루션 49

제 10 장 웹 포털 네트워크 51

제 4 부 Edge Components 설치 53

제 11 장 Edge Components에 대한 요구사항	55
하드웨어 및 소프트웨어 전제조건	55
Caching Proxy 구성 및 관리 양식에 브라우저 사용	55
Load Balancer 온라인 도움말에 브라우저 사용	57

제 12 장 설치 프로그램을 사용한 Edge Components 설치	59
Windows용 설치 프로그램 사용	59
Linux 및 UNIX용 설치 프로그램 사용	61

제 13 장 시스템 패키징 도구를 사용한 Caching Proxy 설치	63
시스템 도구를 사용하여 Caching Proxy 설치 제거	65

제 14 장 시스템 패키징 도구를 사용한 Load Balancer 설치	67
AIX 설치	68
설치하기 전에	68
설치 프로시저	69
HP-UX에 설치	71
설치하기 전에	71
설치 프로시저	71
Linux에 설치	73
설치하기 전에	73
설치 단계	74
Solaris에 설치	75
설치하기 전에	75
설치 단계	76
<hr/>	
제 5 부 Edge Components를 사용하여 네트워크 빌드	79
제 15 장 Caching Proxy 네트워크 작성	81

Workflow	82
필수 컴퓨터 시스템 및 소프트웨어 검토	82
빌드 서버 1(Linux 및 UNIX 시스템)	82
서버 1 작성(Windows 시스템)	82
서버 1 구성	83
Caching Proxy 네트워크 검사	84
제 16 장 Load Balancer 네트워크 작성	85
Workflow	85
필수 컴퓨터 시스템 및 소프트웨어 검토	85
네트워크 구성	86
Dispatcher 구성	87
명령행을 사용한 구성	87
구성 마법사를 사용한 구성	88
GUI를 사용한 구성	89
Load Balancer 네트워크 검사	89
주의사항	91
상표	93

그림

1. 역방향 프록시로 작동하는 Caching Proxy	16	10. 웹 콘텐츠를 최대한으로 사용 가능하게 하기 위해 기본 및 백업 Load Balancer 사용.	34
2. 전방향 프록시로 작동하는 Caching Proxy	17	11. 콘텐츠 호스트에서 백업 Load Balancer 찾기	35
3. 투명 전방향 프록시로 작동하는 Caching Proxy	19	12. CBR로 HTTP 요청 경로 지정	38
4. 로드 밸런싱된 클러스터에 대해 프록시 서버의 역할을 하는 Caching Proxy	20	13. CBR로 라우트된 HTTP 요청의 로드 밸런싱	39
5. 다중 콘텐츠 호스트의 로드 밸런싱	28	14. B2C 네트워크(1단계)	44
6. 다중 역방향 프록시 서버 및 콘텐츠 호스트의 로드 밸런싱	29	15. B2C 네트워크(2단계)	46
7. Dispatcher를 사용하여 여러 개의 Caching Proxy 로드 밸런싱	31	16. B2C 네트워크(3단계)	47
8. 기본 및 백업 Dispatcher를 사용하여 인터넷 액세스의 높은 가용성을 제공	32	17. B2C बैंक 솔루션	50
9. Caching Proxy 시스템에서 백업 Dispatcher 찾기	33	18. 웹 포털	52
		19. Caching Proxy 데모 네트워크	81
		20. Load Balancer 데모 네트워크	85

이 책에 대한 정보

*WebSphere® Application Server Edge Components*용 개념, 계획 및 설치에서는 WebSphere Application Server Edge Components에 대해 소개합니다. 이 책에서는 상위 레벨의 제품 개요, 핵심 컴포넌트에 대한 자세한 기능 설명, 네트워크 경계 시나리오, 설치 및 초기 구성 정보, 데모 네트워크를 제공합니다.

이 책의 사용자

*WebSphere Application Server Edge Components*용 개념, 계획 및 설치는 운영 체제 및 인터넷 서비스 제공에 익숙하고 숙련된 네트워크 및 시스템 관리자를 위한 것입니다. WebSphere Application Server 또는 WebSphere Application Server Edge Components에 대한 경험이 있어야 하는 것은 아닙니다.

내게 필요한 옵션

내게 필요한 옵션 기능은 지체 부자유 또는 시각 장애와 같은 신체 장애를 가진 사용자가 소프트웨어 제품을 잘 사용할 수 있도록 도와줍니다. 다음은 WebSphere Application Server, 버전 6.1의 내게 필요한 옵션 기능입니다.

- 화면 판독기 소프트웨어와 디지털 음성 신디사이저를 사용하면 화면에 표시된 내용을 들을 수 있습니다. IBM® ViaVoice™와 같은 음성 인식 소프트웨어를 사용하여 데이터를 입력하고 사용자 인터페이스를 탐색할 수 있습니다.
- 모든 기능은 마우스 대신 키보드를 사용하여 작동할 수 있습니다.
- 제공된 그래픽 인터페이스 대신 표준 문서 편집기나 명령행 인터페이스를 사용하여 Application Server 기능을 구성하고 관리할 수 있습니다. 특정 기능의 액세스에 필요한 옵션에 대한 자세한 정보는 해당 기능 문서를 참조하십시오.

이 책에 사용된 규약 및 용어

이 문서에는 다음과 같은 서체와 키 규약을 사용합니다.

표 1. 이 책에 사용된 규약

규약	의미
굵은체	GUI와 관련될 때, 굵은체로 메뉴, 메뉴 항목, 레이블, 단추, 아이콘 및 폴더를 나타냅니다. 또한 주위의 텍스트와 혼동될 수 있는 명령 이름을 강조하는 데에도 사용될 수 있습니다.
모노스페이스	명령 프롬프트에 입력해야 할 텍스트를 모노스페이스로 표시합니다. 화면 텍스트, 코드 예제 및 파일 발췌 부분도 나타냅니다.
기울임꼴	제공해야 하는 변수값을 기울임꼴로 나타냅니다. (예를 들어, <i>fileName</i> 에 파일 이름을 제공합니다.) 강조 및 책 제목을 표시합니다.

표 1. 이 책에 사용된 규약 (계속)

규약	의미
Ctrl-x	여기서, x는 키 이름으로 제어 문자 순서를 표시합니다. 예를 들어, Ctrl-c는 Ctrl 키를 누른 상태에서 c 키를 누르는 것을 의미합니다.
Return	Return, Enter 또는 왼쪽 화살표로 표시된 키를 나타냅니다.
%	루트 권한이 필요하지 않은 명령의 Linux 및 UNIX® 명령 셸 프롬프트를 표시합니다.
#	루트 권한이 필요한 Linux 및 UNIX 명령 셸 프롬프트를 표시합니다.
C:\	Windows 명령 프롬프트를 표시합니다.
명령 입력	명령을 “입력” 또는 “실행”할 때 명령을 입력하고 Return을 누릅니다. 예를 들어, “Enter the ls command”라는 명령은 명령 프롬프트에 ls 를 입력하고 Return을 누르는 것을 의미합니다.
[]	구문 설명에 선택적 항목을 넣습니다.
{ }	선택할 항목이 있는 목록을 구문 설명에 넣습니다.
	구문 설명에서 { }에 있는 선택사항 목록의 항목을 구분합니다.
...	구문 설명에서 줄임표는 앞의 항목을 한 번 이상 반복할 수 있다는 것을 의미합니다. 예제에서 줄임표는 간결하게 하기 위해 예제에서 정보를 생략했음을 의미합니다.

제 1 부 개요

이 부분에서는 WebSphere Application Server Edge Components, Caching Proxy 및 Load Balancer를 소개하며 Application Server와의 통합에 대해 설명합니다. 또한 Caching Proxy 및 Load Balancer 컴포넌트도 정의합니다. 또한 기타 관련 WebSphere 그룹 제품도 소개합니다.

다음 장으로 구성되어 있습니다.

- 3 페이지의 제 1 장 『WebSphere Application Server Edge Components 소개』
- 9 페이지의 제 2 장 『Edge Components 및 WebSphere 그룹』
- 11 페이지의 제 3 장 『Application Server 및 Edge Components에 대한 자세한 정보』

제 1 장 WebSphere Application Server Edge Components 소개

WebSphere는 B2B e-commerce용 e-business 응용프로그램과 같은 차세대 e-business 응용프로그램을 개발, 전개 및 통합할 수 있도록 하는 인터넷 하부 구조 소프트웨어입니다. WebSphere 미들웨어는 간단한 웹 공개에서부터 엔터프라이즈 규모의 트랜잭션 처리에 이르는 비즈니스 응용프로그램을 지원합니다.

WebSphere 비즈니스의 기초로서 WebSphere Application Server는 사용자가 비즈니스 응용프로그램을 설계, 구현 및 관리할 수 있게 하는 포괄적인 미들웨어 세트를 제공합니다. 이 응용프로그램의 범위는 웹 사이트 상점 입구에서부터 완벽한 조직의 컴퓨팅 하부 구조의 개정에 이를 수 있습니다.

개인화와 같은 프로세서 위주의 기능은 모든 e-business에 경쟁력 있는 장점을 제공합니다. 그러나 습관적으로 이러한 기능을 중앙 서버에 이관하면 유용한 기능이 인터넷으로 확장되지 못할 수 있습니다. 그러므로 지속적으로 새 웹 응용프로그램을 추가할 경우 비즈니스의 인터넷 하부 구조의 범위 및 영향력도 커져야 합니다. 또한 신뢰성 및 보안도 e-business에는 매우 중요합니다. 잠시만 서비스가 중단되어도 비즈니스가 손실을 입을 수 있습니다.

Edge Components(이전 Edge Server)는 이제 WebSphere Application Server 오퍼링의 일부입니다. Edge Components는 WebSphere Application Server와 함께 웹 서버에 대한 클라이언트 액세스를 제어하고, 비즈니스 엔터프라이즈가 인터넷 또는 회사 인트라넷을 통해 웹 기본 콘텐츠에 액세스하는 사용자에게 보다 향상된 서비스를 제공할 수 있게 하는 데 사용될 수 있습니다. Edge Components를 사용하면 웹 서버의 혼잡을 줄이고 콘텐츠 가용성을 늘리며 웹 서버의 성능을 향상시킬 수 있습니다. 이름에 표시된 대로 Edge Components는 일반적으로 엔터프라이즈의 인트라넷과 인터넷 간의 경계에 인접한(네트워크 구성 의미에서) 시스템에서 실행합니다.

WebSphere Application Server는 Caching Proxy 및 Load Balancer Edge Components를 포함합니다.

중요사항: Caching Proxy는 다음 경우를 제외하고 모든 Edge Components 설치에 사용할 수 있습니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행되는 Edge Components 설치에 사용할 수 없습니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge Components 설치에 사용할 수 없습니다.

Caching Proxy

Caching Proxy는 하나 이상의 백엔드 콘텐츠 서버에 대한 연결 지점을 제공하여 대역 폭 사용을 줄이고 웹 사이트의 속도 및 신뢰성을 개선합니다. Caching Proxy는 정적 콘텐츠 및 WebSphere Application Server에서 동적으로 생성된 콘텐츠를 캐시 및 제공할 수 있습니다.

Caching Proxy는 네트워크의 연결 지점 또는 요청 및 응답 시간 개선을 위해 작업하는 내부 네트워크 서버를 제공하는 역방향 프록시 서버(기본 구성) 또는 전방향 프록시 서버의 역할로 구성될 수 있습니다. 역방향 및 전방향 구성에 대한 자세한 정보는 15 페이지의 『기본 Caching Proxy 구성』을 참조하십시오.

프록시 서버는 클라이언트의 데이터 요청을 차단하고 콘텐츠 호스팅 시스템에서 요청 정보를 검색하며, 해당 콘텐츠를 클라이언트로 다시 전달합니다. 일반적으로, 대부분의 요청은 웹 서버 시스템(기점 서버 또는 콘텐츠 호스트라고도 부름)에 저장되어 있는 문서에 대한 것으로 HTTP(Hypertext Transfer Protocol)를 통하여 전달됩니다. 그러나 프록시 서버를 구성하면 FTP(File Transfer Protocol) 및 Gopher와 같은 다른 프로토콜을 처리할 수 있습니다.

프록시 서버는 캐시 가능한 콘텐츠를 요청자에게 전달하기 전에 로컬 캐시에 저장합니다. 캐시 가능한 콘텐츠의 예에는 정적 웹 페이지와 동적으로 생성되지만 자주 변경되는 정보가 포함된 JSP(JavaServer Page) 파일이 있습니다. 캐시를 사용하면 프록시 서버가 콘텐츠 호스트에서 다시 검색하는 것보다 훨씬 빨리 로컬 캐시에서 직접 전달하여, 동일한 콘텐츠에 대한 후속 요청을 충족시킬 수 있습니다.

Caching Proxy용 플러그인은 프록시 서버에 기능을 추가합니다.

- ICP 플러그인으로 인해 프록시 서버가 HTML 페이지 및 기타 캐시 가능 자원 검색 시 기타 ICP(Internet Caching Protocol) 준수 캐시를 조회할 수 있습니다
- Tivoli® Access Manager(이전 Policy Director) 플러그인은 프록시 서버가 Access Manager의 통합 권한 부여 또는 인증 서비스를 사용할 수 있도록 합니다.
- PAC-LDAP 인증 모듈은 권한 또는 인증 루틴을 수행할 때 프록시 서버가 LDAP 서버에 액세스할 수 있도록 합니다.
- WebSphere Transcoding Publisher 플러그인을 WebSphere Transcoding Publisher와 함께 사용할 경우 프록시 서버는 모빌 장치에 대한 여러 개의 코드 변환된 버전을 캐시할 수 있습니다.

API(Application Programming Interface)에 대한 조정 플러그인 모듈을 작성하여 Caching Proxy의 기능을 좀더 확장할 수 있습니다. API는 유연성이 있으므로 사용하기 쉬우며 플랫폼에 독립적입니다. 프록시는 처리하는 각각의 클라이언트 요청에 대하여 일련의 단계를 수행합니다. 플러그인은 요청 처리 워크플로우(예: 클라이언트 인증 및 요청 필터) 내의 단계를 수정하거나 바꿀 수 있습니다. 예를 들어, 강력한 변형 인터

페이스는 HTTP 데이터에 대한 액세스를 제공하고 URL 및 웹 콘텐츠의 대체 또는 변환을 가능하게 합니다. 플러그인은 지정된 처리 단계를 수정하거나 바꿀 수 있으며 특정 단계에 대해 둘 이상의 플러그인을 호출할 수 있습니다.

Load Balancer

Load Balancer는 혼잡을 줄이고 여러 다른 서비스 및 시스템에서의 로드를 밸런싱하여 네트워크 통신 플로우를 조정하는 네트워크 경계 시스템을 작성합니다. Load Balancer는 사이트 선택, 워크로드 관리, 세션 유사성 및 투명한 실패복구를 제공합니다.

Load Balancer는 인터넷과 엔터프라이즈의 백엔드 서버(콘텐츠 호스트 또는 Caching Proxy 시스템일 수 있음) 사이에 설치됩니다. Load Balancer는 요청이나 콘텐츠가 많아서 엔터프라이즈가 백엔드 서버를 여러 개 사용하는 경우에도 인터넷에서 엔터프라이즈의 단일 연결 지점 모드로 사용됩니다. 또한 기본 Network Dispatcher가 일시적으로 장애를 일으킬 경우에 대신할 백업 Load Balancer를 설치하여 고가용성을 보장할 수 있습니다.

Load Balancer는 클라이언트에서 데이터 요청을 인터셉트하여 각 요청을 현재 요청을 가장 만족시킬 수 있는 서버로 전달합니다. 다시 말해서 동일한 유형의 요청을 서비스하는 정의된 시스템 집합 사이의 수신 요청 로드를 밸런싱합니다. Load Balancer는 WebSphere Application Server 및 Caching Proxy 시스템을 포함한 여러 유형의 서버로 요청을 분배할 수 있습니다. 사용자 정의 어드바이저를 사용하여 특정 응용프로그램 또는 플랫폼에 맞게 로드 밸런스를 사용자 정의할 수 있습니다. WebSphere Application Server의 로드 밸런스에 필요한 정보를 얻기 위한 특수 목적의 어드바이저가 사용 가능합니다.

Content Based Routing 컴포넌트를 Caching Proxy와 함께 설치할 경우, HTTP 및 HTTPS 요청을 URL 또는 관리자가 판단한 특성을 기본으로 하여 분배할 수 있으므로 모든 백엔드 서버에 동일한 콘텐츠를 저장할 필요가 없습니다. Dispatcher 컴포넌트는 HTTP 요청에 대해서도 동일한 기능을 제공할 수 있습니다.

로드 밸런스는 HTTP 서버, Application Server, 콘텐츠 서버의 대행자인 프록시 서버를 포함한 콘텐츠 서버를 투명하게 클러스터하여 웹 사이트의 가용성 및 확장성을 향상시킵니다. 가용성은 병렬화, 로드 밸런스 및 실패복구 지원을 통해 완성됩니다. 서버가 중단되면 비즈니스가 인터럽트됩니다. 백엔드 처리 성능을 투명하게 추가할 수 있기 때문에 하부 구조의 확장성이 크게 향상됩니다.

IPv6 지원: IPv6의 확장된 IP 주소 지정 설계 지원이 "IPv4 및 IPv6용 Load Balancer"를 통해 사용 가능합니다. IPv4 및 IPv6용 Load Balancer는 Dispatcher 컴포넌트만으로 구성된 개별 설치 이미지입니다. 이 설치 유형은 Dispatcher의 MAC 기반 패킷 전달을 사용하여 네트워크 내에 구성된 서버의 IPv4 및 IPv6 통신량 모두에 대한 로드 밸런스를 제공합니다. IPv4 및 IPv6용 Load Balancer를 설치하기 전에 반드시 이

전 Load Balancer를 설치 제거해야 합니다. 두 개의 Load Balancer를 동일한 시스템에 설치할 수 없습니다. (Dispatcher 컴포넌트의 간략한 개요는 『Dispatcher』를 참조하십시오.)

Load Balancer에는 다음과 같은 컴포넌트가 포함됩니다.

Dispatcher

HTTP, FTP, HTTPS 및 Telnet과 같은 모든 인터넷 서비스의 경우, Dispatcher 컴포넌트는 LAN 또는 WAN에서 서버의 로드 밸런스를 수행합니다. HTTP 서비스의 경우, Dispatcher는 클라이언트 요청의 URL 콘텐츠를 기본으로 서버의 로드 밸런스를 수행합니다.

Dispatcher 컴포넌트는 대규모의 확장 가능한 네트워크 서버를 안정적이고 효율적으로 관리할 수 있게 합니다. Dispatcher를 사용하여, 하나의 가상 서버에 여러 대의 개별 서버를 연결할 수 있습니다. 따라서 사용자 사이트는 전세계에서 하나의 IP 주소로 나타납니다.

IPv4 및 IPv6용 Load Balancer 설치를 사용하는 경우, *WebSphere Application Server Load Balancer 관리 안내서*에서 제한사항 및 구성 차이점에 대한 정보가 포함되어 있는 IPv4 및 IPv6용 Load Balancer에 Dispatcher 전개 장을 참조하십시오.

Content Based Routing

HTTP와 HTTPS 서비스의 경우, Content Based Routing 컴포넌트는 클라이언트 요청의 콘텐츠에 기반한 서버의 로드 밸런싱을 수행합니다. Content Based Routing은 Application Server Caching Proxy 컴포넌트와 함께 작동합니다.

중요사항: CBR(Content Based Routing) 컴포넌트는 다음 경우를 제외하고 지원되는 모든 플랫폼에서 사용 가능합니다.

- CBR은 64비트 JVM을 실행 중인 플랫폼에서 Load Balancer를 설치할 수 없습니다.

대신, 이 유형의 설치에 대해 Caching Proxy를 사용하지 않고 Load Balancer Dispatcher 컴포넌트의 cbr 전달 메소드를 사용하여 HTTP 및 HTTPS 요청의 Content Based Routing을 제공할 수 있습니다. 자세한 정보는 *WebSphere Application Server Load Balancer 관리 안내서*를 참조하십시오.

- CBR은 IPv4 및 IPv6용 Load Balancer 설치 시 사용할 수 없습니다.

IPv4 및 IPv6용 Load Balancer는 Dispatcher 컴포넌트의 mac 전달 메소드만을 지원합니다. nat 및 cbr 전달 메소드는 지원되지 않습니다.

Site Selector

Site Selector는 DNS 이름을 IP 주소에 매핑하여 네트워크 및 로드 밸런스 수신 요청에 대한 연결 지점 노드 역할을 하게 하여 로드 밸런스 시스템을 강화합니다. Metric Server와 함께 사용할 경우, Site Selector는 서버의 활동 레벨을 모니터링하여 서버에 로드가 가장 적은 시기와 실패한 서버를 발견합니다.

이 컴포넌트는 다음 경우를 제외하고 모든 Edge Components에서 지원됩니다.

- 이 컴포넌트는 IPv4 및 IPv6용 Load Balancer의 Edge Components와 함께 사용할 수 없습니다.

Cisco CSS Controller

Cisco CSS Controller 컴포넌트는 서버 선택, 로드 최적화 및 결함 허용을 위해 Cisco CSS Switch로 전송되는 서버 가중치 메트릭을 생성합니다.

이 컴포넌트는 다음 경우를 제외하고 모든 Edge Components에서 지원됩니다.

- 이 컴포넌트는 IPv4 및 IPv6용 Load Balancer의 Edge Components와 함께 사용할 수 없습니다.

Nortel Alteon Controller

Nortel Alteon Controller 컴포넌트는 서버 선택, 로드 최적화 및 결함 허용을 위해 Nortel Alteon Switch로 전송되는 서버 가중치 메트릭을 생성합니다.

이 컴포넌트는 다음 경우를 제외하고 모든 Edge Components에서 지원됩니다.

- 이 컴포넌트는 IPv4 및 IPv6용 Load Balancer의 Edge Components와 함께 사용할 수 없습니다.

Metric Server

Metric Server 컴포넌트는 로드 밸런싱이 된 서버에서 디먼으로 실행되며 시스템 로드 에 대한 정보를 Load Balancer에 제공합니다.

제 2 장 Edge Components 및 WebSphere 그룹

IBM WebSphere 그룹은 사용자의 e-business 기대를 실현할 수 있도록 설계되었습니다. 이는 사용자가 고성능 웹 사이트를 개발 및 관리하고 웹 사이트를 신규 또는 기존의 웹 비즈니스가 아닌 정보 시스템에 통합시킬 수 있게 하는 소프트웨어 제품 세트입니다.

WebSphere 그룹은 Edge Components를 포함한 WebSphere Application Server 및 WebSphere Application Server에 완전히 통합된 WebSphere 그룹 소프트웨어로 구성되어 성능이 향상됩니다. WebSphere Application Server 및 컴포넌트에 대한 개요는 3 페이지의 제 1 장 『WebSphere Application Server Edge Components 소개』를 참조하십시오.

Tivoli Access Manager

Tivoli Access Manager(이전 Tivoli Policy Director)는 별도로 사용 가능합니다. 이는 기존 웹 응용프로그램에 대한 액세스 제어 및 중앙 집중화된 보안을 제공하며, 한번의 인증으로 여러 웹 자원에 액세스할 수 있는 기능을 제공합니다. Caching Proxy 플러그인은 Access Manager의 보안 프레임워크를 활용하여 프록시 서버가 Access Manager의 통합 권한 부여 또는 인증 서비스를 사용할 수 있도록 합니다.

WebSphere Portal Server

WebSphere Portal Server(별도로 사용 가능)는 Portal에 연관된 프리젠테이션, 보안, 확장성, 가용성 문제점을 만족시키는 프레임워크를 제공합니다. 회사는 Portal Server로 자체 조정 Portal 웹 사이트를 작성하여 직원, 비즈니스 파트너 및 고객의 요구를 처리합니다. 사용자는 Portal에 사인온하여 그들이 필요로 하는 정보, 사람 및 응용프로그램에 대한 액세스를 제공하는 개인화된 웹 페이지를 받을 수 있습니다. 모든 필수 자원에 대한 이러한 개인화된 단일 액세스 지점은 정보의 과부하를 줄이고 생산성을 가속화시키며 웹 사이트 사용량을 늘립니다.

WebSphere Portal Server는 WebSphere Application Server 클러스터에서 실행하여 확장성 및 신뢰성을 얻습니다. 추가의 로드 밸런스 및 고가용성을 위해 Application Server Load Balancer 컴포넌트를 사용할 수도 있습니다.

WebSphere Site Analyzer

WebSphere Site Analyzer(별도로 사용 가능)를 사용하면 엔터프라이즈가 용량 및 성능 문제점을 예측할 수 있습니다. Site Analyzer와 함께 Caching Proxy와 Load Balancer 로그 및 기타 관리 가능 도구를 사용하여 웹 사이트 사용량을 모니터링, 분석 및 보고함으로써 추가 자원의 수요를 예측할 수 있습니다. 또한 Site Analyzer 관리 가능 컴포넌트는 Edge Components를 설치 및 업그레이드하며, 구성을 관리 및 저장하고, Edge Components를 원격으로 작동하고 이벤트를 탐색 및 보고하는 사용자를 지원합니다.

WebSphere Transcoding Publisher

WebSphere Transcoding Publisher(별도로 사용 가능)는 모바일 장치(예: 인터넷 기능 전화)에서 볼 수 있도록 웹 페이지를 변환하고 웹 콘텐츠를 사용자가 선호하는 자국어로 변환(WebSphere Translation Server를 눌러)하며 마크업 언어를 변환할 수 있습니다. Transcoding Publisher는 다른 장치 및 사용자에게 콘텐츠를 제공할 수 있게 함으로써 Caching Proxy의 성능을 향상시켰습니다. 웹 서버에서 콘텐츠에 액세스한 후 Transcoding Publisher를 호출하여 다양한 캐시 및 재사용이 가능하도록 데이터를 변환하고, 태그를 표시하도록 Caching Proxy의 변형 인터페이스를 구성할 수 있습니다. Caching Proxy의 사후 인증 인터페이스에서 Transcoding Publisher는 프록시 서버에서 사용자 및 장치 요구사항과 일치하는 콘텐츠가 있는지 확인한 다음, 일치하는 내용이 있으면 프록시 서버 캐시의 콘텐츠를 제공합니다.

제 3 장 Application Server 및 Edge Components에 대한 자세한 정보

WebSphere Application Server Edge Components에 특정한 다음 문서는 Edge Components Information Center에서 볼 수 있습니다.

- *Edge Components*용 프로그래밍 안내서 GA30-2915-00
- *Caching Proxy* 관리 안내서 GA30-2916-00
- *Load Balancer* 관리 안내서 GA30-2917-00

기타 WebSphere Application Server 문서는 WebSphere Application Server 라이브러리 페이지에서 사용 가능합니다.

Edge Components에 대한 기술 지원 정보는 WebSphere Application Server 지원 페이지에서 볼 수 있습니다.

다음은 Edge Components에 관한 정보 또는 관련 정보를 얻을 수 있는 웹 사이트 목록입니다.

- IBM 웹 사이트 홈: <http://www.ibm.com/>
- IBM WebSphere Application Server:
<http://www.ibm.com/software/webservers/appserv/>
- IBM WebSphere Application Server 라이브러리 웹 사이트:
<http://www.ibm.com/software/webservers/appserv/library.html>
- IBM WebSphere Application Server 지원 웹 사이트:
<http://www.ibm.com/software/webservers/appserv/support.html>
- IBM WebSphere Application Server Information Center:
<http://www.ibm.com/software/webservers/appserv/infocenter.html>
- IBM WebSphere Application Server Edge Components Information Center:
<http://www.ibm.com/software/webservers/appserv/ecinfocenter.html>

제 2 부 Edge Components 개념 및 설명

제 2 부에는 Edge Components에서 사용 가능한 일부 기능을 강조하는 자세한 설명이 있습니다. Application Server Caching Proxy 컴포넌트에 대한 개요는 3 페이지의 제 1 장 『WebSphere Application Server Edge Components 소개』를 참조하십시오.

다음 장으로 구성되어 있습니다.

15 페이지의 제 4 장 『캐싱』

23 페이지의 제 5 장 『네트워크 성능』

27 페이지의 제 6 장 『가용성』

37 페이지의 제 7 장 『Content Based Routing』

제 4 장 캐싱

Caching Proxy의 캐싱 기능은 네트워크 대역폭 이용을 최소화하고 일반 사용자가 보다 빨리 신뢰성 있는 서비스를 받을 수 있게 합니다. 이렇게 할 수 있는 이유는 프록시 서버에서 수행되는 캐싱이 백엔드 서버 및 피어 연결을 오프로드하기 때문입니다. Caching Proxy는 정적 콘텐츠 및 WebSphere Application Server에서 동적으로 생성된 콘텐츠를 캐시할 수 있습니다. 확장된 캐싱을 제공하기 위해 Caching Proxy는 Application Server Load Balancer 컴포넌트와 함께 작동합니다. 이 시스템 소개에 대한 자세한 정보는 3 페이지의 제 1 장 『WebSphere Application Server Edge Components 소개』를 참조하십시오.

중요사항: Caching Proxy는 다음 경우를 제외하고 모든 Edge Components 설치에 사용할 수 있습니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행되는 Edge Components 설치에 사용할 수 없습니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge Components 설치에 사용할 수 없습니다.

기본 Caching Proxy 구성

Caching Proxy는 역방향 Caching Proxy 서버(기본 구성) 또는 전방향 Caching Proxy 서버의 역할로 정의될 수 있습니다. 콘텐츠 호스트에서 사용되는 경우 Caching Proxy는 인터넷과 엔터프라이즈 콘텐츠 호스트 사이에 있는 역방향 Caching Proxy 서버의 역할에 구성됩니다. 인터넷 액세스 프로바이더에서 사용되는 경우 Caching Proxy는 클라이언트와 인터넷 사이에 있는 전방향 Caching Proxy 서버의 역할에 구성됩니다.

역방향 Caching Proxy(기본 구성)

역방향 프록시 구성을 사용하면 Caching Proxy 시스템은 인터넷 및 엔터프라이즈의 콘텐츠 호스트 사이에 위치합니다. 프록시 서버는 대리자로 작동하면서, 인터넷에서 도착한 사용자 요청을 인터셉트한 다음 적절한 콘텐츠 호스트로 요청을 전달한 후 리턴된 데이터를 캐시하여 해당 데이터를 인터넷을 통해 사용자에게 전달합니다. 캐싱을 사용하면으로써 Caching Proxy는 동일한 콘텐츠에 대한 후속 요청을 캐시로부터 곧바로 만족시킬 수 있는데 이는 콘텐츠 서버에서 다시 콘텐츠를 검색하는 것보다 훨씬 빠릅니다. 캐시될 수 있는 정보는 언제 만기되는지, 캐시 크기가 어느 정도인지 및 언제 정보를 갱신할 것인지에 따라 다릅니다. 캐시 히트에 대한 다운로드 시간이 짧을수록 고객에 대한 서비스 품질이 우수하다는 것을 의미합니다. 16 페이지의 그림 1은 기본 Caching Proxy 기능을 나타냅니다.

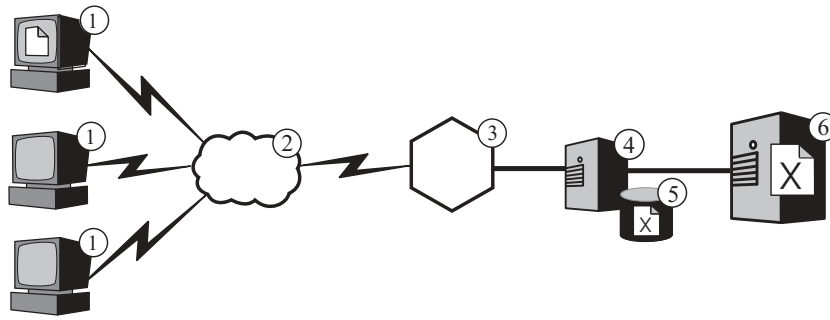


그림 1. 역방향 프록시로 작동하는 Caching Proxy. 범례: 1—클라이언트 2—인터넷 3—라우터/게이트웨이 4—Caching Proxy 5—캐시 6—컨텐츠 호스트

이 구성에서 프록시 서버(4)는 URL에 컨텐츠 호스트의 호스트 이름(6)이 포함되어 있는 요청을 차단합니다. 클라이언트(1)가 파일 X를 요청하면 요청은 인터넷(2)을 거쳐 인터넷 게이트웨이(3)를 통해 엔터프라이즈의 내부 네트워크로 들어갑니다. 프록시 서버는 요청을 차단하고 원래 주소로서 고유한 IP 주소가 있는 요청을 새로 생성하고 새 요청을 컨텐츠 호스트(6)로 전송합니다.

컨텐츠 호스트는 파일 X를 일반 사용자에게 직접 보내지 않고 프록시 서버로 리턴합니다. 파일 캐시가 가능하면 Caching Proxy는 일반 사용자에게 전달하기 전에 캐시(5)에 사본을 저장합니다. 캐시 가능 컨텐츠에 대해 가장 적합한 예제는 정적 웹 페이지입니다. 그러나 Caching Proxy는 WebSphere Application Server에서 동적으로 생성된 컨텐츠를 캐시 및 제공할 수 있는 기능을 제공합니다.

전방향 Caching Proxy

일반 사용자에게 직접 인터넷 액세스를 제공하는 것은 비효율적일 수 있습니다. 웹 서버에서 주어진 파일을 폐치하는 모든 사용자는 파일이 변경되지 않는 경우에도 파일을 폐치하는 첫 번째 사용자와 같이 네트워크와 인터넷 게이트웨이에서 같은 양의 통신량을 발생시킵니다. 해결 방법은 게이트웨이 근처에서 전방향 Caching Proxy를 설치하는 것입니다.

전방향 프록시 구성을 사용하면 Caching Proxy 시스템은 클라이언트와 인터넷 사이에 위치합니다. Caching Proxy는 클라이언트의 요청을 인터넷에 있는 컨텐츠 호스트에 전달하며 검색된 데이터를 캐시하고 검색된 데이터를 클라이언트에 전달합니다.

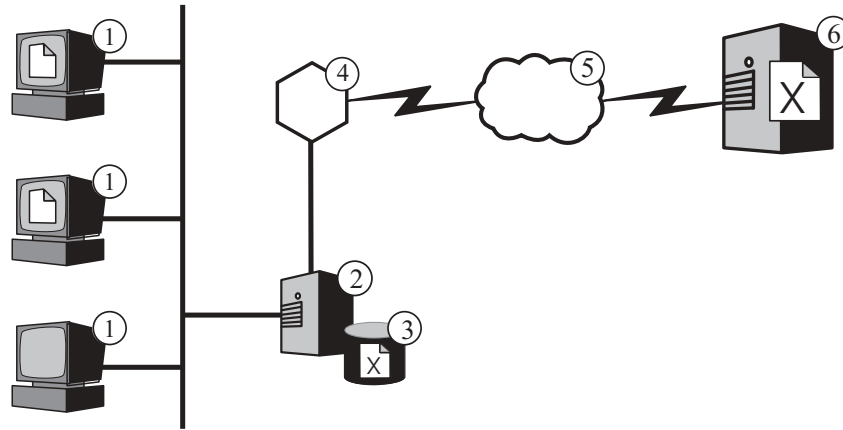


그림 2. 전방향 프록시로 작동하는 Caching Proxy. 범례: 1—클라이언트 2—Caching Proxy 3—캐시 4—라우터/게이트웨이 5—인터넷 6—컨텐츠 호스트

그림 2는 전방향 Caching Proxy 구성을 나타냅니다. 이 클라이언트 브라우저 프로그램 (1로 표시된 시스템)은 요청을 인터셉트하도록 구성된 전방향 Caching Proxy(2)에 요청을 직접 전달하도록 구성됩니다. 일반 사용자가 컨텐츠 호스트(6)에 저장된 X 파일을 요청하는 경우 전방향 Caching Proxy는 요청을 인터셉트하고 원래 주소로서 고유한 IP 주소가 있는 요청을 새로 생성하고 새 요청을 인터넷(5)을 통해 엔터프라이즈 라우터 (4)의 수단으로 전송합니다.

이 방법에서 기점 서버는 X 파일을 일반 사용자에게 직접 보내지 않고 전방향 Caching Proxy로 리턴합니다. 전방향 Caching Proxy의 캐싱 기능을 사용하면 Caching Proxy는 만기 날짜 및 파일이 동적으로 생성되는지에 대한 인디케이션과 같은 리턴 헤더의 설정을 점검하여 X 파일이 캐시에 적합한지를 식별합니다. 파일 캐시가 가능하면 Caching Proxy는 일반 사용자에게 전달하기 전에 캐시(3)에 사본을 저장합니다. 기본적으로 캐시가 사용 가능하고 전방향 Caching Proxy가 메모리 캐시를 사용합니다. 그러나 다른 유형의 캐싱을 구성할 수 있습니다.

X 파일에 대한 첫 번째 요청의 경우 전방향 Caching Proxy는 인터넷 액세스의 효율성을 많이 개선하지는 않습니다. 실제 X 파일을 액세스하는 첫 번째 사용자의 응답 시간은 전방향 Caching Proxy를 사용하지 않을 때보다 더 느릴 수도 있습니다. 이것은 전방향 Caching Proxy가 원래 요청 패킷을 처리하고 수신할 때 캐시 가능 정보에 대한 X 파일의 헤더를 검사하는 데 더 많은 시간을 사용하기 때문입니다. 전방향 Caching Proxy를 사용하면 다른 사용자가 연속으로 X 파일을 요청하는 경우 유용합니다. 전방향 Caching Proxy는 X 파일의 캐시된 사본이 여전히 유효한지를(만기되지 않음) 점검하고 유효한 경우 인터넷을 통해 요청을 컨텐츠 호스트로 전달하지 않고 캐시에서 X 파일을 직접 받습니다.

전방향 Caching Proxy가 요청 파일이 만기되었음을 발견한 경우에도 컨텐츠 호스트에서 파일을 다시 페치할 필요가 없습니다. 대신 전방향 Caching Proxy는 특별한 상태

점점 메시지를 콘텐츠 호스트로 보냅니다. 콘텐츠 호스트가 파일이 변경되지 않았음을 표시하는 경우 전방향 Caching Proxy는 캐시된 버전을 요청 사용자에게 전달할 수 있습니다.

이 방법에서 전방향 Caching Proxy의 구성은 Caching Proxy가 브라우저의 측면에서 활동하며 인터넷을 통해 요청을 콘텐츠 호스트로 전달하기 때문에 전방향 프록시라고 불립니다. 캐시가 있는 전방향 프록시의 이점은 다음 두 가지로 볼 수 있습니다.

- 파일이 캐시되면 전방향 Caching Proxy가 로컬 네트워크에 있게 되므로 일반 사용자는 요청이 인터넷을 경유해야 되는 경우보다 더 빨리 파일을 받을 수 있습니다. 더 많은 파일이 캐시되면 인터넷 요청에 대해 사용자가 경험하게 되는 전체 응답 시간은 점점 더 느려집니다.
- 엔터프라이즈 로컬 네트워크 외부에서 생성되는 통신량이 없습니다. 이것은 캐시되지 않은 파일에 대한 요청을 핸들링하기 위해 엔터프라이즈의 게이트웨이를 사용 가능하게 하여 인터넷에 대한 엔터프라이즈 게이트웨이의 용량을 효율적으로 증가시킵니다. 또한 비용이 몇 개의 패킷에 기반하는 환경에서 중요한 역할을 하는 인터넷 액세스 비용을 줄입니다.

Caching Proxy는 HTTP(Hypertext Transfer Protocol), FTP(File Transfer Protocol) 및 Gopher와 같은 일부 네트워크 전송 프로토콜을 프록시할 수 있습니다.

투명 전방향 Caching Proxy(Linux 시스템 전용)

전방향 Caching Proxy의 변형은 투명 Caching Proxy입니다. 이 역할에서 Caching Proxy는 기본 전방향 Caching Proxy와 같은 기능을 수행합니다. 그러나 클라이언트의 존재를 인식하지 않고 수행합니다. 투명 Caching Proxy 구성은 Linux 시스템에서만 지원됩니다.

이 구성은 16 페이지의 『전방향 Caching Proxy』에서 설명되며 각 클라이언트 브라우저는 특정 전방향 Caching Proxy에 요청을 직접 전달하도록 개별적으로 구성됩니다. 이러한 구성을 유지보수하는 것은 불편한 작업이 될 수 있습니다(특히 많은 수의 클라이언트 시스템이 있는 경우). Caching Proxy는 관리를 간편하게 하는 몇 가지 대체 방법을 지원합니다. 그 중 한 방법은 19 페이지의 그림 3에서 설명하는 대로 투명 프록시에 대해 Caching Proxy를 구성하는 것입니다. 일반 전방향 Caching Proxy에서는 투명 Caching Proxy가 게이트웨이 근처의 시스템에 설치되지만 클라이언트 브라우저 프로그램은 전방향 Caching Proxy에 요청을 직접 전달하도록 구성되지 않습니다. 클라이언트는 구성에 있는 프록시를 인식하지 않습니다. 대신 라우터가 클라이언트 요청을 인터셉트하고 이 요청을 투명 Caching Proxy에 직접 전달하도록 구성됩니다. 1로 표시된 시스템 중 하나에서 작업 중인 클라이언트가 콘텐츠 호스트(6)에 저장된 X 파일을 요청하는 경우 라우터(2)는 해당 요청을 Caching Proxy로 전달합니다. Caching Proxy는 원래 주소로서 고유한 IP 주소가 있는 요청을 새로 생성하고 새 요청을 인터넷(5)을 통해 라우터(2)의 수단으로 전송합니다. X 파일이 도착하면 Caching Proxy는

적절한 경우(16 페이지의 『전방향 Caching Proxy』에 설명된 조건 참조) 파일을 캐시하고 요청 클라이언트에게 이 파일을 전달합니다.

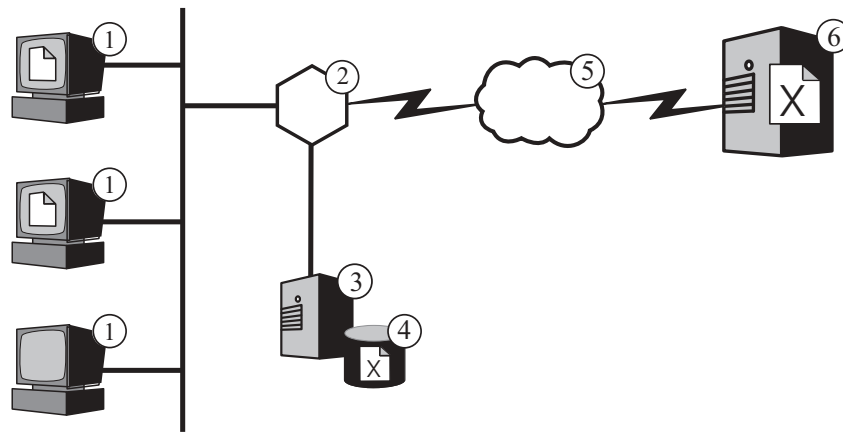


그림 3. 투명 전방향 프록시로 작동하는 Caching Proxy. 범례: 1—클라이언트 2—라우터/게이트웨이 3—Caching Proxy 4—캐시 5—인터넷 6—컨텐츠 호스트

HTTP 요청의 경우, 각 브라우저에서 프록시 구성 정보를 유지하는 다른 대체 방법은 Netscape Navigator 버전 2.0 이상 및 Microsoft Internet Explorer 버전 4.0 이상과 같은 일부 브라우저 프로그램에서 사용 가능한 자동 프록시 구성 기능을 사용하는 것입니다. 이러한 경우 하나 이상의 중앙 PAC(Proxy Automatic Configuration) 파일을 작성하고 로컬 프록시 구성 정보 대신 이 파일 중 하나를 참조하여 브라우저를 구성합니다. 브라우저는 PAC의 변경사항을 자동으로 인식하며 프록시 사용을 적절하게 조정합니다. 이 방법을 사용하여 각 브라우저의 개별 구성 정보를 유지보수할 필요가 없으며 프록시 서버를 사용할 수 없을 때 요청을 다시 라우팅하기가 쉽습니다.

세 번째 대체 방법은 Internet Explorer 버전 5.0 이상과 같은 일부 브라우저 프로그램에서 사용 가능한 WPAD(Web Proxy Auto Discovery) 메커니즘을 사용하는 것입니다. 브라우저에서 이 기능을 사용하면 네트워크에서 WPAD 호환 프록시 서버를 찾아서 웹 요청을 직접 전달합니다. 이 경우 중앙 프록시 구성 파일을 유지보수할 필요가 없습니다. Caching Proxy는 WPAD 호환 가능합니다.

고급 캐싱

로드 밸런스된 Caching Proxy 클러스터

보다 고급인 캐싱 기능을 제공하려면 Load Balancer 컴포넌트와의 결합에서 Caching Proxy를 역방향 프록시로 사용하십시오. 캐시 및 로드 밸런스 성능을 통합하여 효율적이고 관리하기 쉬운 웹 성능 하부 구조를 작성할 수 있습니다.

20 페이지의 그림 4는 요청이 많을 때에도 웹 콘텐츠를 효과적으로 전달할 수 있도록 Caching Proxy와 Load Balancer를 결합하는 방법을 나타냅니다. 이 구성에서 프록시

서버(4)는 Load Balancer(6)가 로드 밸런싱할 콘텐츠 호스트 클러스터(7)의 호스트 이름이 URL에 포함되어 있는 요청을 차단하도록 구성됩니다.

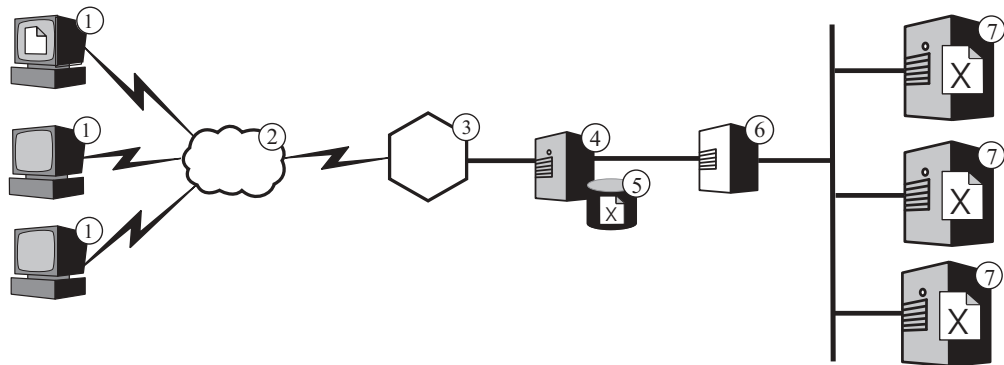


그림 4. 로드 밸런싱된 클러스터에 대해 프록시 서버의 역할을 하는 Caching Proxy. 범례:

1—클라이언트 2—인터넷 3—라우터/게이트웨이 4—Caching Proxy 5—캐시
6—Load Balancer 7—콘텐츠 호스트

클라이언트(1)가 파일 X를 요청하면 요청은 인터넷(2)을 거쳐 인터넷 게이트웨이(3)를 통해 엔터프라이즈의 내부 네트워크로 들어갑니다. 프록시 서버는 요청을 인터셉트하여 기점 주소로서 고유 IP 주소를 갖는 새 요청을 생성한 후 새 요청을 클러스터 주소에 있는 Load Balancer로 전송합니다. Load Balancer는 로드 밸런스 알고리즘을 사용하여 파일 X에 대한 요청을 가장 만족시킬 수 있는 콘텐츠 호스트를 판별합니다. 해당 콘텐츠 호스트는 파일 X를 Load Balancer를 통하지 않고 프록시 서버로 전송합니다. 프록시 서버는 앞에서 설명한 것과 같은 방식으로 캐시 여부를 결정하여 일반 사용자에게 전달합니다.

동적 콘텐츠 캐싱

Caching Proxy의 동적 캐싱 플러그인에서도 고급 캐싱 기능을 제공합니다. WebSphere Application Server와 함께 사용할 경우 Caching Proxy는 JSP(JavaServer Pages) 양식의 동적 콘텐츠 또는 WebSphere Application Server에서 생성된 Servlet 응답을 캐시, 서브, 무효화할 수 있는 기능이 있습니다.

일반적으로 표준 시간 기본 캐시 만기 논리로는 적시에 제거할 수 없기 때문에, 만기 시간이 없는 동적 콘텐츠는 "캐시하지 않음"으로 표시되어야 합니다. 동적 캐시 플러그인의 이벤트 위주 만기 논리를 사용하면 프록시 서버가 만기 시간이 없는 콘텐츠를 캐시할 수 있습니다. 네트워크 경계에 이러한 콘텐츠를 캐시함으로써 콘텐츠 호스트가 클라이언트로부터의 요청을 충족시키기 위해 Application Server를 반복적으로 호출하는 것을 줄일 수 있습니다. 이는 다음과 같은 이점도 제공합니다.

- 웹 서버, WebSphere Application Server 및 백엔드 콘텐츠 호스트에서의 워크로드를 줄입니다.
- 네트워크 지연을 제거하여 사용자에게 보다 신속하게 응답합니다.

- 인터넷 순회가 줄어 대역폭의 사용을 감소시킵니다.
- 동적 생성 콘텐츠를 제공하는 웹 사이트의 확장성이 보다 향상됩니다,

Servlet 응답 캐시는 응용프로그램 논리 또는 이벤트(예: 데이터베이스로부터의 메시지)를 기본으로 만기되는 동적 생성 웹 페이지에 이상적입니다. 이러한 페이지의 지속 시간은 한정되어 있으나 만기 트리거를 미리 알 수 없기 때문에 작성 시 지속 시간 값을 설정할 수 없습니다. 이러한 페이지의 지속 시간을 0으로 설정하면 콘텐츠 호스트가 동적 콘텐츠를 제공할 때 어려움이 많습니다.

Caching Proxy 및 Application Server의 동적 캐시를 동기화하는 책임은 두 시스템에서 공유합니다. 예를 들어, 현재 일기 예보를 제공하는 응용프로그램에서 동적으로 작성한 공용 웹 페이지를 Application Server가 내보내서 Caching Proxy가 캐시할 수 있습니다. Caching Proxy는 페이지가 유효하지 않다는 통지를 받을 때까지 반복해서 응용프로그램 실행 결과를 여러 사용자에게 제공할 수 있습니다. 캐시가 가득 차 있어서 프록시 서버가 항목을 제거하고 Caching Proxy 구성 파일의 ExternalCacheManager 지시문에 설정된 기본 시간 종료가 만기되거나 Caching Proxy가 캐시에서 콘텐츠를 제거하라고 지시하는 무효화 메시지를 받을 때까지 Caching Proxy Servlet 응답 캐시의 콘텐츠는 유효합니다. 무효화 메시지는 콘텐츠를 소유하는 WebSphere Application Server에서 발생하여 구성된 각 Caching Proxy로 전달됩니다.

주: 동적으로 생성된 개인용 페이지(예: 사용자의 장바구니 콘텐츠를 보여주는 페이지)는 일반적으로 Caching Proxy에 의해 캐시될 수 없으며 캐시되어서도 안됩니다. Caching Proxy는 개인용 페이지를 의도한 사용자에게만 제공하기 위해 인증 및 권한 부여를 수행하도록 구성된 경우에만 개인용 페이지를 캐시하여 제공할 수 있습니다.

추가 캐싱 기능

Caching Proxy는 다른 핵심 고급 캐싱 기능도 제공합니다.

- 초대형 캐시 사용 기능
- 가장 많이 액세스된 페이지로 캐시를 새로 고칠 수 있는 옵션
- 헤더 정보에 매번 폐치하라고 되어 있는 페이지를 캐시할 수 있는 기능
- 서버 성능 향상 및 캐시 유지보수를 위해 매일 가비지 콜렉션 구성 가능
- 여러 Caching Proxy 시스템이 동일한 캐시를 공유하여 캐시된 콘텐츠의 중복을 줄일 수 있게 하는 RCA(Remote Cache Access)
- Caching Proxy가 HTML 페이지 및 기타 캐시 가능 자원 검색 시 기타 ICP(Internet Caching Protocol) 준수 캐시를 조회할 수 있게 하는 ICP 플러그인

제 5 장 네트워크 성능

Caching Proxy 기능을 도입하면 네트워크 성능에 영향을 줄 수 있습니다. 네트워크의 성능을 향상시키려면 Caching Proxy를 단독으로 사용하거나 Load Balancer와 함께 사용하십시오. 이 시스템의 소개에 대한 자세한 정보는 3 페이지의 제 1 장 『WebSphere Application Server Edge Components 소개』를 참조하십시오.

엔터프라이즈 내에서의 Caching Proxy의 성능은 Caching Proxy가 실행 중인 하드웨어 및 Caching Proxy를 도입하는 시스템의 구조 수준에 지나지 않습니다. 네트워크 성능을 향상시키려면 하드웨어 및 전반적인 네트워크 구조를 프록시 서버의 특성에 맞게 모델링하십시오.

Caching Proxy 소프트웨어의 기본 구성 및 관리와 운영 체제 레벨에서의 조정도 Caching Proxy의 성능에 지대한 영향을 미칩니다. 소프트웨어 구성을 많이 변경하면 성능을 향상시킬 수 있습니다. 여기에는 로깅 지시문, 맵핑 규칙, 플러그인, 시간 종료 값, 캐시 구성 값, 활성 스레드 값이 포함되나 이것으로 제한되지는 않습니다. Caching Proxy 소프트웨어 구성에 대한 세부사항은 *Caching Proxy 관리 안내서*에 나와 있습니다.

대부분의 운영 체제 구성 변경은 성능을 향상시킬 수도 있습니다. 여기에는 TCP 및 ARP 조정, 파일 설명자 제한 증가, 시스템 시계 동기화, NIC(Network Interface Card) 조정, 시스템 관리 작업을 수행할 경우 다음과 같은 공통적인 유익한 활용이 포함되나 이것으로 제한되지는 않습니다.

중요사항: Caching Proxy는 다음 경우를 제외하고 모든 Edge Components 설치에 사용할 수 있습니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행되는 Edge Components 설치에 사용할 수 없습니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge Components 설치에 사용할 수 없습니다.

네트워크 하드웨어

이 절에서는 Caching Proxy 기능을 네트워크에 도입할 때 고려해야 할 네트워크 하드웨어 문제점에 대해 설명합니다.

메모리 고려사항

대량의 프록시 서버 전용 메모리가 필요합니다. 대량의 메모리 전용 캐시를 구성할 경우 Caching Proxy는 2GB의 가상 주소 공간을 사용할 수 있습니다. 커널, 공유 라이

브러리 및 네트워크 버퍼용 메모리도 필요합니다. 그러므로 프록시 서버가 3GB 또는 4GB의 실제 메모리를 사용하게 될 수 있습니다. 메모리 전용 캐시는 원시 디스크 캐시보다 훨씬 빠르며 이 구성 변경사항만으로도 성능 향상을 느낄 수 있습니다.

하드 디스크 고려사항

중요한 점은 Caching Proxy가 설치되어 있는 시스템에 대량의 디스크 공간이 있어야 한다는 점입니다. 이는 디스크 캐시를 사용할 경우에 특히 중요합니다. 하드 디스크를 읽고 기록하는 것은 컴퓨터의 주요 프로세스입니다. Caching Proxy의 입출력 프로시저가 효율적이라 할지라도 디스크 캐시를 사용하도록 Caching Proxy를 구성하면 하드 드라이브의 시스템적인 제한으로 인해 성능이 제한될 수 밖에 없습니다. 디스크 입출력 병목 현상은 공 캐시 장치 및 로그 파일에 여러 개의 하드 드라이브를 사용하고 찾기 시간, 회전 속도 및 전송 비율이 빠른 디스크 드라이브를 사용함으로써 완화할 수 있습니다.

네트워크 고려사항

속도, 유형 및 NIC 수, 프록시 서버에 대한 네트워크 연결 속도와 같은 네트워크 요구사항은 Caching Proxy의 성능에 영향을 미칩니다. 프록시 서버 시스템에서 두 개의 NIC(통신 수신용과 통신 전송용으로 하나씩)를 사용하기 위해서는 일반적으로 성능이 가장 중요합니다. 단일 NIC가 HTTP 요청 및 응답 통신만으로 최대 한계에 도달할 가능성이 있습니다. 더욱이 NIC는 최소 100MB여야 하며 항상 전이중 조작용으로 구성해야 합니다. 이유는 경로 지정 및 전환 장비 간의 자동 중재 시 오류가 발생하여 처리량이 감소할 수 있기 때문입니다. 마지막으로 네트워크 연결 속도도 매우 중요합니다. 예를 들어, Caching Proxy 시스템에 대한 연결이 포화 상태의 T1 반송자이면 많은 요청 로드와 최적의 처리량을 처리할 수 있다고 기대할 수 없습니다.

CPU 고려사항

Caching Proxy 시스템의 CPU(Central Processing Unit)는 한계 요소가 될 가능성이 있습니다. CPU 능력은 요청을 처리하는 데 걸리는 시간에 영향을 미치며 네트워크 내의 CPU의 수는 확장성에 영향을 미칩니다. 특히 프록시 서버가 처리할 최대 요청 로드를 모델화하려면 프록시 서버의 CPU 요구사항을 환경에 일치시키는 것이 중요합니다.

네트워크 구조

전반적인 성능면에서 일반적으로 단지 하드웨어를 개별적으로 추가하는 것보다는 구조를 확장하는 것이 유리합니다. 단일 시스템에 하드웨어를 아무리 많이 추가해도 해당 하드웨어는 여전히 최대 성능 레벨을 갖고 있습니다.

이 절에서는 Caching Proxy 기능을 네트워크에 추가할 때 고려해야 할 네트워크 구조 문제점에 대해 설명합니다.

웹 사이트 관심 및 프록시 서버 로드 고려사항

기업의 웹 사이트가 인기가 있으면, 콘텐츠에 대해 단일 프록시 서버가 효율적으로 만족시킬 수 있는 것보다 더 큰 수요가 생길 수 있으며 이로 인해 응답 시간이 느려질 수 있습니다. 네트워크 성능을 최적화하려면 클러스터되고 로드 밸런스가 이루어진 Caching Proxy 시스템을 포함시키거나 전체 네트워크 구조에서 RCA와 공유하는 캐시를 사용하는 것을 고려하십시오.

- 로드 밸런스 클러스터

구조를 확장하는 한 가지 방법은 프록시 서버를 클러스터하고 Load Balancer 컴포넌트를 사용하여 클러스터 간의 로드 밸런스를 유지하는 것입니다. 프록시 서버를 클러스터하는 것은 성능 및 확장성 뿐만 아니라 중복 및 신뢰성을 위해서도 유리한 설계 고려사항입니다. 단일 프록시 서버는 장애가 발생할 수 있는 유일한 지점임을 나타냅니다. 즉, 고장이 나거나 네트워크 장애로 인해 액세스가 불가능하게 되면 사용자는 웹 사이트에 액세스할 수 없습니다.

- 공유 캐시 구조

RCA와의 공유 캐시 구조도 고려하십시오. 공유 캐시 구조를 사용하면 일반적으로 ICP(Internet Cache Protocol) 또는 CARP(Cache Array Routing Protocol)와 같은 내부 캐시 프로토콜을 사용하는 여러 Caching Proxy 서버로 총 가상 캐시를 확장합니다. RCA는 대형 가상 캐시를 제공함으로써 클러스터된 캐시 히트 비율을 최대화하도록 설계되었습니다.

프록시 서버 RCA 배열(단일 독립형 Caching Proxy 또는 독립형 Caching Proxy 시스템 클러스터와 대조됨)을 사용하면 성능이 향상됩니다. 대부분의 경우, 캐시 히트 비율을 최대화하고 캐시 불일치 및 지연을 최소화하는 총 가상 캐시 크기를 늘리면 성능이 향상됩니다. RCA에서는 특정 문서의 단 하나의 사본만이 캐시에 상주합니다. 프록시 서버 클러스터에서는 총 캐시 크기는 늘어나지만 여러 프록시 서버가 동일한 정보를 폐치하여 캐시합니다. 그러므로 총 캐시 히트 비율은 증가하지 않습니다.

RCA는 일반적으로 대형 엔터프라이즈의 콘텐츠 호스팅 시나리오에 사용됩니다. 그러나 대형 엔터프라이즈 전개 시에만 RCA가 유용한 것은 아닙니다. 네트워크 로드에서 캐시 서버 클러스터가 필요하고 대부분의 요청이 캐시 히트인 경우 RCA 사용을 고려하십시오. 네트워크 설정에 따라 RCA를 구성할 때 클라이언트가 사용하는 TCP 연결 수가 증가하므로 항상 엔터프라이즈 성능이 개선되는 것은 아닙니다. 이유는 RCA 구성원이 가장 점수가 높은 URL을 제공할 뿐 아니라 점수가 높지 않은 URL에 대한 요청을 받은 경우 요청을 다른 구성원에게도 전달해야 하기 때문입니다. 이는 독립형 서버로서 작동할 경우 모든 제공된 RCA 배열 구성원이 자신이 열 수 있는 것보다 더 많은 TCP 연결을 열 수 있음을 의미합니다.

통신량 유형 고려사항

향상된 성능에 크게 기여하는 것은 Caching Proxy의 캐시 성능입니다. 그러나 제대로 구성하지 않을 경우, 프록시 서버의 캐시에는 병목 현상이 생길 수 있습니다. 가장 좋은 캐시 구성을 판별하기 위해서는 통신량 특성을 분석하는 데 상당한 노력을 기울여야 합니다. 콘텐츠의 유형, 크기, 양 및 속성이 기점 서버에서 문서를 검색하고 서버에 로드하는 데 걸리는 시간 측면에서 프록시 서버의 성능에 영향을 미칩니다. Caching Proxy가 프록시하거나 캐시로부터 제공할 통신 유형을 이해하면 프록시 서버를 구성할 때 이러한 특성을 참작할 수 있습니다. 예를 들어, 캐시되는 오브젝트의 80%가 이미지(*.gif 또는 *.jpg)이고 크기가 200KB 정도라는 사실을 안다면 캐시 매개변수를 조정하고 캐시의 크기를 판별하는 데 큰 도움이 될 수 있습니다. 또한 대부분의 콘텐츠가 캐시 후 보가 아닌 개인화된 동적 페이지라는 사실을 이해하는 것도 Caching Proxy를 조정하는 데 도움이 됩니다.

통신량 특성을 분석하여 메모리 캐시 또는 디스크 캐시 중 어느 것이 캐시 성능을 최적화할 수 있는지 판별할 수 있습니다. 또한 네트워크 통신량 특성을 잘 알면 Caching Proxy의 동적 캐시 기능을 사용할 경우 성능이 향상될 수 있는지 여부를 판별할 수 있습니다.

• 메모리 캐시 대 디스크 캐시

디스크 캐시는 대량의 정보를 캐시해야 하는 사이트에 적합합니다. 예를 들어, 사이트의 콘텐츠가 많고(5GB 이상) 캐시 히트 비율이 80 - 90%이면 디스크 캐시가 적합합니다. 그러나 메모리(RAM) 캐시가 더 빠르다고 알려져 있으므로 대형 사이트에 메모리 전용 캐시를 사용하는 것이 적당한 경우 여러 가지 시나리오가 있습니다. 예를 들어, Caching Proxy의 캐시 히트 비율이 중요하지 않거나 공유 캐시 구성을 사용할 경우에는 메모리 캐시가 효과적입니다.

• 동적 생성 콘텐츠 캐시

Application Server 캐시를 네트워크 기본 캐시로 가상 확장할 경우, Caching Proxy는 WebSphere Application Server 동적 캐시에서 생성된 동적 콘텐츠(JSP 및 Servlet 결과)를 캐시 및 무효화할 수 있습니다. 동적 생성 콘텐츠의 캐시를 사용 가능하게 하면 응용프로그램의 논리 또는 이벤트(예: 데이터베이스로부터의 메시지)를 기본으로 만기되는 동적 생성 공용 웹 페이지에 대한 요청이 많은 환경에서 네트워크 성능이 향상됩니다. 페이지 지속 시간은 한정되어 있으나 페이지 작성 시 만기 트리거는 설정할 수 없습니다. 따라서 동적 캐시 및 무효화 기능이 없는 호스트는 지속 시간이 0인 페이지를 가리켜야 합니다.

지속 시간 중 두 명 이상의 사용자가 동적 생성 페이지를 한 번 이상 요청할 경우 동적 캐시는 유용한 오프로드를 제공하여 네트워크 콘텐츠 호스트에서의 워크로드를 줄입니다. 동적 캐시를 사용하면 네트워크 지연을 제거하고 적은 인터넷 통과 횟수로 대역폭 사용을 줄여 사용자에게 보다 빠른 응답을 제공함으로써 네트워크 성능이 향상됩니다.

제 6 장 가용성

Application Server Load Balancer 컴포넌트는 WebSphere Application Server와 같은 콘텐츠 서버 또는 Application Server Caching Proxy 컴포넌트와 함께 작동하여 네트워크의 가용성 및 확장성을 향상시킬 수 있습니다. (이 Edge Components의 개요에 대해서는 3 페이지의 제 1 장 『WebSphere Application Server Edge Components 소개』를 참조하십시오.) Load Balancer는 엔터프라이즈 네트워크에서 사용되며 인터넷과 엔터프라이즈의 백엔드 서버 사이에 설치됩니다. Load Balancer는 요청이나 콘텐츠가 많아서 엔터프라이즈가 백엔드 서버를 여러 개 사용하는 경우에도 인터넷에서 엔터프라이즈의 단일 연결 지점으로 사용됩니다.

가용성은 로드 밸런스 및 실패복구 지원을 통해 완성됩니다.

중요사항: Caching Proxy는 다음 경우를 제외하고 모든 Edge Components 설치에 사용할 수 있습니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행되는 Edge Components 설치에 사용할 수 없습니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge Components 설치에 사용할 수 없습니다.

로드 밸런스

로드 밸런스는 프록시 서버 및 Application Server를 투명하게 클러스터하여 웹 사이트의 가용성 및 확장성을 향상시킵니다. 백엔드 처리 성능을 투명하게 추가할 수 있으므로 IT 하부 구조의 확장성은 현저히 향상됩니다.

다중 콘텐츠 호스트의 로드 밸런스

여러 호스트에 있는 콘텐츠를 복제하여 수 많은 요청을 충족시킬 수 있지만, 호스트 사이에 로드 밸런스를 맞출 수 있어야 합니다. DNS(Domain Name Service)가 기본 라운드 로빈 로드 밸런스를 제공할 수 있지만 실행되지 않는 상황도 있습니다.

여러 개의 콘텐츠 호스트를 로드 밸런스하기 위한 보다 근본적인 해결책은 28 페이지의 그림 5와 같이 Load Balancer Dispatcher 컴포넌트를 사용하는 것입니다. 이 구성에서 모든 콘텐츠 호스트(5로 표시된 시스템)는 동일한 콘텐츠를 저장합니다. 콘텐츠 호스트는 로드 밸런스 클러스터 양식으로 정의되어 있으며, Load Balancer 시스템(4)의 네트워크 인터페이스 중 하나에 클러스터 전용 호스트 이름 및 IP 주소가 할당되어 있습니다. 1로 표시된 시스템 중 하나에서 작업하는 일반 사용자가 파일 X를 요청하면 요청은 인터넷(2)을 거쳐 인터넷 게이트웨이(3)를 통해 기업 내부 네트워크로 들어갑니다. Dispatcher는 Dispatcher의 호스트 이름 및 IP 주소가 URL에 매핑되기 때문에 요

청을 인터셉트합니다. Dispatcher는 클러스터의 호스트 중에서 현재 요청을 가장 잘 처리할 수 있는 호스트를 판별하여 요청을 해당 호스트로 전달하고 MAC 전달 메소드가 구성되어 있으면 파일 X를 직접 클라이언트에게 리턴합니다(파일 X는 Load Balancer를 통해 전달되지 않음).

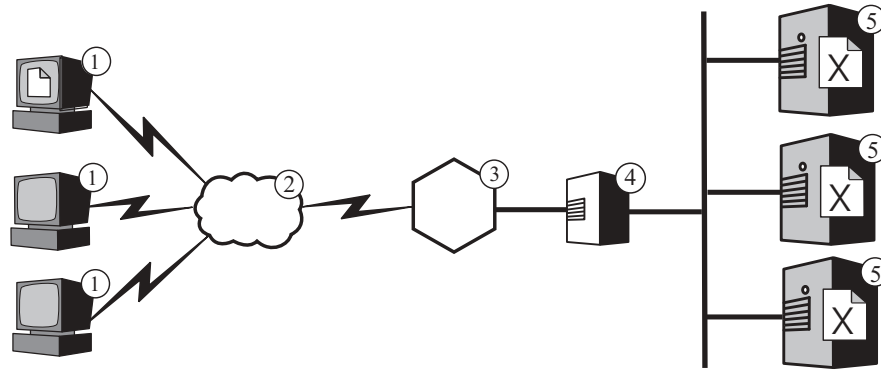


그림 5. 다중 콘텐츠 호스트의 로드 밸런스 범례: 1—클라이언트 2—인터넷 3—라우터/게이트웨이 4—Dispatcher 5—콘텐츠 호스트

주: Dispatcher는 세 가지 전달 메소드를 제공합니다.

- MAC 전달 메소드는 서버로 수신 요청을 로드 밸런싱하는 데 사용됩니다. 응답은 클라이언트로 직접 리턴됩니다.
- NAT/NAPT 전달 메소드는 원격에 위치한 서버에 사용됩니다. 수신 요청은 서버로부터 응답을 수신하여 클라이언트로 리턴하는 Dispatcher에 의해 로드 밸런싱됩니다. (IPv4 및 IPv6용 Load Balancer 설치 시, 이 전달 메소드는 지원되지 않습니다.)
- CBR(Content Based Routing) 메소드는 Caching Proxy를 사용하지 않고 HTTP 및 HTTPS 요청의 Content Based Routing을 제공합니다. HTTP에는 "콘텐츠" 유형 규칙을 사용하고 HTTPS에는 SSL 세션 ID 연관 관계를 사용하여 Content Based Routing을 수행합니다. (IPv4 및 IPv6용 Load Balancer 설치 시, 이 전달 메소드는 지원되지 않습니다.)

기본값으로 Dispatcher는 DNS와 같은 라운드 로빈 로드 밸런스를 사용하지만 DNS의 여러 단점을 보충해 줍니다. DNS와는 달리 콘텐츠 호스트 사용 여부, 액세스 가능 여부를 추적하여 클라이언트를 사용할 수 없는 서버로 연결시키지 않습니다. 또한 콘텐츠 호스트의 현재 로드 상태를 고려하여 연결이 새롭게 되었는지, 활성화되었는지, 종료되었는지를 추적합니다. 사용자는 Load Balancer의 선택적 어드바이저 및 관리자 컴포넌트를 활성화시켜서 로드 밸런스를 최적화할 수 있으며, 콘텐츠 호스트의 상태를 보다 정확히 추적하고 로드 밸런스 결정 프로세스내에 추가 정보를 포함시킵니다. 관리자를 사용하여 결정 프로세스에서 사용된 다른 요소에 각기 다른 책임을 할당할 수 있어서 사용자 사이트에 대한 로드 밸런스를 사용자 정의할 수 있습니다.

다중 역방향 프록시 서버의 로드 밸런스

Load Balancer의 Dispatcher는 여러 Caching Proxy 시스템에 대해서도 로드 밸런스를 수행할 수 있습니다. 기업의 웹 사이트가 인기가 있으면, 단일 프록시 서버가 효과적으로 만족할 수 있는 것보다 더 큰 수요가 생길 수 있으며 프록시 서버의 성능이 저하될 가능성도 있습니다.

단일 콘텐츠 호스트에 대해 프록시 기능을 수행하는 여러 개의 Caching Proxy를 사용할 수 있지만(16 페이지의 그림 1에 설명된 구성과 유사함) 사이트가 여러 개의 프록시 서버를 사용할 정도로 인기가 있으면 Load Balancer에 의해 로드 밸런스되는 여러 개의 콘텐츠 호스트가 필요할 수도 있습니다. 그림 6에 이 구성이 설명되어 있습니다. 4의 Dispatcher는 두 개의 프록시 서버(5) 클러스터를 로드 밸런스하며 7의 Dispatcher는 세 개의 콘텐츠 호스트(8) 클러스터를 로드 밸런스합니다.

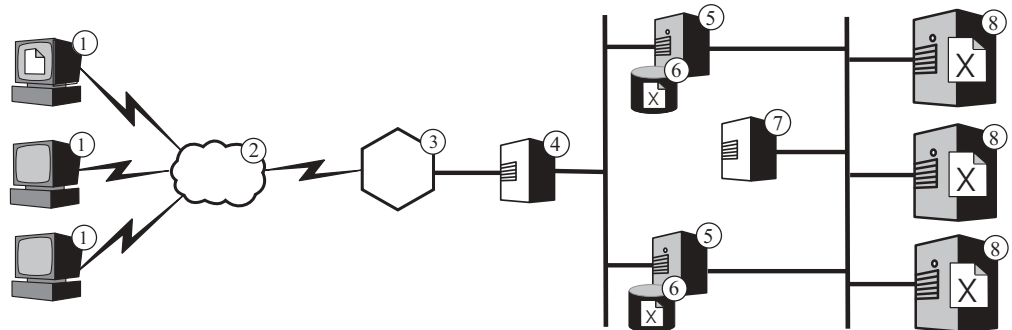


그림 6. 다중 역방향 프록시 서버 및 콘텐츠 호스트의 로드 밸런스 1—클라이언트 2—인터넷 3—라우터/게이트웨이 4—Dispatcher 5—Caching Proxy 6—캐시 7—Dispatcher 8—콘텐츠 호스트

4의 Dispatcher의 클러스터 호스트 이름은 기업 웹 콘텐츠의 URL 내로 표시되는 호스트 이름입니다(인터넷상에서 볼 수 있는 웹 사이트의 이름). 7의 Dispatcher 클러스터 호스트 이름은 인터넷상에서 보이지 않기 때문에 원하는 값은 무엇이든 넣을 수 있습니다. 예를 들면, ABC사의 경우 4에 해당하는 Dispatcher의 적절한 호스트 이름은 www.abc.com인 반면 7의 Dispatcher는 http-balancer.abc.com과 같은 식으로 부를 수 있습니다.

1의 클라이언트 시스템 중 하나에 있는 브라우저가 8의 콘텐츠 서버에 저장되어 있는 파일 X에 액세스해야 한다고 가정해 봅시다. HTTP 요청은 인터넷(2)을 통해 게이트웨이(3)에서 기업 내부 네트워크로 들어갑니다. 라우터는 요청을 4의 Dispatcher로 전송하는데, 로드 밸런스 알고리즘에 따라 현재 최상으로 처리가 가능한 프록시 서버(5)로 요청을 전달합니다. 프록시 서버가 캐시(6)에 파일 X를 가지고 있으면 4의 Dispatcher는 생략하고 브라우저로 직접 리턴합니다.

프록시 서버가 캐시 내에 파일 **X**의 사본을 가지고 있지 않으면, 헤더의 기점 필드 내에 고유 호스트 이름이 있는 새로운 요청을 작성하여 7의 Dispatcher로 전송합니다. Load Balancer 콘텐츠 호스트(8) 중 가장 적합한 것을 찾아서 요청을 전송합니다. 콘텐츠 호스트는 기억 영역에서 파일 **X**를 검색하고 7의 Dispatcher 생략하고 프록시 서버로 직접 파일을 리턴합니다. 그러면 프록시 서버는 파일 **X**를 캐시할 수 있는 경우 캐시하고 4의 Dispatcher를 생략하고 브라우저로 전송합니다.

다중 전방향 프록시 서버를 사용하는 Load Balancer

많은 수의 고객에서 인터넷 액세스를 제공하는 경우 고객들은 단일 프록시가 효율적으로 제공할 수 있는 양보다 더 많은 인터넷 액세스를 요구할 수 있습니다. Caching Proxy에서 요청이 과부하되면 고객은 직접 인터넷에 액세스할 때보다 응답 시간이 느려지는 것을 느끼게 됩니다. Caching Proxy가 실패하거나 네트워크 장애 때문에 액세스할 수 없게 되면 인터넷 액세스도 사용할 수 없습니다. 이 문제를 해결하려면 여러 개의 Caching Proxy 시스템을 설치하고 이 시스템 간의 로드를 밸런싱하는 Load Balancer Dispatcher를 사용하는 것입니다.

라우터가 두 개 이상의 Caching Proxy에 같은 유형의 트래픽을 라우팅할 수 있는 경우에 한해서(모든 라우터가 이 기능을 지원하지는 않음) Dispatcher 없이 여러 개의 Caching Proxy 시스템에 투명 프록시를 제공할 수 있습니다. Dispatcher 없이 여러 개의 Caching Proxy 시스템에 일반 전방향 프록시 서비스를 제공할 수는 있지만 Caching Proxy 시스템 중 하나를 기본 프록시로 사용하려면 클라이언트 브라우저를 명시적으로 구성해야 합니다. Caching Proxy가 실패하거나 과부화되거나 액세스할 수 없으면 일반 사용자는 인터넷에 액세스할 수 없게 됩니다. 이러한 상황을 방지하기 위해 하나 이상의 보조 Caching Proxy에 브라우저를 직접 전달하도록 PAC(Proxy Automatic Configuration) 파일(18 페이지의 『투명 전방향 Caching Proxy(Linux 시스템 전용)』의 설명 참조)을 작성할 수 있습니다. PAC 파일은 Caching Proxy 시스템 간의 로드를 밸런싱할 필요에 주목하지는 않지만 한 Caching Proxy가 다른 Caching Proxy보다 아주 많은 요청을 받는 경우 성능이 저하되기 쉬우며 브라우저의 고객은 응답 시간이 느려지는 것을 경험하게 됩니다. 비슷한 성능을 경험하는 모든 고객에 대해 각 Caching Proxy를 사용하도록 대략 같은 수의 브라우저를 구성해야 하며 브라우저를 추가하거나 제거할 때도 로드를 유지할 수 있도록 수동으로 분배를 추적해야 합니다.

31 페이지의 그림 7은 Dispatcher가 Caching Proxy 시스템의 클러스터를 로드 밸런싱하는 네트워크 구성에 대해 설명합니다. Dispatcher 시스템의 네트워크 인터페이스 중 하나는 클러스터의 전용 호스트 이름 및 IP 주소를 가지도록 구성됩니다. 클라이언트 브라우저는 인터넷 요청을 클러스터 호스트 이름으로 직접 전달하도록 구성됩니다. 예를 들어 1로 표시된 클라이언트 시스템 중 하나의 브라우저가 콘텐츠 호스트(7)에서 **X** 파일을 액세스해야 하는 경우, 해당 브라우저는 이 요청을 클러스터 호스트 이름 또는 주소로 전달합니다. 여기서 Dispatcher(2)가 요청을 인터셉트하고 적절한 Caching Proxy(3)로 전달합니다. Caching Proxy는 새 요청을 작성하고 이 요청을 엔터프라이즈 게이트

웨이(5)와 인터넷(6)을 통해 전달하며 적절한 경우 리턴된 파일을 캐시(4)에 저장합니다(자세한 정보는 16 페이지의 『전방향 Caching Proxy』 참조).

주: 투명 프록시 기능은 Linux 시스템에서만 사용할 수 있습니다.

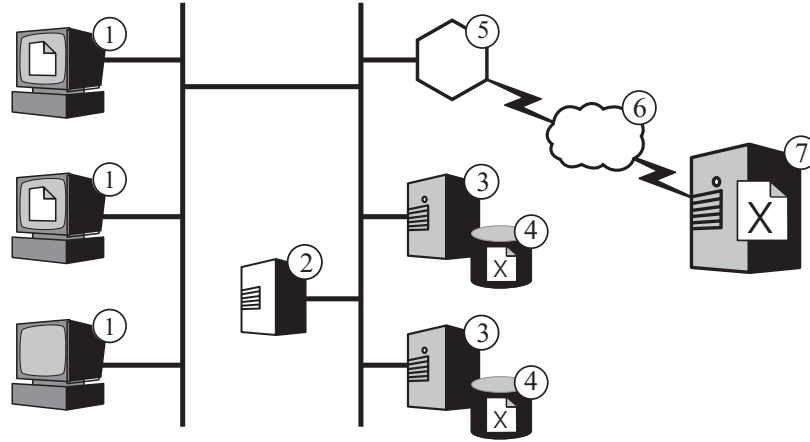


그림 7. Dispatcher를 사용하여 여러 개의 Caching Proxy 로드 밸런싱. 범례: 1—클라이언트 2—Dispatcher 3—Caching Proxy 4—캐시 5—라우터/게이트웨이 6—인터넷 7—컨텐츠 호스트

Dispatcher는 Caching Proxy 시스템 중 하나가 사용 불가능하거나 요청이 다른 시스템으로 자동으로 라우팅되는 것을 발견합니다. Dispatcher를 사용하면 인터넷 액세스를 인터럽트하지 않고 유지보수할 Caching Proxy 시스템을 종료할 수 있습니다. Dispatcher에는 로드 밸런싱 의사 결정을 위해 고려해야 할 요소를 제어할 수 있는 여러 구성 옵션이 있습니다. 또한 Caching Proxy 시스템에 보조 Dispatcher 프로그램을 설치하여 시스템의 상태를 모니터링하고 Dispatcher에 그 정보를 리턴할 수 있습니다. 자세한 정보는 *WebSphere Application Server Load Balancer* 관리 안내서를 참조하십시오. 여러 개의 Caching Proxy를 사용하면 잠재적인 비효율성을 가지게 됩니다. 이것은 다른 클라이언트가 다른 Caching Proxy 시스템을 통해 파일을 요청하는 경우 두 개 이상의 Caching Proxy가 같은 파일을 캐시할 수 있기 때문입니다. 이러한 중복성을 제거하기 위해 캐시의 콘텐츠를 다른 프록시와 공유하도록 정의된 그룹에서 모든 프록시를 사용 가능하게 하는 RCA(Remote Cache Access)를 구성할 수 있습니다. RCA 그룹 내의 프록시는 Caching Proxy가 주어진 URL에 대해 응답할 수 있도록 모두 같은 알고리즘을 사용합니다. Caching Proxy가 응답할 수 없는 URL을 인터셉트하는 경우, Caching Proxy는 이 요청을 응답 가능한 Caching Proxy로 전달합니다. 응답 가능한 Caching Proxy는 캐시에서 요청을 검색하거나 관련된 콘텐츠 호스트로 요청을 전달하거나 또는 적절한 경우 리턴된 파일을 캐싱하여 이 요청에 대해 필요한 작업을 수행합니다. 그런 다음 응답 가능한 Caching Proxy는 파일을 요청한 일반 사용자에게 전달하는 원래 Caching Proxy로 이 파일을 전달합니다.

주어진 URL에 대해 응답 가능한 Caching Proxy가 실패한 경우, RCA 그룹에서 클라이언트 요청을 받은 원래 Caching Proxy가 콘텐츠 호스트(또는 정의되어 있는 경우

백업 Caching Proxy 서버)를 직접 액세스합니다. 이것은 최소 하나 이상의 Caching Proxy가 RCA 그룹에서 제대로 작동하고 있으면 사용자가 파일을 액세스할 수 있다는 것을 의미합니다.

이 구성은 Dispatcher를 사용하여 여러 Caching Proxy 시스템의 요청을 로드 밸런싱 하므로 인터넷 액세스 수요가 많은 경우에도 가능합니다. 잠재적인 문제점은 Dispatcher가 실패하게 되는 경우입니다. Dispatcher가 실패하거나 네트워크 장애 때문에 액세스할 수 없게 되는 경우, 브라우저 클라이언트는 Caching Proxy 또는 인터넷에 액세스할 수 없습니다. 이 문제를 해결하려면 기본 Dispatcher에 대한 백업으로 작동하는 다른 Dispatcher를 구성하면 됩니다(그림 8의 설명 참조).

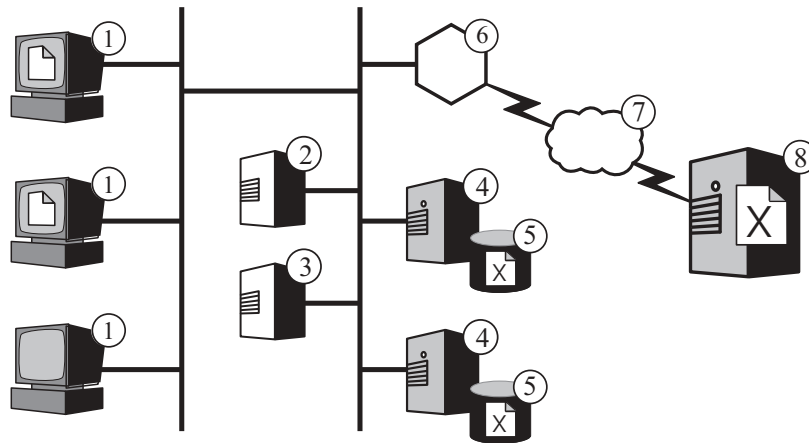


그림 8. 기본 및 백업 Dispatcher를 사용하여 인터넷 액세스의 높은 가용성을 제공. 범례:
1—클라이언트 2—기본 Dispatcher 3—백업 Dispatcher 4—Caching Proxy 5—캐시 6—라우터/게이트웨이 7—인터넷 8—컨텐츠 호스트

보통의 경우 1로 표시된 시스템 중 하나에서 실행하는 브라우저는 X 파일의 요청을 기본 Dispatcher(2)에 전달합니다. 브라우저는 기본 Dispatcher의 로드 밸런싱 기준에 선택된 Caching Proxy(4)로 요청을 라우팅합니다. Caching Proxy는 새 요청을 작성하고 이 요청을 인터넷(7)으로 엔터프라이즈 게이트웨이(6)를 통해 컨텐츠 호스트(8)로 라우팅하며 적절한 경우 리턴된 X 파일을 캐시(5)에 저장합니다(이 프로세스에 대한 자세한 내용은 16 페이지의 『전방향 Caching Proxy』 참조).

이 구성에서 백업 Dispatcher(3)는 기본 Dispatcher가 작동하는 한 로드 밸런스를 수행하지 않습니다. 기본 및 백업 Dispatcher는 하트비트라는 메시지를 주기적으로 교환하면서 서로의 상태를 추적합니다. 백업 Dispatcher가 기본 Dispatcher에 장애가 생긴 것을 발견하면 기본 Dispatcher의 호스트 이름 및 IP 주소에 전달하는 요청을 인터셉트하여 자동으로 로드 밸런스 작업을 수행합니다. 또한 상호적인 고가용성을 위해 두 개의 Dispatcher를 구성할 수 있습니다. 이 경우 각각은 별도의 Caching Proxy의 클러스터에 대해 빈번하게 로드 밸런스를 실행하는 동시에 서로의 백업으로 작동합니다. 자세한 정보는 *WebSphere Application Server Load Balancer* 관리 안내서를 참조하십시오.

일반적으로 Dispatcher는 많은 프로세스나 메모리 자원을 소비하지 않으므로 Dispatcher 시스템에서 다른 응용프로그램을 실행할 수 있습니다. 설비 비용을 최소화하려는 경우, 같은 시스템에서 백업 Dispatcher를 Caching Proxy로 실행할 수도 있습니다. 그림 9는 같은 시스템(3)에서 백업 Dispatcher를 Caching Proxy로 실행하는 구성을 나타냅니다.

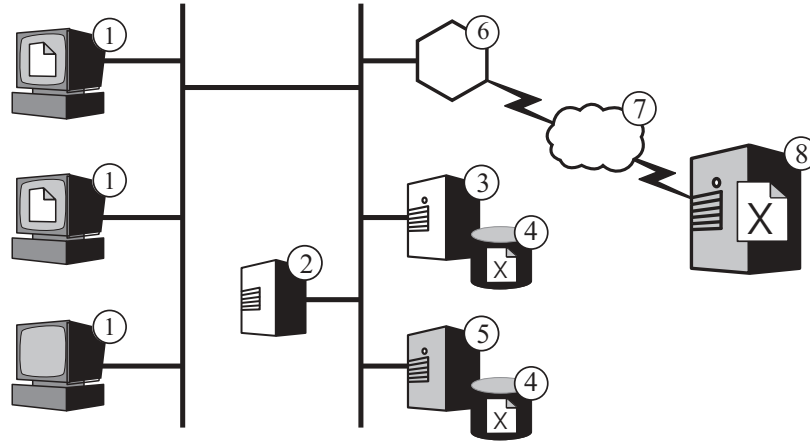


그림 9. Caching Proxy 시스템에서 백업 Dispatcher 찾기. 범례: 1—클라이언트 2—기본 Dispatcher 3—백업 Dispatcher 및 Caching Proxy 4—Caching Proxy 5—캐시 6—라우터/게이트웨이 7—인터넷 8—컨텐츠 호스트

실패복구 지원

Load Balancer는 엔터프라이즈의 콘텐츠 호스트에 대한 단일 연결 지점으로 작동합니다. 사용자는 호스트 이름 및 각 콘텐츠 호스트의 주소가 아니라 DNS의 클러스터 호스트 이름 및 주소를 광고하여 예상치 못한 공격에 대한 보호 레벨과 엔터프라이즈의 웹 사이트에 대한 단일화된 느낌을 제공합니다. 웹 사이트의 가용성을 보다 향상시키려면 34 페이지의 그림 10에 설명되어 있는 대로 다른 Load Balancer를 기본 Load Balancer에 대한 백업으로서 작동하도록 구성하십시오. 하나의 Load Balancer가 고장 나거나 네트워크 장애로 인해 액세스가 불가능하게 되더라도 사용자는 계속해서 콘텐츠 호스트에 접속할 수 있습니다.

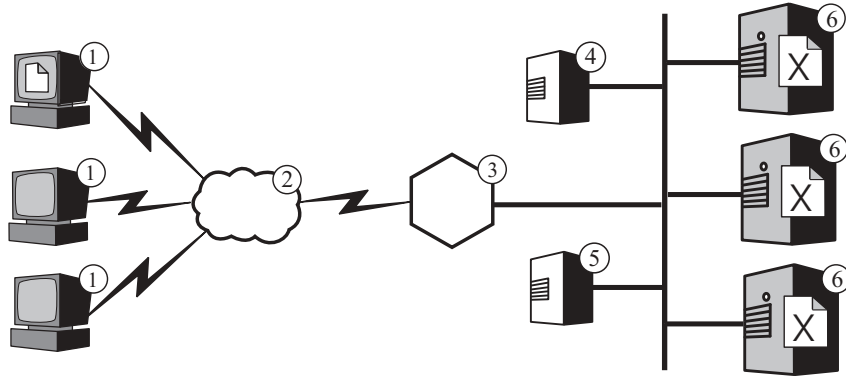


그림 10. 웹 콘텐츠를 최대한으로 사용 가능하게 하기 위해 기본 및 백업 Load Balancer 사용.
 범례: 1—클라이언트 2—인터넷 3—라우터/게이트웨이 4—1차 Dispatcher 5—백업 Dispatcher 6—콘텐츠 호스트

보통의 경우 1로 표시된 시스템 중 하나에서 실행하는 브라우저는 파일 X의 요청을 기본 Load Balancer(4)에 대응하는 클러스터 호스트 이름에 전송합니다. Dispatcher는 요청을 Dispatcher의 로드 밸런스 기준을 기초로 선택한 콘텐츠 호스트(6)로 라우트합니다. 콘텐츠 호스트는 파일 X를 직접 브라우저로 전송하고 기업 게이트웨이(3)를 통해 인터넷(2)을 거쳐 파일을 라우트하지만 Load Balancer는 생략합니다.

백업 Dispatcher(5)는 기본 Dispatcher가 작동 중인 한 로드 밸런스를 수행하지 않습니다. 기본 및 백업 Dispatcher는 핵심 메시지라는 메시지를 주기적으로 교환하면서 서로의 상태를 추적합니다. 백업 Dispatcher가 기본 Dispatcher에 장애가 생긴 것을 발견하면 기본 Dispatcher의 클러스터 호스트 이름 및 IP 주소에 지시된 요청을 가로채어 자동으로 로드 밸런스 작업을 수행합니다.

또한 상호적인 고가용도를 위해 두 개의 Dispatcher를 구성할 수 있습니다. 이 경우 각각은 별도의 콘텐츠 호스트 클러스터를 위해 활발하게 로드 밸런스를 실행하는 동시에 서로의 백업으로 작동합니다. (IPv4 및 IPv6용 Load Balancer 설치 시, 간단한 고가용성은 지원되지만 상호적 고가용성은 지원되지 않습니다.)

일반적으로 Dispatcher는 많은 프로세스나 메모리 자원을 처리하지 않기 때문에, 다른 응용프로그램을 Load Balancer 시스템에서 실행할 수 있습니다. 설비 비용을 최소화하려면, 로드 밸런스 중인 클러스터 내의 시스템 중 하나에 백업 Dispatcher를 실행합니다. 35 페이지의 그림 11은 클러스터의 콘텐츠 호스트(5)에서 실행되는 백업 Dispatcher의 구성을 나타냅니다.

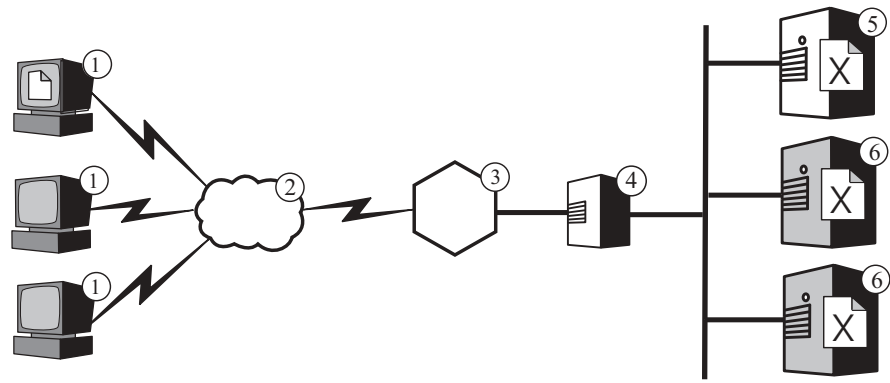


그림 11. 콘텐츠 호스트에서 백업 Load Balancer 찾기. 범례: 1—클라이언트 2—인터넷
3—라우터/게이트웨이 4—1차 Dispatcher 5—백업 Dispatcher 및 콘텐츠 호스트
6—콘텐츠 호스트

제 7 장 Content Based Routing

중요사항: CBR(Content Based Routing) 컴포넌트는 다음 경우를 제외하고 지원되는 모든 플랫폼에서 사용 가능합니다.

- CBR은 64비트 JVM을 실행 중인 플랫폼에서 Load Balancer를 설치할 수 없습니다.

대신, 이 유형의 설치에 대해 Caching Proxy를 사용하지 않고 Load Balancer Dispatcher 컴포넌트의 cbr 전달 메소드를 사용하여 HTTP 및 HTTPS 요청의 Content Based Routing을 제공할 수 있습니다. 자세한 정보는 *WebSphere Application Server Load Balancer 관리 안내서*를 참조하십시오.

- CBR은 IPv4 및 IPv6용 Load Balancer 설치 시 사용할 수 없습니다.

IPv4 및 IPv6용 Load Balancer는 Dispatcher 컴포넌트의 mac 전달 메소드만을 지원합니다. nat 및 cbr 전달 메소드는 지원되지 않습니다.

Application Server Load Balancer 컴포넌트를 Application Server Caching Proxy 컴포넌트와 함께 작동하여 요청을 다른 콘텐츠를 호스트하는 여러 백엔드 서버로 분배할 수 있습니다. (이 Edge Components의 개요에 대해서는 3 페이지의 제 1 장 『WebSphere Application Server Edge Components 소개』를 참조하십시오.)

Load Balancer의 CBR(Content Based Routing) 컴포넌트를 Caching Proxy와 같이 설치하면 HTTP 요청은 URL이나 다른 관리자가 지정한 특성을 기초로 분배될 수 있어서 모든 백엔드 서버에 동일한 콘텐츠를 저장할 필요가 없습니다.

특히 웹 서버가 몇 가지 서로 다른 기능을 수행해야 하거나 여러 가지 유형의 서비스를 제공해야 하는 경우 CBR을 사용하는 것이 적합합니다. 예를 들어, 온라인 소매업체의 웹 사이트는 대부분 고정적인 카탈로그는 물론 항목 번호, 고객 번호를 승인하도록 CGI(Common Gateway Interface) 스크립트 같은 상호작용 응용프로그램을 실행하는 주문서를 표시해야 합니다. 보통, 두 세트의 시스템이 서로 다른 고유한 기능을 수행하도록 하고 CBR을 사용하여 각 시스템에 대해 서로 다른 통신 유형을 라우트하도록 하는 것이 효율적입니다. 비슷한 차원에서 기업은 CBR을 사용하여 유료 요청에 대해 보다 강력한 웹 서버로 경로 지정하여 우연히 들린 방문자보다 유료 고객에게 더 나은 서비스를 제공할 수 있습니다.

CBR은 사용자가 기록한 규칙을 근거로 하는 요청을 라우트합니다. 가장 일반적인 유형은 콘텐츠 규칙이며, URL의 경로 이름을 기초로 하는 요청을 지시합니다. 예를 들어, ABC사는 `http://www.abc.com/catalog_index.html` URL에 대한 요청을 서버의 클러스터 중 하나로 전송하고, `http://www.abc.com/orders.html`은 다른 클러스터로 전송

하는 규칙을 기록할 수 있습니다. 또한 요청을 전송하는 클라이언트의 IP 주소를 기초로 또는 기타 특성을 기초로 하여 요청을 라우트하는 규칙이 있습니다. 자세한 사항은 *WebSphere Application Server Load Balancer 관리 안내서*의 CBR 구성 및 고급 Load Balancer와 CBR 기능을 참조하십시오. 규칙에 관한 구문 정의는 *WebSphere Application Server Load Balancer 관리 안내서*의 부록 CBR 규칙 유형을 참조하십시오.

그림 12는 Load Balancer의 CBR 컴포넌트 및 Caching Proxy가 4로 표시된 시스템에 함께 설치되어 다른 콘텐츠가 있는 세 개의 콘텐츠 호스트(6, 7, 8)에 요청을 라우트하는 간단한 구성을 나타냅니다. 1로 표시된 시스템 중 하나에서 작업하는 일반 사용자가 파일 X를 요청하면, 요청은 인터넷(2)을 거쳐 인터넷 게이트웨이(3)를 통해 기업 내부 네트워크로 들어갑니다. 프록시 서버는 요청을 인터셉트하여 동일한 시스템에 있는 CBR 컴포넌트로 전달합니다. 여기서는 요청의 URL을 구문 분석하여 콘텐츠 호스트 6에 파일 X가 들어 있는지를 판별합니다. 프록시 서버는 파일 X에 대한 새로운 요청을 생성하고 캐시 기능이 사용 가능한 경우 호스트 6에서 파일을 리턴할 때 파일을 캐시할 수 있는지 여부를 판별합니다. 파일 캐시가 가능하면 프록시 서버는 일반 사용자에게 전달하기 전에 캐시(5)에 사본을 저장합니다. 다른 파일의 경로 지정도 같은 방식으로 수행됩니다. 즉, 파일 Y에 대한 요청은 콘텐츠 호스트 7로 이동하고, 파일 Z에 대한 요청은 콘텐츠 호스트 8로 이동합니다.

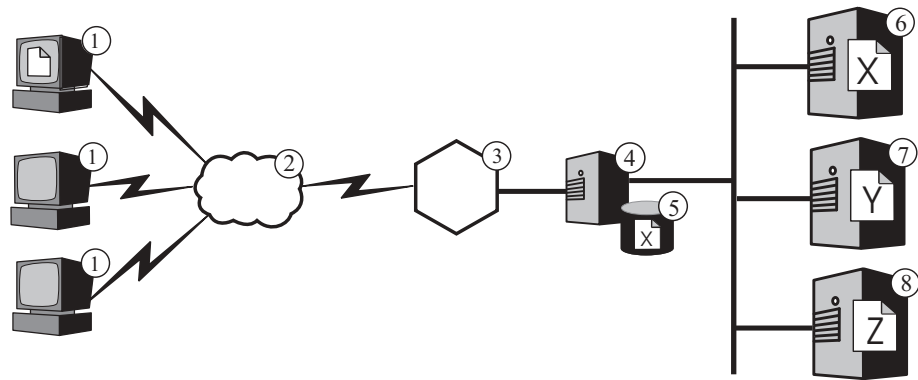


그림 12. CBR로 HTTP 요청 경로 지정. 범례: 1—클라이언트 2—인터넷 3—라우터/게이트웨이 4—Caching Proxy 및 Load Balancer의 CBR 컴포넌트 5—캐시 6, 7, 8—콘텐츠 호스트

39 페이지의 그림 13은 온라인 소매업자에게 적합한 보다 복잡한 구성을 나타냅니다. Load Balancer의 CBR 컴포넌트 및 프록시 서버는 4로 표시된 시스템에 함께 설치되어 있으며 두 대의 Load Balancer 시스템으로 요청을 라우트합니다. 6으로 표시된 Load Balancer 시스템은 소매업체의 카탈로그에서 주로 정적 콘텐츠를 넣는 콘텐츠 호스트(8)의 클러스터를 로드 밸런싱하는 반면, 7로 표시된 Load Balancer는 주문을 처리하는 웹 서버(9)의 클러스터를 로드 밸런싱합니다.

1로 표시된 시스템 중 하나에서 작업하는 일반 사용자가 소매업체의 카탈로그로 가려고 URL에 액세스하면, 요청은 인터넷(2)을 거친 후 인터넷 게이트웨이(3)를 통해 기업 내부 네트워크로 들어갑니다. 프록시 서버는 요청을 인터셉트하여 URL을 구문 분석하며 6의 Load Balancer 시스템에 해당 URL을 처리하는지 판별하는 동일한 시스템의 CBR 컴포넌트로 요청을 전달합니다. 프록시 서버는 새로운 액세스 요청을 작성하여 Load Balancer로 전송합니다. 여기서는 사용자가 정의한 기준을 기본으로 8로 표시된 콘텐츠 호스트 중 어느 것이 현재 요청에 최상의 서비스를 제공할 것인지를 판별합니다. 콘텐츠 호스트는 Load Balancer를 생략하고 카탈로그를 직접 프록시 서버로 전달합니다. 앞의 예제에서처럼 프록시 서버는 콘텐츠가 캐시 가능한 지를 판별하고 캐시가 가능한 경우 캐시(5)에 저장합니다.

일반 사용자는 카탈로그의 하이퍼링크를 통해 소매업체의 주문용 URL에 액세스하여 주문합니다. 요청은 시스템 4의 CBR 컴포넌트가 7의 Load Balancer 시스템으로 라우트하는 것을 제외하고 카탈로그 액세스 요청과 동일한 경로를 통해 이동합니다. Load Balancer는 프록시 서버에 직접 응답하는 9의 웹 서버 중 가장 적합한 서버로 요청을 전달합니다. 주문 정보는 일반적으로 동적으로 생성되기 때문에 프록시 서버는 이를 캐시하지 않을 것입니다.

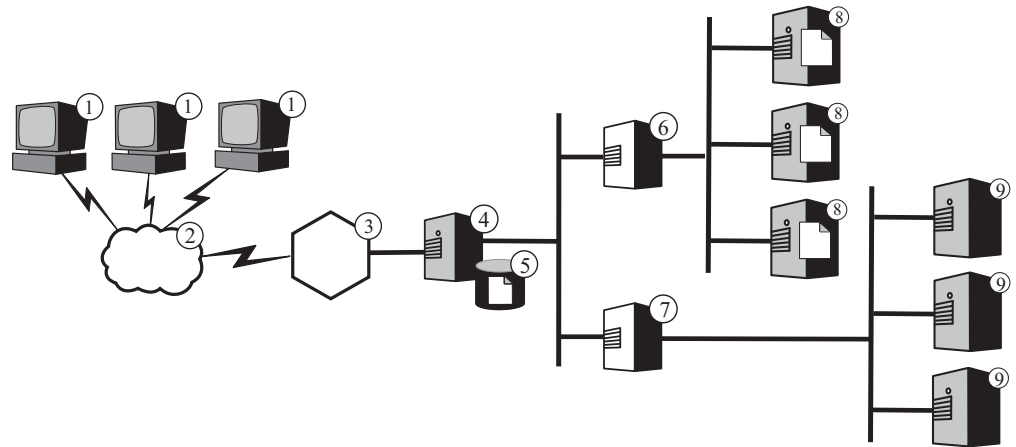


그림 13. CBR로 라우트된 HTTP 요청의 로드 밸런스 범례: 1—클라이언트 2—인터넷 3—라우터/게이트웨이 4—Caching Proxy 및 Load Balancer의 CBR 컴포넌트 5—캐시 6, 7—Load Balancer 8—콘텐츠 호스트 9—웹 서버

Load Balancer의 CBR 기능은 쿠키 유사성을 지원합니다. 이는 일반 사용자의 최초 요청을 서비스한 서버의 ID가 서버의 응답에 포함된 데이터의 특별 패킷(쿠키)에 기록되는 것을 의미합니다. 정의한 시간 내에 일반 사용자가 동일한 URL에 다시 액세스하고 요청에 쿠키가 포함되어 있으면 CBR은 표준 규칙을 다시 적용시키는 대신 원래 서버로 요청을 라우트합니다. 이는 서버가 또 다시 일을 필요가 없는(예: 신용 카드 번호) 일반 사용자에게 관한 정보를 저장한 경우 일반적으로 응답 시간을 단축시켜 줍니다.

제 3 부 시나리오

제 3 부에서는 IBM WebSphere Application Server Edge Components를 사용하는 비즈니스 시나리오에 대해 설명합니다. 이 솔루션은 뛰어난 성능, 가용성, 확장 가능성 및 안정성을 제공하는 구조적으로 완전하며 테스트를 거친 솔루션입니다.

다음 장으로 구성되어 있습니다.

43 페이지의 제 8 장 『B2C 네트워크』

49 페이지의 제 9 장 『B2C banking 솔루션』

51 페이지의 제 10 장 『웹 포털 네트워크』

제 8 장 B2C 네트워크

기본 전자 상거래 웹 사이트는 B2C 네트워크입니다. 인터넷 성장의 첫 번째 단계에서 비즈니스는 일반적으로 단순히 웹 작성에만 초점을 맞추었습니다. 회사의 정보 및 상품 카탈로그가 디지털 형식으로 변환되어 웹 사이트에서 사용 가능하게 되었습니다. 전자 우편 주소, 전화 번호 및 팩스 번호, 자동화된 양식을 제공함으로써 구매도 사용 가능해졌습니다. 그러나 실제 온라인 구매는 사용 가능하지 않습니다. 사람이 주문을 처리해야 하기 때문에 모든 트랜잭션은 잠재되어 있었습니다.

두 번째 단계에서 비즈니스는 이러한 잠재성을 제거하고 직접적인 온라인 구매용 보안 장비구비를 구현하여 판매 조작을 간소화했습니다. 웨어하우스 데이터베이스와의 동기화와 बैंकिंग 시스템과의 통합은 이 판매 트랜잭션을 완료하는 데 매우 중요합니다. 사용할 수 없는 상품은 팔 수 없으며 고객의 계정에 해당 항목에 대한 금액을 청구할 수 없습니다. 마찬가지로 유효한 금융 트랜잭션이 발생할 때까지는 자원 명세 목록에서 상품을 빼거나 고객에게 운송할 수 없습니다.

세 번째 단계에서 회사 웹 사이트는 소비자가 클라이언트의 모습에 익숙해지기 시작하고 개인화된 콘텐츠를 제공하는 동적 웹 프리젠테이션 사이트로 발전합니다.

다음 시나리오에는 Load Balancer 및 Caching Proxy 모두가 포함됩니다.

중요사항: Caching Proxy는 다음 경우를 제외하고 모든 Edge Components 설치에 사용할 수 있습니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행되는 Edge Components 설치에 사용할 수 없습니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge Components 설치에 사용할 수 없습니다.

1단계

44 페이지의 그림 14에 효율적으로 카탈로그를 볼 수 있도록 설계된 소규모 상거래 웹 사이트가 나와 있습니다. 모든 클라이언트 요청은 방화벽을 통해 웹 서버에 대한 대리자 서버로 작동하는 활성 캐시가 있는 프록시 서버 클러스터로 요청을 라우트하는 Dispatcher로 전달됩니다. Metric Server는 Dispatcher에 로드 밸런스 데이터를 제공하기 위해 프록시 서버와 같은 곳에 배치됩니다. 이러한 배열은 웹 서버에서의 네트워크 로드를 줄여 인터넷에 직접 접속하지 않아도 되게 합니다.

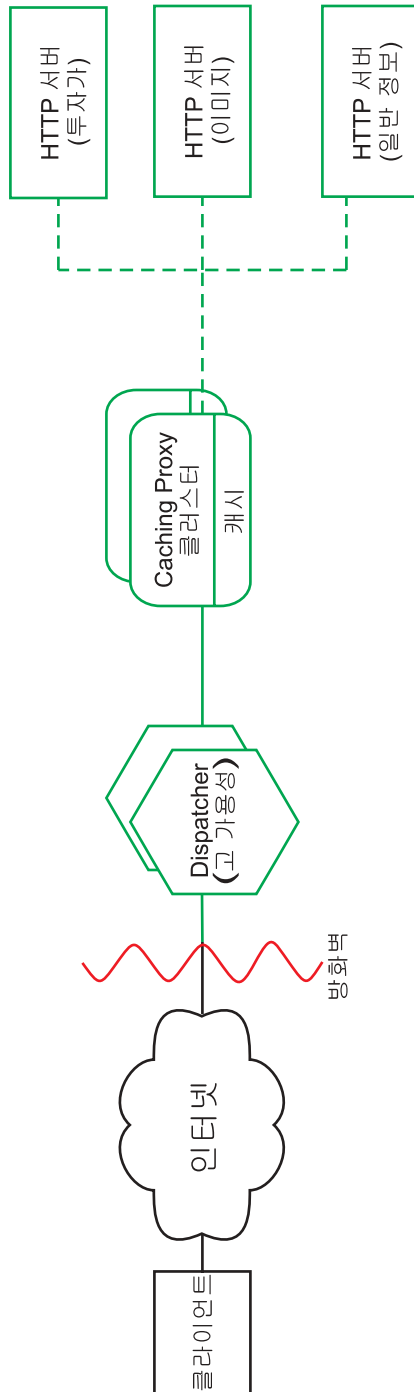


그림 14. B2C 네트워크(1단계)

2단계

46 페이지의 그림 15에 잠재 고객이 효율적으로 카탈로그를 찾아보고 보안되고 빠른 장비구니가 제공되도록 설계된 상거래 웹 사이트에 대한 두 번째 발전 단계가 표시되어 있습니다. 모든 고객 요청은 인터넷 프로토콜을 기본으로 요청을 분리하는 Dispatcher에 의해 적절한 네트워크 분기로 라우트됩니다. HTTP 요청은 정적 웹 사이트로 이동하고 HTTPS 요청은 구매 네트워크로 이동합니다. 기본 정적 웹 사이트는 계속해서 웹 서버의 대리자 역할을 하는 활성 캐시를 가진 프록시 서버 클러스터에 의해 제공됩니다. 이 네트워크 파트는 첫 번째 단계의 네트워크를 이중화합니다.

웹의 전자 상거래 부분은 프록시 서버의 클러스터에 의해서도 제공됩니다. 그러나 여러 가지 플러그인 모듈을 사용하여 Caching Proxy 노드가 향상되었습니다. SSL 핸드셰이킹은 암호 하드웨어 카드로 오프로드되며, 인증은 Access Manager(이전 Policy Director) 플러그인을 통해 수행됩니다. 동적 캐시 플러그인은 일반 데이터를 저장하여 WebSphere Application Server의 작업 부하를 감소시킵니다. Application Server의 플러그인은 필요할 때 Dynacache에 오브젝트를 무효화합니다.

모든 장비구니 응용프로그램은 사용자를 인증하기 위해 사용된 고객 데이터베이스에 연결됩니다. 이는 사용자가 개인 정보를 시스템에 두 번(인증 및 쇼핑 각각에) 입력해야 하는 번거로움을 제거합니다.

이 네트워크는 기본 웹 사이트에서 프로세서 집중형 SSL 인증 및 전자 상거래 장비구니를 제거하며, 클라이언트 사용법에 따라 통신량을 분할합니다. 이중 트랙 웹 사이트는 네트워크 관리자가 네트워크 내부의 서버 역할을 기초로 우수한 성능을 제공하기 위해 다양한 서버를 조정하도록 합니다.

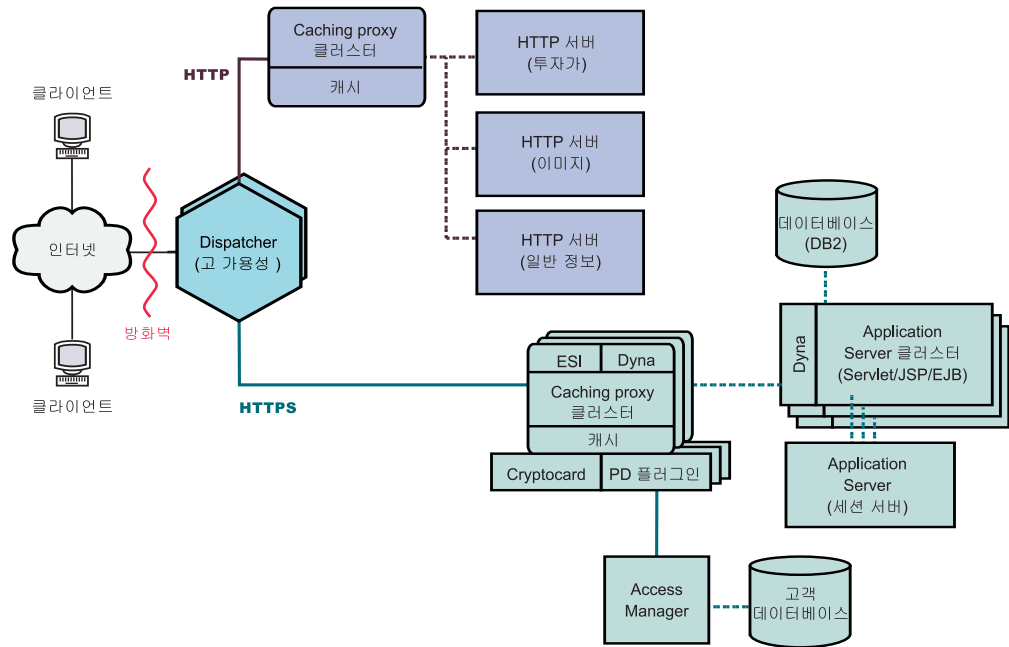


그림 15. B2C 네트워크(2단계)

3단계

47 페이지의 그림 16에 동적 프리젠테이션 메소드를 채택한 정적 웹이 있는 B2C 네트워크의 세 번째 발전 단계가 나와 있습니다. 프록시 서버 클러스터는 동적 웹 콘텐츠의 캐시와 ESI(Edge Side Includes) 프로토콜을 준수하도록 작성된 페이지 단편 어셈블리를 지원하도록 향상되었습니다. ESI 메커니즘은 정보 포함 메커니즘을 사용하여 콘텐츠 서버에 웹 페이지를 작성하여 전체 네트워크에 이러한 클라이언트에 특정한 캐시 불가능 페이지를 전달하기 보다는 네트워크 경계에 있는 캐시된 콘텐츠로부터 페이지를 어셈블할 수 있도록 하기 때문에 대역폭 사용을 줄이고 응답 시간도 단축할 수 있습니다.

ESI 메커니즘은 각 클라이언트가 웹 사이트로부터 개인화된 홈 페이지를 받는 세 번째 단계 시나리오에서는 매우 중요합니다. 이 페이지의 작성 블록은 WebSphere Application Server 시리즈에서 검색됩니다. 중요한 비즈니스 논리를 포함하며 보안 데이터베이스에 연결된 Application Server는 방화벽 뒤에 분리되어 있습니다.

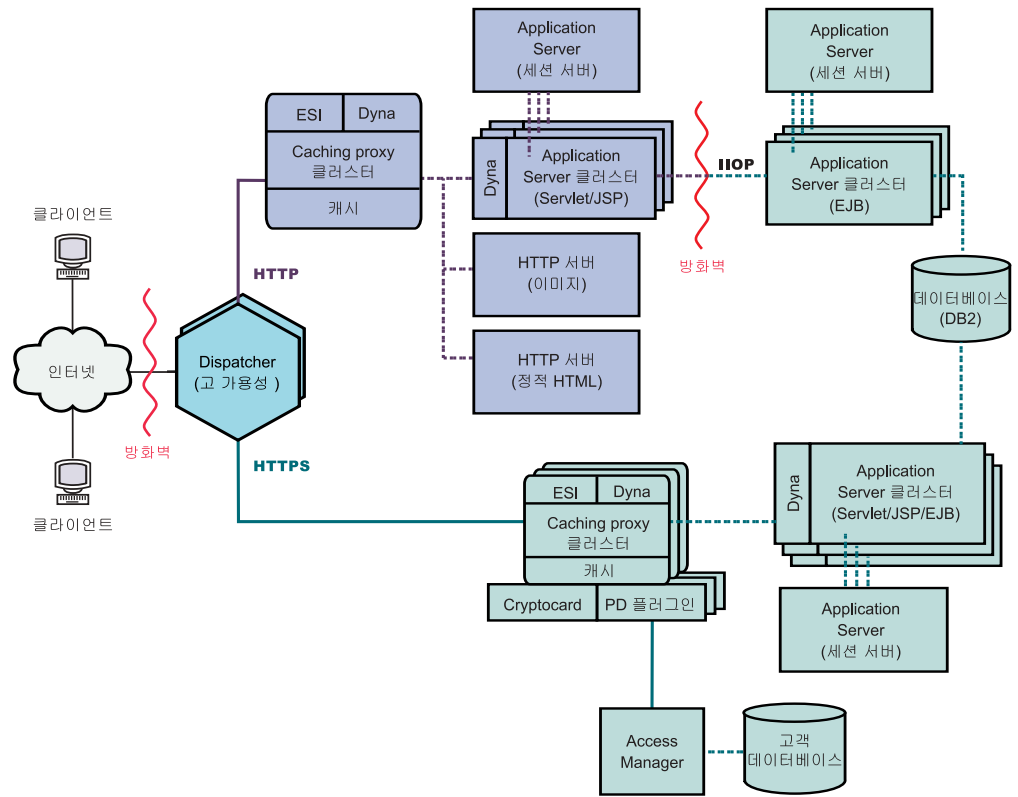


그림 16. B2C 네트워크(3단계)

제 9 장 B2C banking 솔루션

50 페이지의 그림 17에 43 페이지의 제 8 장 『B2C 네트워크』에서 설명한 B2C 네트워크와 유사한 효율적인 온라인 banking 솔루션이 나와 있습니다. 모든 클라이언트 요청은 방화벽을 통해 인터넷 프로토콜에 따라 통신량을 분리하는 Dispatcher에 전달됩니다. HTTP 요청은 웹 서버에 대한 대리자로 작동하는 활성 캐시가 있는 프록시 서버 클러스터로 전달됩니다. Metric Server는 Dispatcher에 로드 밸런스 데이터를 제공하기 위해 프록시 서버와 같은 곳에 배치됩니다. 이러한 배열은 웹 서버에서의 네트워크 로드를 줄여 웹 서버와 인터넷 간에 추가 버퍼를 작성합니다.

HTTPS 요청은 클라이언트에게 개인 금융 정보를 제공하고 온라인 banking 트랜잭션을 허용하도록 설계된 보안 네트워크로 전달됩니다. 향상된 프록시 서버 클러스터는 사이트에 확장성을 제공합니다. 이러한 프록시 서버는 동적 웹 콘텐츠의 캐시와 ESI(Edge Side Includes) 프로토콜을 준수하도록 작성된 페이지 단편 어셈블리를 지원합니다. 암호 하드웨어 카드는 SSL 핸드셰이크를 관리하며, 이는 프록시 서버 호스트에서 필요한 처리를 현저히 줄이고, Access Manager(이전 Policy Director)는 클라이언트 인증을 지시합니다.

Application Server 클러스터 모음은 EJB 컴포넌트에 포함된 비즈니스 논리를 Servlet 및 JSP 파일에 포함된 프리젠테이션 계층과 분리하여 요청의 처리를 분배합니다. 이 클러스터 각각은 별도의 세션 서버에 의해 관리됩니다.

다음 시나리오에는 Load Balancer 및 Caching Proxy 모두가 포함됩니다.

중요사항: Caching Proxy는 다음 경우를 제외하고 모든 Edge Components 설치에 사용할 수 있습니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행되는 Edge Components 설치에 사용할 수 없습니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge Components 설치에 사용할 수 없습니다.

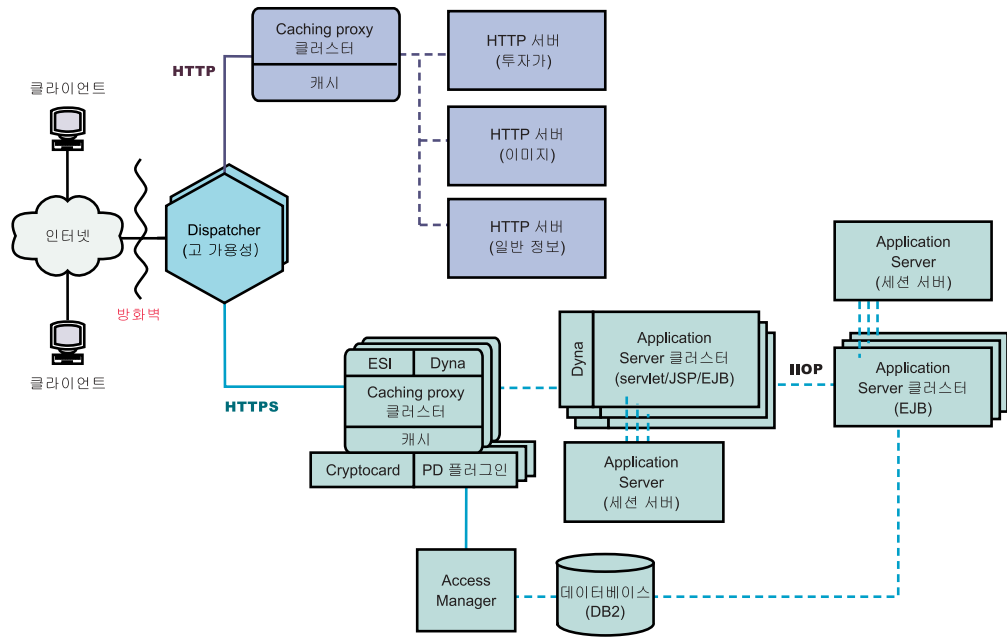


그림 17. B2C banking 솔루션

제 10 장 웹 포털 네트워크

52 페이지의 그림 18에 각 클라이언트에게 개인화된 콘텐츠를 제공하면서 막대한 통신량을 지원하도록 설계된 웹 포털 네트워크가 나와 있습니다. 여러 서버에서의 처리 로드를 최소화하기 위해 모든 네트워크 부분에서 SSL 통신량을 전달하지 않습니다. 포털은 중요한 데이터를 전달하지 않으므로 보안은 중요한 문제점이 아닙니다. 클라이언트 ID, 암호 및 적절하게 보호되고 손상되지 않아야 하는 설정을 포함하는 데이터베이스에 대해서는 보안이 중요하나 이 요구사항은 나머지 웹 사이트의 성능에 영향을 미치지 않습니다.

모든 클라이언트 요청은 웹 서버의 대리자로 작동하는 활성 캐시를 가진 프록시 서버 클러스터를 통해 네트워크 로드를 밸런싱하는 Dispatcher로 전달됩니다. Metric Server는 Dispatcher에 로드 밸런스 데이터를 제공하기 위해 프록시 서버와 같은 곳에 배치됩니다.

실제 동적 웹 사이트는 어셈블리를 위해 프록시 서버로 전달되는 ESI 단편을 생성하는 Application Server 클러스터입니다. 보안 중요성이 감소되었기 때문에 각 Application Server가 웹 사이트 구성에 필요한 모든 기능을 수행합니다. 모든 Application Server가 식별됩니다. 하나의 Application Server가 서비스를 제공할 수 없으면 세션 서버가 요청을 다른 서버로 라우트하여 전체 사이트에 대한 고가용성을 제공합니다. 이 구성은 또한 과도한 통신량이 발생할 경우(예: 포털별로 특수 이벤트 호스트) 신속하게 웹 사이트를 확장할 수 있습니다. 사이트에 추가 프록시 서버 및 Application Server를 신속하게 구성할 수 있습니다.

모든 정적 콘텐츠(예: 이미지 파일 및 반복 텍스트)를 별도의 웹 서버에 저장하여 보다 복잡한 Application Server를 손상할 위험 없이 필요에 따라 콘텐츠를 변경할 수 있습니다.

다음 시나리오에는 Load Balancer 및 Caching Proxy 모두가 포함됩니다.

중요사항: Caching Proxy는 다음 경우를 제외하고 모든 Edge Components 설치에 사용할 수 있습니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행되는 Edge Components 설치에 사용할 수 없습니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge Components 설치에 사용할 수 없습니다.

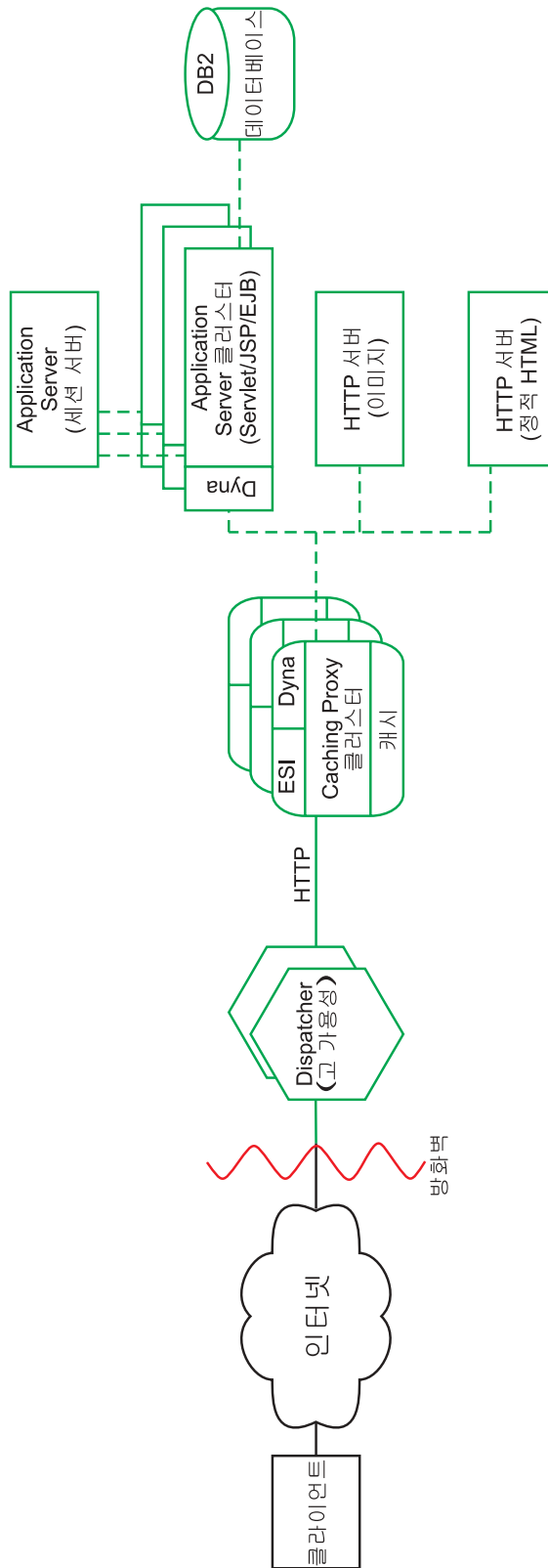


그림 18. 웹 포털

제 4 부 Edge Components 설치

제 4 부에서는 Edge Components 설치 프로시저를 제공합니다.

다음 장으로 구성되어 있습니다.

55 페이지의 제 11 장 『Edge Components에 대한 요구사항』

59 페이지의 제 12 장 『설치 프로그램을 사용한 Edge Components 설치』

63 페이지의 제 13 장 『시스템 패키징 도구를 사용한 Caching Proxy 설치』

67 페이지의 제 14 장 『시스템 패키징 도구를 사용한 Load Balancer 설치』

제 11 장 Edge Components에 대한 요구사항

이 장에서는 Edge Components용 하드웨어 및 소프트웨어 요구사항에 대한 링크와 Caching Proxy 구성 및 관리 양식 및 Load Balancer 온라인 도움말을 포함한 웹 브라우저를 사용하기 위한 지침을 제공합니다.

하드웨어 및 소프트웨어 전제조건

WebSphere Application Server, 버전 6.1 Edge Components에 대해 지원되는 하드웨어 및 소프트웨어 요구사항에 대한 정보는 <http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921> 웹 페이지를 참조하십시오.

SDK 설치: Java 2 SDK는 자동으로 모든 플랫폼에서 Load Balancer와 함께 설치됩니다.

Caching Proxy 구성 및 관리 양식에 브라우저 사용

최소 브라우저 요구사항

구성 및 관리 양식을 사용하여 Caching Proxy를 구성하려면 브라우저가 다음 조건을 만족해야 합니다.

- 프레임 표시
- JavaScript와 Java 모두를 사용 가능하게 함
- 색상 해상도를 최소한 256가지 색으로 설정(운영 체제 설정)
- 캐시 문서로 설정하고 캐시된 문서를 항상 네트워크 문서와 비교

Linux 및 UNIX 시스템의 경우: Mozilla 및 Firefox 브라우저의 권장되는 버전은 다음 웹 사이트를 참조하고 지원되는 소프트웨어 웹 페이지로 가려면 해당 링크를 누르십시오. <http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921>

Windows 시스템의 경우: Internet Explorer, Mozilla 및 Firefox 브라우저의 권장되는 버전은 다음 웹 사이트를 참조하고 지원되는 소프트웨어 웹 페이지로 가려면 해당 링크를 누르십시오. <http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921>

주: 64비트 PowerPC Linux 시스템에서는 이 구조용으로 사용 가능한 SDK가 없으므로 Mozilla 브라우저를 통해 구성 및 관리 양식에 액세스할 수 없습니다. 대신, 지원되는 웹 브라우저를 통해 다른 시스템에서 구성 및 관리 양식에 액세스할 수 있습니다.

제한사항: 펼친 요소의 수가 너무 많아서 브라우저 창에 표시할 수 없는 경우, 관리 양식 왼쪽의 세로 화면이동 막대가 브라우저에 표시되지 않을 수 있습니다. 따라서 목록 맨 아래의 펼쳐진 요소는 브라우저의 현재 보고 있는 창에서 밀려나 해당 요소에는 액세스할 수 없습니다. 이 문제점을 해결하려면 왼쪽 메뉴에 펼쳐지는 요소의 수를 제한하십시오. 펼쳐진 요소의 수가 큰 경우, 목록의 맨 아래에 있는 요소가 브라우저 창에 표시될 때까지 요소를 접으십시오.

양식을 올바르게 표시하려면 실제로 양식을 표시하는 운영 체제(브라우저가 상주하는 운영 체제)에 양식이 기록된 언어에 맞는 글꼴 집합이 있어야 합니다. 그러나 브라우저 인터페이스가 반드시 양식과 동일한 언어로 기록되지 않아도 됩니다.

예를 들어, 프록시 서버의 중국어 버전이 Solaris 9 시스템에서 실행됩니다. 영어 인터페이스의 Mozilla 브라우저는 Solaris 호스트에 로드됩니다. 이 브라우저는 구성 및 관리 양식을 로컬로 편집하는 데 사용할 수 있습니다. (프록시 서버에서 사용하는 문자-세트로 브라우저에 양식이 제공됩니다(예: 중국어). 그러나 브라우저 및 해당 기본 운영 체제가 프록시 서버에서 전송한 문자-세트를 제대로 표시하도록 구성되지 않은 경우 양식이 제대로 표시되지 않을 수 있습니다.)

또는 원격으로 프록시 서버에 연결할 때 중국어가 지원되는 Windows 워크스테이션을 사용할 수 있는 경우, Netscape 브라우저의 중국어 버전을 Windows 워크스테이션에 로드하고 이 브라우저를 사용하여 양식의 값을 편집할 수 있습니다. 두 번째 해결 방법은 관리자가 일관된 언어 인터페이스를 유지할 수 있다는 장점이 있습니다.

운영 체제에 특정된 글꼴 집합은 브라우저에서 특히 2바이트 문자의 다양한 언어 표시에 큰 영향을 미칩니다. 예를 들어, AIX의 특정 중국어 글꼴 집합은 Windows 플랫폼의 중국어 글꼴 집합과 정확히 동일하게 표시되지 않습니다. 이것은 구성 및 관리 양식 내의 HTML 텍스트와 Java 애플릿의 모양이 불규칙하기 때문입니다. 가장 잘 보이도록 Windows 운영 체제에서 실행되는 브라우저만 권장됩니다.

S/390 및 PowerPC의 Mozilla 1.4 브라우저에 대한 정보

관리 양식을 제대로 표시하려면 Mozilla 1.4와 함께 설치된 Java 플러그인을 버전 1.4.2 이상으로 갱신해야 합니다. 플러그인을 갱신하려면 다음 단계를 사용하십시오.

1. <http://plugindoc.mozdev.org>로 링크하십시오.
2. "Documentation" 섹션에서 플랫폼을 선택하십시오.
3. "JRE(Java Runtime Environment)"에 나열된 지시사항을 따라 플러그인을 갱신하십시오.

Load Balancer 온라인 도움말에 브라우저 사용

Load Balancer 온라인 도움말을 사용하려면 브라우저가 다음을 지원해야 합니다.

- HTML 4
- Cascading Style Sheets
- JavaScript 테크놀로지
- Java 애플릿

이 요구사항을 지원하지 않는 브라우저를 사용하면 페이지가 제대로 형식화되지 않거나 기능이 제대로 작동하지 않을 수 있습니다.

- **Linux 및 UNIX 시스템의 경우:** 기본 브라우저는 Mozilla 및 Firefox입니다. 해당 브라우저의 권장되는 버전은 다음 웹 사이트를 참조하고 지원되는 소프트웨어 웹 페이지로 가려면 해당 링크를 누르십시오.

<http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921>

- **Windows 시스템의 경우:** 기본 브라우저는 시스템 브라우저입니다. Internet Explorer, Mozilla 및 Firefox 브라우저의 권장되는 버전은 다음 웹 사이트를 참조하고 지원되는 소프트웨어 웹 페이지로 가려면 다음 링크를 누르십시오.

<http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921>

제 12 장 설치 프로그램을 사용한 Edge Components 설치

이 장은 설치 프로그램을 사용하여 Edge Components를 설치하는 명령을 제공합니다.

Java 2 SDK는 자동으로 모든 플랫폼에서 Load Balancer와 함께 설치됩니다.

설치 이후에 Caching Proxy 패키지의 스크립트는 기본 구성을 사용하여 프록시 서버 시작합니다. 포트 80을 사용 중이면 다른 웹 서버에서와 같이 프록시 서버도 시작에 실패합니다.

중요사항: Caching Proxy는 다음 경우를 제외하고 모든 Edge Components 설치에 사용할 수 있습니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행되는 Edge Components 설치에 사용할 수 없습니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge Components 설치에 사용할 수 없습니다.

Windows용 설치 프로그램 사용

설치 프로그램을 사용하여 다음과 같이 Windows® 시스템에 Edge Components를 설치하십시오.

1. Windows 시스템이 모든 하드웨어 및 소프트웨어 요구사항(55 페이지의 제 11 장 『Edge Components에 대한 요구사항』)을 충족시키는지 확인하십시오.
2. 관리자 권한을 가진 사용자로 로그인하십시오.
3. 시스템의 CD-ROM 드라이브에 Edge Components CD-ROM을 넣으십시오. LaunchPad가 자동으로 시작됩니다.
4. **WebSphere Application Server – Edge Components**의 설치 마법사 실행을 누르십시오. 설치 프로그램이 자동으로 시작됩니다. InstallShield 마법사가 준비되고 환영 창이 열립니다.

주: 시스템이 자동 실행 옵션을 지원하지 않거나 꺼져 있는 경우, CD-ROM의 최상위 레벨 디렉토리에 있는 **setup.exe** 프로그램을 실행하여 설치 프로그램을 수동으로 시작하십시오.

5. 다음을 눌러 설치를 계속하십시오. 소프트웨어 라이선스 계약 창이 열립니다.
6. 라이선스 계약을 읽고 예를 눌러 모든 해당 항목을 승인하십시오. 컴포넌트 선택 창이 열립니다.

Edge Components를 이미 설치한 경우, 컴포넌트 선택 창이 열리기 전에 유지보수 옵션 창이 열립니다. 수정 단일 선택 단추를 선택한 후 다음을 누르십시오. 컴포넌트 선택 창이 열립니다.

7. 설치할 컴포넌트를 선택하십시오.
8. 주어진 컴포넌트에 설치할 부속 컴포넌트의 선택을 변경하려면 이름을 누른 다음 선택할 컴포넌트의 부속 컴포넌트 변경을 누르십시오. 활성 컴포넌트의 부속 컴포넌트를 표시하는 또다른 컴포넌트 선택 창이 열립니다. 동일한 프로시저를 사용하여 설치할 부속 컴포넌트, 컴포넌트의 언어 및 컴포넌트를 설치할 장소를 선택하십시오.
9. 현재 언어 메뉴를 사용하여, 언어 또는 Edge Components를 설치하려는 언어를 선택하십시오. 사용 가능한 언어가 왼쪽의 메뉴에 나열됩니다. 선택한 언어가 오른쪽 메뉴에 나열됩니다.
10. 컴포넌트 선택 창을 사용하여, Edge Components에 대한 설치 위치를 확인하십시오. 기본값을 승인하거나 폴더 변경을 눌러서 새 위치를 지정할 수 있습니다.

주: 기본값 대신 다른 설치 위치를 선택할 경우, 경로 이름에 공백이 없는지 확인하십시오. (예를 들어, C:\My Files\edgeserver\와 같은 이름은 피하십시오.)

11. 컴포넌트 선택 창을 사용하여, 선택한 설치 위치에 사용 가능한 영역이 충분한지 확인하십시오. 선택한 위치에 사용 가능한 영역이 충분하지 않을 경우, 폴더 변경을 누르고 설치 위치를 새로 지정하십시오.
12. Edge Components, 설치 및 언어를 선택한 후 다음을 누르십시오. 열린 설치 확인 창의 정보를 검토하십시오. 선택을 변경하려는 경우, 뒤로를 누르고 컴포넌트 선택 창으로 리턴하여 변경을 작성하십시오. 선택사항을 확인한 후 완료를 누르십시오.
13. Edge Components 제품 설치 프로그램은 지정한 설치 위치에 선택한 Edge Components가 필요한 경우, GSK를 설치하기 시작합니다.
14. 설정 완료 창이 열립니다. Edge Components ReadMe 파일을 읽으려면, 예, **ReadMe** 파일을 보겠습니다 선택란을 선택했는지 확인하십시오. ReadMe 파일이 사용자의 기본 브라우저에 열립니다.
15. 예, 컴퓨터를 재시작합니다 선택란을 선택했는지 확인하고 완료를 누르십시오. ReadMe 파일 보기를 선택한 경우, 파일을 표시하는 브라우저 창을 닫을 때 시스템이 재시작됩니다. 그렇지 않으면 Edge Components 제품 설치 프로그램이 즉시 닫히고 시스템이 재시작됩니다. 새로 설치된 Edge Components를 사용하기 전에 시스템을 재시작해야 함에 유의하십시오.

제한사항: 라이선스 계약 창에서 Tab 키를 사용하면 동의함 및 동의하지 않음 옵션을 토글합니다. 그러나 Tab 키를 사용하여 이전, 다음 또는 취소 탐색 옵션을 선택할 수

는 없습니다. 이 문제를 해결하려면 Shift+Tab을 사용하여 해당 탐색 옵션을 선택하십시오. 또한 Enter 키는 탐색 버튼에서만 작동하므로 동의함 또는 동의하지 않음 옵션을 선택하려면 스페이스바를 사용해야 합니다.

Linux 및 UNIX용 설치 프로그램 사용

CD에서 설치하는 경우 설치 프로그램을 사용하여 다음과 같이 Linux 및 UNIX 시스템에 Edge Components를 설치하십시오.

1. 컴퓨터 서버가 55 페이지의 제 11 장 『Edge Components에 대한 요구사항』에 설명된 모든 하드웨어와 소프트웨어 요구사항을 충족시키는지 확인하십시오.
 - Linux 시스템의 추가사항: GCC 3.3 C++ 호환 라이브러리를 가지는 compat-libstdc++-33 패키지가 설치되어 있어야 합니다.
2. 슈퍼유저로 로그인 시 일반적으로 root를 입력하십시오.
3. 시스템의 CD-ROM 드라이브에 Edge Components CD-ROM을 넣으십시오. 필요하다면 CD-ROM을 마운트하십시오.
4. CD-ROM의 맨 위 레벨로 작업 디렉토리를 변경하십시오.
5. 다음 명령을 입력하여 설치 프로그램을 호출하십시오.

```
# ./install
```

환영 창이 열립니다.

6. 다음을 눌러 설치를 계속하십시오. 소프트웨어 라이선스 계약 창이 열립니다.
7. 라이선스 계약을 읽고 예를 눌러 모든 해당 항목을 승인하십시오. 언어 선택 창이 열립니다.
8. Edge Components의 설치로 지원된 언어를 선택하십시오. 다음을 누르십시오. 컴포넌트 선택 창이 열립니다.
9. 설치할 컴포넌트를 선택하십시오.
10. 다음을 누르십시오. 설치 확인 창이 열립니다.
11. 설치 확인 창의 정보를 검토하십시오. 하나 이상의 선택을 변경하려는 경우, 뒤로를 누르고 컴포넌트 선택 창으로 리턴하여 변경을 작성하십시오. 선택사항을 확인한 후 완료를 누르십시오.

설치 프로그램은 선택한 Edge Components 및 필수 패키지의 설치를 시작합니다.

12. 설치 결과 요약 창이 열립니다. 결과를 검토한 후 완료를 누르십시오.

제한사항: 라이선스 계약 창에서 Tab 키를 사용하면 동의함 및 동의하지 않음 옵션을 토글합니다. 그러나 Tab 키를 사용하여 이전, 다음 또는 취소 탐색 옵션을 선택할 수는 없습니다. 이 문제를 해결하려면 Shift+Tab을 사용하여 해당 탐색 옵션을 선택하십시오. 또한 Enter 키는 탐색 버튼에서만 작동하므로 동의함 또는 동의하지 않음 옵션을 선택하려면 스페이스바를 사용해야 합니다.

Red Hat Linux 3.0 갱신 3의 경우: Edge Components용 설치 프로그램을 실행할 때 GUI 패널을 최대화한 다음 다시 줄이면 단추가 작동되지 않습니다. 이 문제점을 해결하려면 다음을 수행하십시오.

1. 설치 프로그램을 닫으려면 패널의 맨 위 오른쪽 구석에 있는 **X** 단추를 클릭하십시오.
2. "종료하시겠습니까?" 질문에 **예**라고 응답하십시오.
3. 패널을 최대화하여 복원하지 않고 설치 프로그램을 다시 시작하십시오.

Linux 및 UNIX 시스템의 경우: 설치 프로그램을 사용하여 Edge Components를 설치한 경우, GUI 설치 제거 프로그램을 사용하여 Edge Components를 설치 제거할 수 있습니다. 그러나 Edge Components GUI 설치 제거 프로그램은 기본 명령을 사용하여 설치된 갱신 팩을 설치 제거할 수 없습니다. GUI 설치 제거 프로그램을 사용하여 컴포넌트를 설치 제거하기 전에 기본 명령(운영 체제 명령)을 사용하여 갱신 팩을 먼저 설치 제거하십시오.

기본 명령 사용에 대한 정보는 63 페이지의 제 13 장 『시스템 패키징 도구를 사용한 Caching Proxy 설치』 및 67 페이지의 제 14 장 『시스템 패키징 도구를 사용한 Load Balancer 설치』를 참조하십시오.

제 13 장 시스템 패키징 도구를 사용한 Caching Proxy 설치

이 장에서는 시스템 패키징 도구를 사용하여 Caching Proxy를 설치하는 명령을 제공합니다.

설치 이후에 Caching Proxy 패키징의 스크립트는 기본 구성을 사용하여 프록시 서버 시작합니다. 포트 80을 사용 중이면 다른 웹 서버에서와 같이 프록시 서버도 시작에 실패합니다.

중요사항: Caching Proxy는 다음 경우를 제외하고 모든 Edge Components 설치에 사용할 수 있습니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행되는 Edge Components 설치에 사용할 수 없습니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge Components 설치에 사용할 수 없습니다.

운영 체제의 패키지 설치 시스템을 사용하여 64 페이지의 표 2에 나열된 순서에 따라 패키지를 설치하십시오. 다음의 프로시저는 이 작업을 완료하는 데 필요한 일반적 단계를 상세히 설명합니다.

1. Edge Components CD를 CD-ROM 드라이브에 넣고 필요한 경우 드라이브를 마운트하십시오.

2. 로컬 슈퍼유저 루트로 변경하십시오.

```
su - root  
Password: password
```

3. CD의 해당 디렉토리로 변경하십시오.

```
cd mount_point/package_directory/
```

4. 패키지를 설치하십시오.

AIX®의 경우:

```
installp -acXd ./packagename
```

HP-UX의 경우:

```
swinstall -s source/ packagename
```

Linux의 경우:

```
rpm -i ./packagename
```

Solaris의 경우:

```
pkgadd -d ./packagename
```

표 2. Caching Proxy 컴포넌트

컴포넌트	설치한 패키지(권장 순서에 따라)
Caching Proxy	<ol style="list-style-type: none"> 1. gskit7 2. icu 3. admin 4. msg-cp-lang 5. cp
Edge component 문서화	doc-en_US ¹
<p>주:</p> <p>1. Load Balancer 문서는 두 개의 패키지로 제공됩니다. doc-en_US 패키지에는 Load Balancer 문서를 포함하여 모든 Edge Components 문서가 있으며 해당 문서는 ../edge/doc/ 디렉토리에 설치됩니다. Load Balancer 설치(67 페이지의 제 14 장 『시스템 패키징 도구를 사용한 Load Balancer 설치』)와 연관된 문서 패키지에는 Load Balancer 문서만이 있으며 해당 문서는 ../edge/lb/ 디렉토리에 설치됩니다.</p>	

표 3. AIX, HP-UX 및 Solaris 패키지 파일 이름

일반 패키지 이름	AIX 파일 세트	HP-UX 파일 세트	Solaris 파일 이름
admin	wses_admin.rte	WSES-ADMIN	WSESadmin
cp	wses_cp.base	WSES-CP	WSEScp
문서	wses_doc.en_US	WSES-DOC-en_US	WSESdocen
gskit7	gskkm.rte	gsk7bas	gsk7bas
icu	wses_icu.rte	WSES-ICU	WSESicu
msg-cp-lang	wses_cp.msg.lang ¹ .base	WSES-cpmlang ²	WSEScpmlang ³
<p>주:</p> <p>1. AIX에서 lang 변수는 다음 언어 특정 코드 중 하나입니다. en_US, de_CH, de_DE, es_ES, fr_CA, fr_CH, fr_FR, it_CH, it_IT, ja_JP, Ja_JP, ko_KR, pt_BR, zh_CN, ZH_CN, zh_TW, Zh_TW.</p> <p>2. HP-UX에서 lang 변수는 다음 언어 특정 코드 중 하나입니다. de_DE, en_US, es_ES, fr_FR, it_IT, ja_JP, ko_KR, zh_CN, zh_TW. (HP-UX는 브라질 포르투갈어(pt_BR)를 지원하지 않습니다.)</p> <p>3. Solaris에서 변수 lang은 다음 언어 특정 코드(br, cn, cw, de, en, es, fr, it, ja, kr) 중 하나의 대체를 나타냅니다.</p>			

표 4. Linux 패키지 파일 이름

일반 패키지 이름	Linux 파일 이름
admin	WSES_Admin_Runtime-release-version ¹ .hardw ² .rpm
cp	WSES_CachingProxy-release-version ¹ .hardw ² .rpm
문서	WSES_Doc_en_US-release-version ¹ .hardw ² .rpm
gskit7	gsk7bas.rpm
icu	WSES_ICU_Runtime-release-version ¹ .hardw ² .rpm
msg-cp-lang	WSES_CachingProxy_msg_lang ³ -release-version ¹ .hardw ² .rpm

표 4. Linux 패키지 파일 이름 (계속)

일반 패키지 이름	Linux 파일 이름
<p>주:</p> <ol style="list-style-type: none"> 1. <i>release-version</i>은 현재 릴리스입니다(예: 6.1.0-0). 2. <i>hardw</i> 변수는 다음 중 하나입니다. i686, s390, ppc64. 3. <i>lang</i> 변수는 다음 언어 특정 코드 중 하나입니다. de_DE, en_US, es_ES, fr_FR, it_IT, ja_JP, ko_KR, pt_BR, zh_CN, zh_TW. 	

문서 패키지는 영어만을 포함합니다. 자국어로 된 Edge Component 문서 설정은 다음 웹 사이트에 있습니다.

www.ibm.com/software/webervers/appserv/ecinfocenter.html.

시스템 도구를 사용하여 Caching Proxy 설치 제거

패키지를 설치 제거하려면 다음을 수행하십시오.

AIX의 경우:

```
installp -u packagename
```

모든 Caching Proxy 패키지를 설치 제거하려면 다음 명령을 사용하십시오.

```
installp -u wses
```

HP-UX의 경우:

```
swremove packagename
```

설치된 Caching Proxy 패키지를 조회하려면 다음 명령을 사용하십시오.

```
swlist | grep WSES
```

패키지는 설치된 순서의 반대로 제거해야 합니다.

Linux의 경우:

```
rpm -e packagename
```

설치된 Caching Proxy 패키지를 조회하려면 다음 명령을 사용하십시오.

```
rpm -qa |grep -i wses
```

패키지는 설치된 순서의 반대로 제거해야 합니다.

Solaris의 경우:

```
pkgrm packagename
```

설치된 Caching Proxy 패키지를 조회하려면 다음 명령을 사용하십시오.

```
pkginfo | grep WSES
```

패키지는 설치된 순서의 반대로 제거해야 합니다.

제 14 장 시스템 패키징 도구를 사용한 Load Balancer 설치

이 장에서는 AIX, HP-UX, Linux 및 Solaris 시스템에 Load Balancer 설치하는 방법을 설명합니다.

- 68 페이지의 『AIX 설치』
- 71 페이지의 『HP-UX에 설치』
- 73 페이지의 『Linux에 설치』
- 75 페이지의 『Solaris에 설치』

설치 유형에 따라 이 섹션에 나열된 모든 Load Balancer 컴포넌트 패키지가 제공되지 않을 수 있습니다.

- Load Balancer와 Caching Proxy 모두를 제공할 수 있는 Edge Components 설치의 경우, 모든 Load Balancer 설치 컴포넌트 패키지를 사용할 수 있습니다.
- Load Balancer를 제공하지만 Caching Proxy는 제공할 수 없는 Edge Components 설치의 경우, CBR 컴포넌트 패키지는 Load Balancer에 포함되지 않습니다.
- IPv6용 Edge Components 설치(IPv4 및 IPv6용 Load Balancer)의 경우, Dispatcher 컴포넌트 패키지가 Load Balancer에 포함됩니다. CBR, Site Selector 및 Controller 컴포넌트 패키지는 포함되지 않습니다.

패키지를 설치하는 바람직한 순서는 IPv4 및 IPv6용 Load Balancer 설치에 대해 조금씩 다릅니다. 그러나 관리 컴포넌트 패키지는 반드시 Dispatcher 컴포넌트 패키지 설치 후에 설치되어야 함을 주의하십시오. 시스템 도구를 사용하여 IPv4 및 IPv6용 Load Balancer를 설치하는 바람직한 순서는 기본, 라이선스, Dispatcher 컴포넌트, 관리, 문서, Metric Server 순입니다.

이전 버전의 Load Balancer에서 이주하거나 운영 체제를 다시 설치할 경우, 설치 전에 Load Balancer의 이전 구성 파일 또는 스크립트 파일을 저장할 수 있습니다.

- 설치 후 ...ibm/edge/lb/servers/configurations/**component** 디렉토리에 구성 파일을 넣으십시오(여기서, **component**는 Dispatcher, cbr, ss, cco 또는 nal).
- 설치 후에 스크립트 파일(예: goIdle 및 goStandby)을 실행하려면 .../ibm/edge/lb/servers/bin 디렉토리에 넣으십시오.

Load Balancer가 설치된 후 시스템에서 로그오프한 경우 다시 로그인할 때 모든 Load Balancer 서비스를 다시 시작해야 합니다.

AIX 설치

표 5는 Load Balancer에 대한 AIX 파일 세트를 나열하며 시스템 패키지 설치 도구를 사용하여 설치하는 바람직한 순서대로 나열되어 있습니다.

표 5. AIX 파일 세트

Load Balancer 컴포넌트	AIX 파일 세트
기본	ibmlb.base.rte
관리(메시지)	<ul style="list-style-type: none">• ibmlb.admin.rte• ibmlb.msg.lang.admin
장치 드라이버	ibmlb.lb.driver
라이선스	ibmlb.lb.license
Load Balancer 컴포넌트(메시지)	<ul style="list-style-type: none">• ibmlb.component.rte• ibmlb.msg.lang.lb
문서(메시지)	<ul style="list-style-type: none">• ibmlb.doc.rte• ibmlb.msg.en_US.doc
Metric Server	ibmlb.ms.rte

주:

1. *component* 변수를 disp(Dispatcher), cbr(CBR), ss(Site Selector), cco(Cisco CSS Controller) 또는 nal(Nortel Alteon Controller)로 대체할 수 있습니다.
2. *lang* 변수는 다음 중 하나로 대체될 수 있습니다. en_US, de_CH, de_DE, es_ES, fr_CA, fr_CH, fr_FR, it_CH, it_IT, ja_JP, Ja_JP, ko_KR, pt_BR, zh_CN, ZH_CN, zh_TW, Zh_TW

문서 패키지는 영어만을 포함합니다. 자국어로 된 Edge Component 문서 설정은 다음 웹 사이트에 있습니다.

www.ibm.com/software/webservers/appserv/ecinfocenter.html.

설치하기 전에

AIX용 Load Balancer를 설치하기 전에 다음을 확인하십시오.

- 루트로 로그인했는지 확인하십시오.
- Edge Components 매체를 넣었는지 또는 웹에서 설치할 경우 설치 이미지를 디렉토리로 복사했는지 확인하십시오. 이전 제품 버전을 설치 제거했는지 확인하십시오. 설치 제거하려면, 모든 실행 프로그램과 모든 서버가 정지되었는지 확인하십시오. 그런 다음, 제품 전체를 설치 제거하고 다음 명령을 입력하십시오.

```
installp -u ibmlb
```

이전 버전의 경우에는 다음 명령을 입력하십시오.

```
installp -u ibmnd
```

특정 파일 세트를 설치 제거하려면, 패키지 이름 `ibm1b`를 지정하지 말고 명확하게 해당 파일 세트를 나열하십시오.

제품을 설치할 때, 다음 중에 하나 또는 모두를 설치하기 위한 옵션이 제공됩니다.

- Base Administration
- 관리(메시지)
- 장치 드라이버(필수)
- 라이선스(필수)
- Dispatcher 컴포넌트(메시지)
- CBR 컴포넌트(메시지)
- Site Selector 컴포넌트(메시지)
- Cisco CSS Controller 컴포넌트(메시지)
- Nortel Alteon Controller 컴포넌트(메시지)
- 문서(메시지)
- Metric Server

설치 프로시저

SMIT는 모든 메시지가 자동으로 설치되어야 하므로, SMIT를 사용하여 AIX용 Load Balancer를 설치하는 것이 바람직합니다.

SMIT를 사용하여 AIX용 Load Balancer 설치

1. 소프트웨어 설치 및 관리를 선택하십시오.
2. 소프트웨어 설치 및 갱신을 선택하십시오.
3. 사용 가능한 최신 소프트웨어에서 설치 및 갱신을 선택하십시오.
4. 파일 세트를 포함하는 장치 또는 디렉토리를 입력하십시오.
5. 설치할 *SOFTWARE 필드에 옵션을 지정할(또는 목록을 선택할) 적절한 정보를 입력하십시오.
6. 확인을 누르십시오.
7. 명령이 완료되면 완료를 누르십시오.
8. 편집 메뉴에서 **Smit** 종료를 선택하거나 **F12**를 눌러 SMIT를 닫으십시오. SMITTY를 사용하려면 **F10**을 눌러 프로그램을 닫으십시오.

명령행에서 Load Balancer 설치

1. CD에서 설치할 경우, CD를 마운트하려면 다음 명령을 입력하십시오.

```
mkdir /cdrom
mount -v cdrfs -p -r /dev/cd0 /cdrom
```

2. 필요한 AIX용 Load Balancer 패키지를 설치하기 위해 입력할 명령 또는 명령들을 판별하려면 다음 표를 참조하십시오.

표 6. AIX 설치 명령

패키지	명령
기본	<code>installp -acXgd device ibmlb.base.rte</code>
관리(메시지)	<code>installp -acXgd device ibmlb.admin.rte</code> <code>ibmlb.msg.language.admin</code>
장치 드라이버	<code>installp -acXgd device ibmlb.lb.driver</code>
라이선스	<code>installp -acXgd device ibmlb.lb.license</code>
Load Balancer 컴포넌트(메시지). 포함: Dispatcher, CBR, Site Selector, Cisco CSS Controller 및 Nortel Alteon Controller	<code>installp -acXgd device ibmlb.component.rte</code> <code>ibmlb.msg.language.lb</code>
문서(메시지)	<code>installp -acXgd device ibmlb.doc.rte</code> <code>ibmlb.msg.en_US.lb</code>
Metric Server	<code>installp -acXgd device ibmlb.ms.rte</code>

여기서, *device*는 다음과 같습니다.

- CD에서 설치할 경우 */cdrom*
- 파일 시스템에서 설치할 경우 */dir*(파일 세트를 포함하는 디렉토리)

3. 요약의 결과 열에 설치할 Load Balancer의 각 파트마다 SUCCESS가 표시되어 있는지 확인하십시오(APPLYing). 설치할 모든 파트가 적용될 때까지 계속 진행하지 마십시오.

주: 모든 사용 가능한 메시지 카탈로그를 포함하여 지정된 장치에 파일 세트 목록을 생성하려면 다음 명령을 입력하십시오.

```
installp -ld device
```

CD에서 설치하는 경우, CD를 마운트 해제하려면 다음 명령을 입력하십시오.

```
umount /cdrom
```

다음 명령을 입력하여 제품이 설치되었는지 확인하십시오.

```
lsipp -h | grep ibmlb
```

전체 제품이 설치되고 나면 이 명령은 다음을 표시합니다.

```
ibmlb.base.rte
ibmlb.admin.rte
ibmlb.lb.driver
ibmlb.lb.license
ibmlb.component.rte
```


ibmlb.doc.rte
 ibmlb.ms.rte
 ibmlb.msg.language.admin
 ibmlb.msg.en_US.doc
 ibmlb.msg.language.lb

Load Balancer 설치 경로는 다음을 포함합니다.

- 관리 — /opt/ibm/edge/lb/admin
- Load Balancer 컴포넌트 — /opt/ibm/edge/lb/servers
- Metric Server — /opt/ibm/edge/lb/ms
- 문서(관리 안내서) — /opt/ibm/edge/lb/documentation

HP-UX에 설치

이 절에서는 제품 CD를 사용하여 Load Balancer를 HP-UX에 설치하는 방법을 설명합니다.

설치하기 전에

설치 프로시저를 시작하기 전에 소프트웨어를 설치하기 위한 루트 권한이 있는지 확인하십시오.

이전 버전이 설치되어 있으면 현재 버전을 설치하기 전에 해당 사본을 설치 제거해야 합니다. 먼저 실행 프로그램 및 서버 모두를 반드시 중지하십시오. 그리고 Load Balancer의 설치 제거를 수행하려면 73 페이지의 『패키지 설치 제거에 대한 지시사항』을 참조하십시오.

설치 프로시저

표 7에서는 Load Balancer에 대한 설치 패키지의 이름 및 시스템의 패키지 설치 도구를 사용하여 패키지를 설치하는 바람직한 순서를 나열합니다.

표 7. Load Balancer에 대한 HP-UX 패키지 설치 세부사항

패키지 설명	HP-UX 패키지 이름
기본	ibmlb.base
관리 및 메시지	ibmlb.admin ibmlb.nlv-lang
Load Balancer 라이선스	ibmlb.lic
Load Balancer 컴포넌트	ibmlb.component
문서	ibmlb.doc
Metric Server	ibmlb.ms

주:

1. *lang* 변수는 다음 언어 특정 코드 중 하나로 대체될 수 있습니다. de_DE, es_ES, fr_FR, it_IT, ja_JP, ko_KR, zh_CN, zh_TW.
2. *component* 변수는 disp(dispatcher), cbr(CBR), ss(Site Selector), cco(Cisco CSS Controller) 또는 nal(Nortel Alteon Controller) 중 하나의 대체를 나타냅니다.
3. 문서 패키지(ibmlb.doc)는 영어만을 포함합니다. 자국어로 된 Edge Components 문서 설정은 다음 웹 사이트에 있습니다. www.ibm.com/software/webservers/appserv/ecinfocenter.html.

HP-UX는 포르투갈 브라질어(pt_BR) 로케일을 지원하지 않습니다. HP-UX에서 지원되는 로케일은 다음과 같습니다.

- de_DE.iso88591
- en_US.iso88591
- es_ES.iso88591
- fr_FR.iso88591
- de_DE.iso88591
- it_IT.iso88591
- ja_JP.SJIS
- ko_KR.eucKR
- zh_CN.hp15CN
- zh_TW.big5

패키지 설치에 대한 지시사항

다음 프로시저는 이 작업을 완료하기 위해 필요한 단계를 자세히 설명합니다.

1. 로컬 슈퍼유저 루트로 변경하십시오.

```
su - root
Password: password
```

2. 설치 명령 실행을 통한 패키지 설치

설치 명령 실행

```
swinstall -s /source package_name
```

여기서, *source*는 패키지 위치에 대한 절대 디렉토리 경로이며 *package_name*은 패키지 이름입니다.

예를 들어, CD의 루트에서 설치하는 경우 다음은 Load Balancer(ibmlb.base)에 대한 기본 패키지를 설치합니다.

```
swinstall -s /source ibmlb.base
```

CD의 루트에서 설치하는 경우, Load Balancer에 대한 모든 패키지를 설치하려면 다음 명령을 실행하십시오.

```
swinstall -s /source ibmlb
```

3. Load Balancer 패키지의 설치 확인

swlist 명령을 실행하여 설치한 모든 패키지를 나열하십시오. 예를 들면, 다음과 같습니다.

```
swlist -l fileset ibmlb
```

패키지 설치 제거에 대한 지시사항

swremove 명령을 사용하여 패키지를 설치 제거하십시오. 패키지는 설치된 반대 순서대로 제거해야 합니다. 예를 들어, 다음을 실행하십시오.

- 모든 Load Balancer 패키지 설치 제거:

```
swremove ibmlb
```

개별 패키지(예: Cisco CSS Controller) 설치 제거:

```
swremove ibmlb.cco
```

Load Balancer 설치 경로는 다음을 포함합니다.

- 관리 — /opt/ibm/edge/lb/admin
- Load Balancer 컴포넌트 — /opt/ibm/edge/lb/servers
- Metric Server — /opt/ibm/edge/lb/ms
- 문서(관리 안내서) — /opt/ibm/edge/lb/documentation

Linux에 설치

이 절에서는 Load Balancer를 Edge Components CD를 사용하여 Linux에 설치하는 방법을 설명합니다.

설치하기 전에

Load Balancer를 설치하기 전에 다음을 확인하십시오.

- 루트로 로그인했는지 확인하십시오.
- 이전 제품 버전을 설치 제거했는지 확인하십시오. 설치 제거하려면, 모든 실행 프로그램과 모든 서버가 정지되었는지 확인하십시오. 그런 다음, 전체 제품을 설치 제거하려면 다음 명령을 입력하십시오.

```
rpm -e pkgname
```

설치 제거할 경우 패키지 설치에 사용한 순서와 반대로 수행하여 관리 패키지가 맨 마지막에 설치 제거되게 하십시오.

설치 단계

1. Edge Components 매체를 넣거나 웹 사이트에서 제품을 다운로드한 후 RPM(Red Hat Packaging Manager)을 사용하여 설치 이미지를 설치하십시오.

설치 이미지는 `lblinux-version.tar` 형식의 파일입니다.

2. 다음 명령을 입력하여 임시 디렉토리에 tar 파일을 해제하십시오.

```
tar -xf lblinux-version.tar
```

결과적으로 확장자가 .rpm인 파일 세트가 생성됩니다.

- `ibmlb-base-release-version.hardw.rpm`(기본)
- `ibmlb-admin-release-version.hardw.rpm` (Administration)
- `ibmlb-lic-release-version.hardw.rpm`(라이선스)
- `ibmlb-component-release-version.hardw.rpm`(LB 컴포넌트)
- `ibmlb-doc-release-version.hardw.rpm`(문서)
- `ibmlb-ms-release-version.hardw.rpm`(Metric Server)

여기서

- `release-version`은 현재 릴리스입니다(예: 6.1.0-0).
- `hardw`는 다음 값 중 하나입니다(`i386`, `ppc64`, `ppc`, `s390`, `s390x`, `x86_64`).
- `component`는 `disp`(Dispatcher 컴포넌트), `cbr`(CBR 컴포넌트), `ss`(Site Selector 컴포넌트), `cco`(Cisco CSS Controller), `nal`(Nortel Alteon Controller) 중 하나의 값입니다.

문서 패키지는 영어만을 포함합니다. 자국어로 된 Edge Component 문서 설정은 다음 웹 사이트에 있습니다.

www.ibm.com/software/webservers/appserv/ecinfocenter.html.

3. RPM 파일이 위치한 디렉토리에서 각 패키지를 설치하기 위한 명령을 실행하십시오. 예를 들면, 다음과 같습니다.

```
rpm -i package.rpm
```

Red Hat Linux 시스템의 경우: 알려진 Red Hat Linux 문제점으로 인해 `_db*` RPM 파일을 삭제해야 합니다. 그렇지 않으면 오류가 발생합니다.

다음 패키지 목록에 표시된 순서대로 각 컴포넌트에 필요한 패키지를 설치하는 것이 중요합니다.

- Base(base)
- Administration(admin)
- License(lic)
- Load Balancer 컴포넌트(ds, cbr, ss, cco, nal)

- Metric Server(ms)
- Documentation(doc)

주: 적어도 하나 이상의 RPM 파일이 Java™가 설치되고 RPM 데이터베이스에 등록되도록 요구합니다. Java가 설치되었으나 RPM 데이터베이스에 등록되지 않으면 다음과 같이 no dependencies 옵션과 함께 설치 명령을 사용하십시오.

```
rpm -i --nodeps package.rpm
```

4. 다음 명령을 입력하여 제품이 설치되었는지 확인하십시오.

```
rpm -qa | grep ibmlb
```

제품 전체를 설치하면 다음과 같은 출력이 생성됩니다.

- *ibmlb-base-release-version*
- *ibmlb-admin-release-version*
- *ibmlb-lic-release-version*
- *ibmlb-dsp-release-version*
- *ibmlb-cbr-release-version*
- *ibmlb-ss-release-version*
- *ibmlb-cco-release-version*
- *ibmlb-nal-release-version*
- *ibmlb-doc-release-version*
- *ibmlb-ms-release-version*

Load Balancer 설치 경로는 다음을 포함합니다.

- 관리 — /opt/ibm/edge/lb/admin
- Load Balancer 컴포넌트 — /opt/ibm/edge/lb/servers
- Metric Server — /opt/ibm/edge/lb/ms
- 문서 — /opt/ibm/edge/lb/documentation

패키지를 설치 제거해야 할 경우 패키지 설치에 사용한 순서와 반대로 수행하여 관리 패키지가 맨 마지막에 설치 제거되게 하십시오.

Solaris에 설치

이 절에서는 Edge Components CD를 사용하여 Solaris에 Load Balancer를 설치하는 방법에 대해 설명합니다.

설치하기 전에

설치 프로시저를 시작하기 전에 루트로 로그인했는지와 이전 제품 버전을 모두 설치 제거했는지 확인하십시오.

설치 제거하려면 모든 실행 프로그램 및 서버를 중지하십시오. 그런 다음, 다음 명령을 입력하십시오.

```
pkgrm pkgname
```

설치 단계

1. Load Balancer 소프트웨어가 들어 있는 CD-ROM을 해당 드라이브에 넣으십시오.
2. 명령 프롬프트에서 다음 명령을 입력하십시오.

```
pkgadd -d pathname
```

여기서, `-d pathname`은 패키지가 있는 하드 디스크의 디렉토리 또는 CD-ROM 드라이브의 장치 이름입니다(예: `-d /cdrom/cdrom0/`).

다음은 표시되는 패키지 목록이며 설치되어야 하는 순서(권장)대로 나열되었습니다.

- `ibmlbbase`(기본)
- `ibmlbadm`(관리)
- `ibmlblic`(라이센스)
- `ibmlbdisp`(Dispatcher 컴포넌트)
- `ibmlbcbr`(CBR 컴포넌트)
- `ibmlbss`(Site Selector 컴포넌트)
- `ibmlbcc`(Cisco CSS Controller 컴포넌트)
- `ibmlbnal`(Nortel Alteon Controller 컴포넌트)
- `ibmlbdoc` (문서)
- `ibmlbms`(Metric Server)

문서 패키지(`ibmlbdoc`)는 영어만을 포함합니다. 자국어로 된 Edge Component 문서 설정은 다음 웹 사이트에 있습니다. www.ibm.com/software/webservers/appserv/ecinfocenter.html.

모든 패키지를 설치하려면 `a11`을 입력한 후 리턴 키만 누르면 됩니다. 일부 컴포넌트만 설치하려면 설치할 패키지에 해당되는 이름을 공백이나 쉼표로 구분하여 입력한 후 리턴 키를 누르십시오. 기존 디렉토리나 파일의 권한을 변경하라는 프롬프트가 표시될 수 있습니다. 단순히 리턴을 누른 다음 Yes라고 응답하십시오. 설치는 알파벳순이며 전제조건순이 아니므로 전제조건 패키지를 설치해야 합니다. `a11`을 입력한 다음 모든 프롬프트에 대해 Yes라고 응답하면 설치가 완료됩니다.

문서 및 Metric Server 서버에 Dispatcher 컴포넌트만 설치하려는 경우, `ibmlbbase`, `ibmlbadm`, `ibmlblic`, `ibmlbdisp`, `ibmlbdoc` 및 `ibmlbms` 패키지를 설치해야 합니다.

3. 다음 명령을 입력하여 제품이 설치되었는지 확인하십시오.

```
pkginfo | grep ibm
```

Load Balancer 설치 경로는 다음을 포함합니다.

- 관리 — /opt/ibm/edge/lb/admin
- Load Balancer 컴포넌트 — /opt/ibm/edge/lb/servers
- Metric Server — /opt/ibm/edge/lb/ms
- 문서 — /opt/ibm/edge/lb/documentation

제 5 부 Edge Components를 사용하여 네트워크 빌드

제 5 부에서는 Edge Components를 사용하여 기본 설명 네트워크를 작성하기 위한 프로시저를 제공합니다. 이들 네트워크는 생산 환경에서 사용하고자 한 것이 아닙니다. 초기 네트워크 구성 프로세스는 제품을 처음 담당하는 관리자에게 여러 가지 네트워크 경계(edge-of-network) 개념을 분명히 알려줄 수 있습니다. 모든 컴포넌트 기능의 완전한 적용 범위와 자세한 구성 정보는 *Caching Proxy* 관리 안내서 및 *Load Balancer* 관리 안내서를 참조하십시오.

프로시저는 모든 노드에서 컴포넌트가 지원하는 모든 컴퓨터 시스템을 사용할 수 있게 합니다.

다음 장으로 구성되어 있습니다.

81 페이지의 제 15 장 『Caching Proxy 네트워크 작성』.

85 페이지의 제 16 장 『Load Balancer 네트워크 작성』.

제 15 장 Caching Proxy 네트워크 작성

그림 19는 세 개의 네트워크 노드에 위치하는 세 개의 컴퓨터 시스템을 사용하는 프록시 서버 네트워크를 표시합니다. 이 네트워크가 프록시 서버를 서버 2에 위치하는 전용 콘텐츠 호스트(IBM HTTP Server)에 바인드하면 프록시 서버는 호스트를 서브합니다. 워크스테이션과 서버 1 사이에 위치한 인터넷이 이를 시각적으로 나타냅니다.

중요사항: Caching Proxy는 다음 경우를 제외하고 모든 Edge Components 설치에 사용할 수 있습니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행되는 Edge Components 설치에 사용할 수 없습니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge Components 설치에 사용할 수 없습니다.

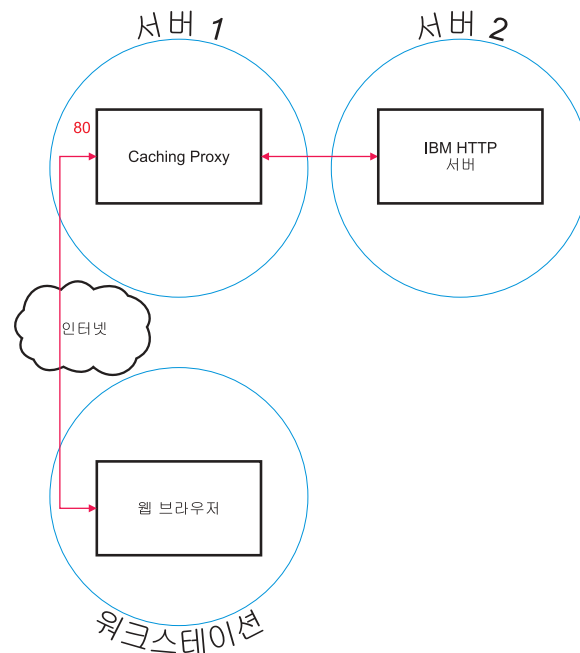


그림 19. Caching Proxy 데모 네트워크

Workflow

Caching Proxy 네트워크를 작성하려면 다음 순서대로 프로시저를 수행하십시오.

1. 필수 컴퓨터 시스템 및 소프트웨어 검토.
2. 빌드 서버 1(Linux 및 UNIX 시스템) 또는 서버 1 작성(Windows 시스템).
3. 서버 1 구성.
4. Caching Proxy 네트워크 검사.

필수 컴퓨터 시스템 및 소프트웨어 검토

다음 컴퓨터 시스템 및 소프트웨어 컴포넌트는 다음을 필요로 합니다.

- 서버 1로 작동하는 컴퓨터 시스템. 이 시스템은 인터넷에 액세스해야 합니다.
- 서버 2로 작동하는 컴퓨터 시스템. HTTP 서버가 콘텐츠 호스트에 설치되어 있어야 합니다.
- 워크스테이션으로 작동하는 컴퓨터 시스템. 웹 브라우저가 설치되어야 합니다.

빌드 서버 1(Linux 및 UNIX 시스템)

Caching Proxy 설치 및 구성은 다음과 같습니다.

1. 컴퓨터 서버가 모든 하드웨어 및 소프트웨어 요구사항을 만족시키는지 확인하십시오.
2. 슈퍼유저로 로그인 시 일반적으로 root를 입력하십시오.
3. Caching Proxy 컴포넌트를 설치하십시오.
4. 다음 명령을 입력하여 구성 및 관리 양식에 액세스할 수 있는 관리자 ID 및 암호를 작성하십시오.

```
# htadm -adduser /opt/ibm/edge/cp/server_root/protect/webadmin.passwd
```

프롬프트가 표시되면 **htadm** 프로그램에서 사용자 이름, 암호 및 관리자의 실제 이름을 입력합니다.

5. 83 페이지의 『서버 1 구성』을 계속하십시오.

서버 1 작성(Windows 시스템)

Caching Proxy 설치 및 구성은 다음과 같습니다.

1. Windows 2000 및 Windows 2003 운영 체제가 모든 하드웨어 및 소프트웨어 요구사항을 만족하는지 확인하십시오.
2. 관리자 권한을 가진 사용자로 로그인하십시오.
3. Caching Proxy 컴포넌트를 설치하십시오.

4. 다음 명령을 입력하여 구성 및 관리 양식에 액세스할 수 있는 관리자 ID 및 암호를 작성하십시오.

```
cd "Program Files\IBM\edge\cp\server_root\protect"  
htadm -adduser webadmin.passwd"
```

프롬프트가 표시되면 **htadm** 프로그램에 사용자 이름, 암호 및 관리자의 실제 이름을 입력합니다.

5. 『서버 1 구성』을 계속하십시오.

서버 1 구성

워크스테이션에서 다음을 수행하십시오.

1. 웹 브라우저를 시작하십시오.
2. 브라우저의 주소 필드에 `http://server_1`을 입력하십시오. 여기서, `server_1`은 실제 호스트 이름 또는 서버 1로 지정된 시스템의 IP 주소를 나타냅니다.
3. 구성 및 관리 양식을 누르십시오.
4. 사용자의 관리자 이름 및 암호를 입력하십시오. 구성 및 관리 양식이 사용자의 브라우저에 열립니다.
5. 서버 구성 -> 요청 처리 -> 요청 경로 지정을 누르십시오.
6. 앞에 삽입 단일 선택 단추와 기존의 와일드 카드 맵핑 규칙의 색인 값을 선택하여 기존의 와일드 카드 맵핑 규칙에 새 규칙을 삽입하십시오.
7. 드롭다운 상자에서 프록시를 선택하십시오.
8. URL 요청 템플릿 필드에 `/*`를 입력하십시오.
9. HTTP 요청을 서버 IP 주소 또는 호스트 이름 필드로 고쳐 보내려는 사이트의 호스트 이름을 입력하십시오. 이 값 앞에 `http://`를 입력하십시오.
10. 제출을 누르십시오.
11. 앞에 삽입 단일 선택 단추를 선택하고 6단계에서 작성한 맵핑 규칙의 색인값을 선택하여 구성 및 관리 양식에 액세스할 수 있는 맵핑 규칙을 만드십시오.
12. 조치 드롭다운 상자에서 전달을 선택하십시오.
13. URL 요청 템플릿 필드에 `/pub/*`를 입력하십시오.
14. 구성 및 관리 양식의 위치를 입력하십시오.
 - Caching Proxy가 Linux 또는 UNIX 시스템에 있을 경우, 서버 IP 주소 또는 호스트 이름 필드에 `/opt/ibm/edge/cp/server_root/pub/en_US/*`를 입력하십시오.
 - Caching Proxy가 Windows 시스템에 상주하는 경우, 서버 IP 주소 또는 호스트 이름 필드에 `"C:\Program Files\IBM\edge\cp\server_root\pub\en_US\"`를 입력하십시오.

15. 제출을 누르십시오.
16. 구성 양식의 맨 위의 서버 재시작 아이콘을 누르십시오.
17. 『Caching Proxy 네트워크 검사』를 계속하십시오.

Caching Proxy 네트워크 검사

워크스테이션에서 다음을 수행하십시오.

1. 웹 브라우저를 시작하십시오.
2. 사용자 브라우저의 주소 필드에 `http://server_1`을 입력하십시오. 서버 2에서 HTML 페이지 수는 전체 서버 1에 프록시되거나 웹 브라우저에 전달될 것입니다.
3. 구성 및 관리 양식에 액세스하려면 브라우저의 주소 필드에 `http://server_1/pub/`를 입력하십시오. 구성 및 관리 양식의 홈 페이지가 표시됩니다.

제 16 장 Load Balancer 네트워크 작성

그림 20에 두 개의 웹 서버 간에 웹 통신량을 로드 밸런싱하기 위해 Dispatcher 컴포넌트의 MAC 전달 메소드를 사용하는 로컬로 접속된 워크스테이션이 있는 기본 Load Balancer 네트워크를 보여줍니다. 기타 다른 TCP 또는 상태 없는 UDP 응용프로그램 통신량을 로드 밸런싱할 경우 구성은 유사합니다.

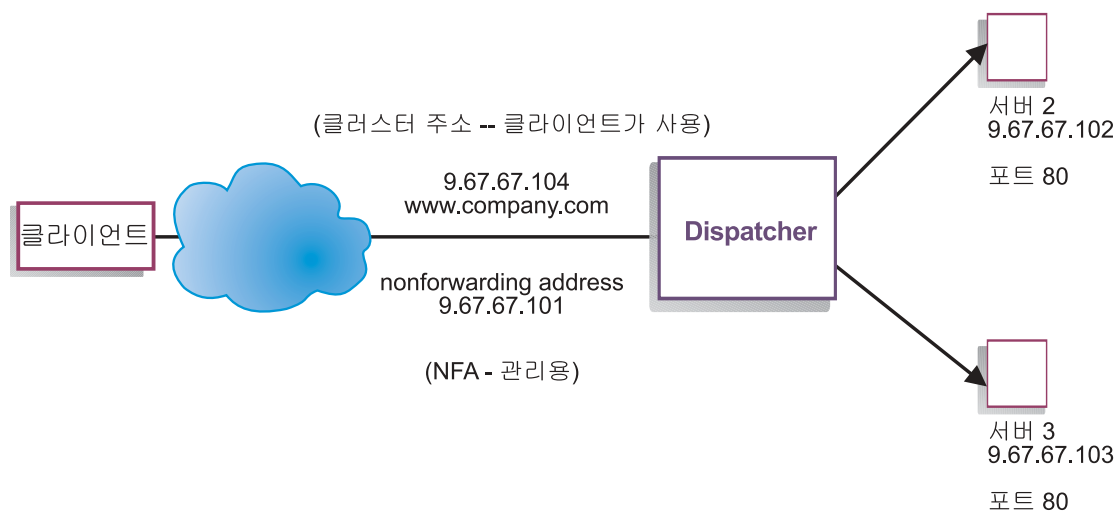


그림 20. Load Balancer 데모 네트워크

주: 이 구성은 Dispatcher(웹 서버 워크스테이션 중 하나에 위치)가 포함된 두 개의 워크스테이션만을 사용해서 완료될 수 있습니다. 이것은 결합 배치 구성을 나타냅니다.

Workflow

Load Balancer 네트워크를 작성하려면 다음 순서대로 프로시저를 수행하십시오.

1. 필수 컴퓨터 시스템 및 소프트웨어 검토.
2. 네트워크 구성.
3. Dispatcher 구성.
4. Load Balancer 네트워크 검사.

필수 컴퓨터 시스템 및 소프트웨어 검토

다음 컴퓨터 시스템 및 소프트웨어 컴포넌트는 다음을 필요로 합니다.

- Dispatcher로 작동하는 컴퓨터 시스템. 이 시스템은 하나의 실제 IP 주소와 로드 밸런스될 하나의 주소가 필요합니다.
- 웹 서버로 작동할 두 대의 컴퓨터 시스템. 웹 서버마다 하나의 IP 주소가 필요합니다.

네트워크 구성

1. 동일한 LAN 세그먼트에 있도록 워크스테이션을 설정하십시오. 세 시스템 간의 네트워크 통신량이 라우터 또는 브리지를 통과해서는 안됩니다.
2. 세 워크스테이션의 네트워크 어댑터를 구성하십시오. 이 예제에서는 다음과 같이 네트워크가 구성되어 있습니다.

워크스테이션	이름	IP 주소
1	server1.company.com	9.67.67.101
2	server2.company.com	9.67.67.102
3	server3.company.com	9.67.67.103
Netmask = 255.255.255.0		

각 워크스테이션에는 표준 이더넷 네트워크 인터페이스 카드가 하나만 있습니다.

3. server1.company.com이 server2.company.com과 server3.company.com을 둘 다 ping할 수 있어야 합니다.
4. server2.company.com과 server3.company.com이 server1.company.com을 ping할 수 있어야 합니다.
5. 두 개의 웹 서버(서버 2와 서버 3)에서 내용이 동일한지 확인하십시오. 이는 두 워크스테이션 모두의 데이터를 복제하고, NFS, AFS® 또는 DFS™와 같은 공유 파일 시스템을 사용하거나 사용자 사이트에 적절한 다른 방법으로 수행될 수 있습니다.
6. server2.company.com과 server3.company.com의 웹 서버가 작동하는지 확인하십시오. 웹 브라우저를 사용하여 <http://server2.company.com>과 <http://server3.company.com>에서 페이지를 직접 요청하십시오.
7. 이 LAN 세그먼트에 유효한 또 다른 IP 주소를 확보하십시오. 이 주소는 사용자 사이트에 액세스할 클라이언트에 제공할 주소입니다. 이 예제의 경우, 정보는 다음과 같습니다.

Name= www.company.com
IP=9.67.67.104

8. www.company.com의 통신을 승인하는 두 개의 웹 서버 워크스테이션을 구성하십시오.

www.company.com 별명을 server2.company.com 및 server3.company.com의 루프백 인터페이스에 추가하십시오.

- AIX의 경우:


```
ifconfig lo0 alias www.company.com netmask 255.255.255.0
```

- Solaris 7의 경우

```
ifconfig lo0:1 www.company.com 127.0.0.1 up
```

9. 루프백 인터페이스의 별명 지정으로 인해 작성되었을 수 있는 추가 라우트를 삭제하십시오.

두 대의 웹 서버 워크스테이션에 필요한 모든 구성 단계를 완료했습니다.

Dispatcher 구성

Dispatcher를 통해 명령행, 구성 마법사 또는 GUI를 사용하여 구성을 작성할 수 있습니다.

주: 매개변수 값은 영문자로 입력해야 합니다. 호스트 이름과 파일 이름의 매개변수 값만 예외입니다.

명령행을 사용한 구성

명령행을 사용 중이면, 다음 단계를 수행하십시오.

1. Dispatcher에서 dsserver를 시작하십시오.
 - AIX, HP-UX, Linux 또는 Solaris의 경우에는 루트 사용자로써 dsserver 명령을 실행하십시오.
 - Windows 플랫폼의 경우, dsserver는 자동으로 시작하는 서비스로서 실행됩니다.
2. Dispatcher의 실행 프로그램 기능을 시작하십시오.

```
dscontrol executor start
```
3. 클러스터 주소를 Dispatcher 구성에 추가하십시오.

```
dscontrol cluster add www.company.com
```
4. Http 프로토콜 포트를 Dispatcher 구성에 추가하십시오.

```
dscontrol port add www.company.com:80
```
5. 각 웹 서버를 Dispatcher 구성에 추가하십시오.

```
dscontrol server add www.company.com:80:server2.company.com
dscontrol server add www.company.com:80:server3.company.com
```
6. 클러스터 주소의 통신량을 승인하려면, 다음과 같이 워크스테이션을 구성하십시오.

```
dscontrol executor configure www.company.com
```
7. Dispatcher의 관리 프로그램 기능을 시작하십시오.

```
dscontrol manager start
```

Dispatcher는 서버 성능에 따라 로드 밸런스를 수행합니다.

8. Dispatcher의 어드바이저 기능을 시작하십시오.

```
dscontrol advisor start http 80
```

Dispatcher에서는 실패한 웹 서버로 클라이언트 요청이 전송되지 않았음을 확인합니다.

로컬로 연결된 서버의 기본 구성을 완료했습니다.

중요사항: **IPv4 및 IPv6용 Load Balancer** 설치에서 Dispatcher 명령(dscontrol)의 구문은 한 가지 중요한 사항을 제외하고 동일합니다. dscontrol 명령의 분리문자가 콜론(:) 대신에 골뱅이(@) 기호입니다. (IPv6 형식이 주소 지정 설계에서 콜론을 사용하므로 콜론 이외의 분리문자를 정의해야 했습니다.)

예(이전 Dispatcher 구성 예)

- IPv4 및 IPv6용 Load Balancer 설치 시 HTTP 프로토콜 포트를 Dispatcher 구성에 추가하려면 다음을 수행하십시오.

```
dscontrol port add www.company.com@80
```

- IPv4 및 IPv6용 Load Balancer 설치 시 각 웹 서버를 Dispatcher 구성에 추가하려면 다음을 수행하십시오.

```
dscontrol server add www.company.com@80@server2.company.com
```

```
dscontrol server add www.company.com@80@server3.company.com
```

IPv4 및 IPv6용 Load Balancer 설치를 사용하는 경우, 자세한 정보는 *WebSphere Application Server Load Balancer 관리 안내서*에서 제한사항 및 구성 차이점에 대한 정보가 포함되어 있는 IPv4 및 IPv6용 Load Balancer의 Dispatcher 전개 장을 참조하십시오.

구성 마법사를 사용한 구성

구성 마법사를 사용 중이면 다음 단계를 수행하십시오.

1. Dispatcher에서 dsserver를 시작하십시오.

- AIX, HP-UX, Linux 또는 Solaris의 경우에는 다음 명령을 루트 사용자로서 실행하십시오.

```
dsserver
```

- Windows 시스템의 경우, dsserver는 자동으로 시작하는 서비스로서 실행됩니다.

2. Dispatcher의 마법사 기능, dswizard를 시작하십시오.

마법사는 Dispatcher 컴포넌트의 기본 구성 작성 프로세스를 단계별로 안내합니다. 여기서는 네트워크에 대한 질문을 한 다음 Dispatcher가 서버 그룹에 대한 통신량 로드 밸런스를 유지하도록 클러스터 설정 단계를 안내합니다.

구성 마법사는 다음 패널을 포함합니다.

- 마법사 소개

- 발생 내용
- 설치 준비
- 구성할 호스트 선택(필요한 경우)
- 클러스터 정의
- 포트 추가
- 서버 추가
- 어드바이저 시작
- 서버 시스템 설치

GUI를 사용한 구성

GUI를 시작하려면 다음 단계를 따르십시오.

1. dsserver 프로세스가 실행 중인지 확인하십시오.
 - AIX, HP-UX, Linux 또는 Solaris의 경우, 다음 명령을 루트로서 실행하십시오.
dsserver
 - Windows 시스템의 경우, dsserver는 자동으로 시작하는 서비스로서 실행됩니다.
2. 그런 다음, 다음 중 하나를 수행하십시오.
 - AIX, HP-UX, Linux 또는 Solaris의 경우에는 lbadmin을 입력하십시오.
 - Windows 시스템의 경우, 시작 > 프로그램 > **IBM WebSphere > Edge Components > IBM Load Balancer > Load Balancer**를 누르십시오.

Load Balancer 네트워크 검사

1. 웹 브라우저에서 <http://www.company.com>으로 이동하여 페이지가 표시되는지 확인하십시오.
2. 웹 브라우저에서 페이지를 다시 로드하십시오.
3. dscontrol server report www.company.com:80: 명령을 실행하십시오. 두 서버의 총 연결 컬럼이 두 개까지 추가되는지 확인하십시오.

주의사항

초판(2006년 5월)

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서는 이 자료에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다.

IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 “현상태대로” 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 이 변경사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들어, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 있는 프로그램 및 이 프로그램에 대해 사용 가능한 모든 라이선스가 있는 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 반드시 테스트하지 않았으므로, 이들 제품과 관련된 성능의 정확성, 호환성 또는 기타 주장에 대해서는 확인할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

이 정보를 소프트웨어로 확인하는 경우, 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

상표

다음 용어는 미국 또는 기타 국가에서 사용되는 IBM Corporation의 상표입니다.

- AFS
- AIX
- DFS
- IBM
- iSeries™
- RS/6000®
- SecureWay®
- Tivoli
- ViaVoice
- WebSphere

Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

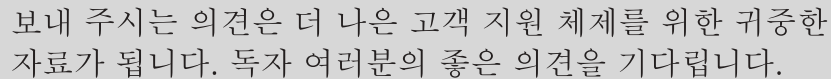
Intel, Intel Inside(로고), MMX 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.

GA30-2919-00



☺ 설문에 답해 주셔서 감사합니다.
귀하의 의견은 저희에게 매우 소중한 것이며, 고객 여러분들께 보다 좋은 제품을 제공해
드리기 위해 최선을 다하겠습니다.



GA30-2919-00



Spine information:



**WebSphere Application
Server**

Edge Components용 개념, 계획 및 설치 버전 6.1

GA30-2919-00