

# WebSphere Extended Deployment (WebSphere XD)

*확장성 증대, 사용가능성 향상,  
및 자율 컴퓨팅 적용 가능성 제공*



## 소개

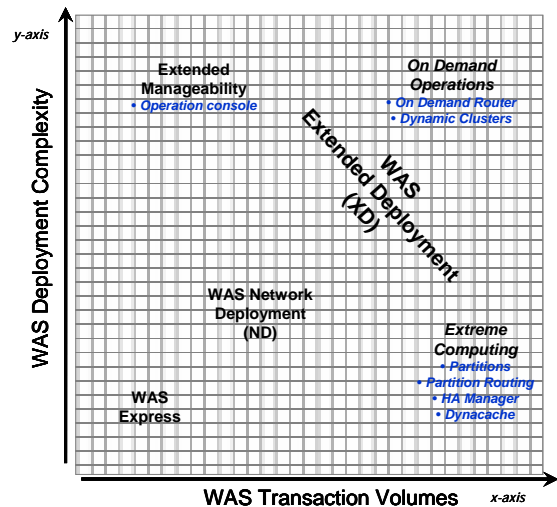
WebSphere Extended Deployment(WebSphere XD)는 WebSphere 플랫폼을 위해 확장성, 사용가능성 및 적응성 강화에 초점을 맞춘, 확장된 QoS(Quality of Service)를 제공합니다. WebSphere XD 는 WebSphere Application Server 를 기본 엔진으로 한 제품 군이 On Demand 환경으로 들어가도록 하는 선구적인 제품으로서, 그리드 컴퓨팅 기술과 자율 컴퓨팅 기술을 지원하는 J2EE 환경을 제공합니다.

WebSphere XD 는 QoS 를 향상하기 위해서 IBM 메인 프레임 환경에서 전통적으로 사용되어온 기능을 분산 환경에 적용하였습니다. 운영 정책, 목표(예를 들면, 응답 속도, TPS 및 CPU 사용량 등) 기반의 워크로드 관리, 서버 프로세스 자원의 자율 관리(필요한 어플리케이션에 서버 자원을 자율 할당)를 통한 동적 운영환경(Dynamic Operation)을 제공하고 빠른 응답속도를 위한 어플리케이션 파티셔닝과 직관적인 강화된 관리기능 등을 제공합니다.

WebSphere XD 는 트랜잭션 처리 용량과 복잡한 운영 환경에서의 관리 능력을 향상합니다. 추가적으로 WebSphere XD 는 향상된 그래픽 인터페이스와 반복 가능하고 자동화 가능한 관리 기능 및 시스템 구성을 위한 새로운 개념 등을 제공함으로써 사용 편의성을 증대하였습니다.

WebSphere XD 는 WebSphere Application Server Network Deployment Edition (WebSphere ND) 위에서 설치됩니다. WebSphere XD 와 WebSphere ND 의 관계는 아래 그림에서 두 축에서 QoS(Quality of Service)의 확장으로서 표시됩니다. 그림에서 y 축은 운영환경의 복잡성이 증가하는 것이고 x 축은 트랜잭션 볼륨이 증가하는 것을 나타냅니다. WebSphere XD 의 On Demand 기능은 이러한 두 가지 요소가 증가하는 환경에서 적절히 대응할 수 있는 기반 기능을 제공합니다.. WebSphere XD 는 WAS 운영환경에서 확장성, 중단 없는 서비스 제공 및 WAS 운영 서버 자원의 사용성(Utilization)을 최대화 합니다. 이러한 강화된 기능은 다음과 같은 고객에게 제공됩니다.

- **On Demand 운영환경(On Demand Operating Environment, ODOE)을 구축함으로써 운영 효율성을 높이려는 고객 -- ODOE의 핵심에는 전체 시스템 확장성, 활용성 및 사용가능성을 증가시키는 가상화된 자율 컴퓨팅을 지원하는 J2EE 환경이 있습니다.** WebSphere XD는 어플리케이션과 비즈니스 목표(응답속도, CUP 사용률 등)를 달성하기 위해 각각의 서비스를 그룹화 하고 각각의 그룹의 비즈니스 목표를 달성하기 위해 서버 자원을 할당함으로써 효율성을 높입니다.
- **선형적 확장성과 99.999% 가용성을 요구하는 아주 대용량 트랜잭션 워크로드를 가지는 서비스를 제공할 필요가 있는 고객(예를 들어 주식거래, 경매, 온라인 게임 또는 재정 분석 등) – 높은 성능과 대용량 트랜잭션을 처리해야 하는 환경에서 성능, 확장성 및 특정 J2EE 어플리케이션의 탄력성을 향상할 수 있는 새로운 디자인 패턴과 기능을 제공합니다.** 디자인 패턴은 파티셔닝, 캐싱, 파티션 인식 워크로드 관리, HA 관리 및 자율적 파티션 관리에 초점을 맞추고 있습니다. .
- **대규모 자원의 관리와 계속적으로 사용 가능한 WAS환경에 관심을 가지는 고객(수백 대의 서버를 가지는 대규모 고객) – WebSphere XD의 구성과 시각화 기능을 제공하는 확장된 관리 콘솔이 제공됩니다.** WebSphere XD 콘솔은 차트 제공, 실시간 통계 트리 및 혁신적인 트리맵 뷰를 포함하는 다양한 형식으로 구성 상태의 포괄적인 그림을 제공합니다. 이러한 다양한 뷰는 사용자가 WebSphere XD의 자율 상태하에서 어떤 일이 일어나는지 보고 이해할 수 있도록 단순하고 통합된 직관적인 그래픽 인터페이스를 제공합니다. 또한 WAS서버의 구성에 대한 권고사항을 제공하여서 최적의 운영환경을 유지할 수 있도록 하는 자율 컴퓨팅 기능을 제공합니다.



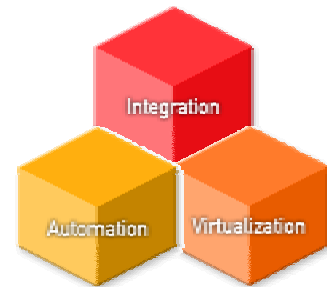
WebSphere XD 의 비전은 WebSphere Application Server 를 기본 엔진으로 하고 있는 모든 제품뿐 만 아니고 타사의 WAS 제품에 있어서 On Demand 환경을 구축할 수 있도록 하는 것입니다.

WebSphere XD 는 각각 엔터프라이즈 프로비저닝과 워크로드 관리를 위한 Tivoli Intelligent Orchestrator/Tivoli Provisioning Manager (TIO/TPM), 및 Enterprise Workload Manager (eWLM) 같은 IBM On Demand 포트폴리오의 제품들과 결합하여 더욱 강력한 기능을 제공합니다.

이 문서는 먼저 WebSphere XD 의 On Demand 인프라스트럭처를 소개하고 계속해서 WebSphere XD 에 의해 제공되는 기능을 설명할 것입니다. 다음으로 고성능, 대용량 트랜잭션 을 처리하는 방식으로서 WebSphere Partition Facility 를 설명할 것입니다. 마지막으로 확장된 시각화 및 구성 패널을 포함하는 WebSphere XD 의 사용자 인터페이스에 대한 설명을 할 것입니다.

## 1. On Demand 기초

On Demand 영역에서의 IBM 의 비전은 통합, 가상화 및 자율 컴퓨팅의 3 가지 입니다. 통합은 WebSphere 에 있어서 새로운 것은 아니고 개념을 압축한 것입니다. WebSphere 는 Back-end 데이터 저장소와 통합하는 미들웨어 솔루션으로서 자리 매김 해 왔고 Web Service, 프로세스 관리 및 메시징 기술을 사용한 엔터프라이즈 서비스 통합을 제공합니다. WebSphere XD 는 J2EE 자원과 어플리케이션 및 그들 사이에서, 서로간의 관계의 전통적인 개념을 확장하는 가상화 인프라스트럭처 위에서 구축됩니다. 이러한 새로운 인프라스트럭처는 WebSphere XD 의 운영을 최적화되고 반복적인 방식으로 자동화하는 능력을 가속화합니다. 자율성의 소개와 함께, WebSphere XD 는 더 안정적이고 더 예측가능하고 더 신뢰성 있는 운영 환경을 제공함으로써 총 소유비용 (TCO)을 절감하도록 합니다. 이 섹션의 나머지는 WebSphere XD 에 포함된 가상화 인프라스트럭처에 대해서 이야기할 것입니다.



### 1.1. 가상화 인프라스트럭처

#### 1.1.1. 가상화 인프라스트럭처는 무엇인가?

WebSphere XD 는 전통적 J2EE 구조 사이의 관계를 재정의합니다. 어플리케이션을 직접 서버에 배포하는 대신에, WebSphere XD 관리자는 어플리케이션을 자원 풀에 매핑합니다. 이 어플리케이션은 구성된 비즈니스 목표에 따라 자원 풀 안에 있는 서버의 서브세트에 배포됩니다. WebSphere XD 의 가상화 인프라스트럭처는 노드 그룹(자원 풀을 대표함)과 동적 클러스터의 두 가지 새로운 개념을 가지고 있습니다.

#### 1.1.2. 노드 그룹

WebSphere XD 에서, 어플리케이션과 그들이 실행될 수 있는 노드의 관계는 “노드그룹”으로 불리우는 중개 개념의 용어로 표현됩니다. 구체적인 용어로, 노드그룹은 WAS 가 운영되는 물리적인 기계의 집합입니다. 더 추상적인 의미로 살펴본다면, 노드그룹은 몇 가지 공통 기능과 자산(예를 들어, 주어진 네트워크 상에서의 연결, 특정 타입의 데이터베이스와 연결 등)을 가지는 기계의 풀입니다. 노드 그룹의 이러한 특징은 명시적으로 정의된 것이 아닙니다. 노드그룹의 속성들은 WebSphere XD 디자인에 포함되어 있습니다.

노드그룹은 하나 또는 여러 개의 동적 클러스터가 만들어진 컴퓨팅 파워의 풀을 나타냅니다. 이러한 자원의 분산은 워크로드 패턴의 변화를 위해 조정되는 비즈니스 목표에 따라 자동적으로 조정됩니다.

노드그룹의 공통 기능과 자산은 어플리케이션에 의해 요구된다고 간주됩니다. 그래서 노드그룹은 주어진 어플리케이션을 수행할 수 있는 기계들의 집합으로 기술됩니다. 주어진 노드 풀에서 참가하는 것에 의해 포함되는 것을 이해하고 어플리케이션이 수용될 풀에 배포하는 것을 확인하는 것이 관리자의 임무입니다. 주어진 풀에서의 자원은 워크로드와 정책에 따라 동적으로 할당됩니다.(섹션 3 을 보십시오.).

#### 1.1.3. 동적 클러스터

어플리케이션을 WebSphere XD 에 배포하는 프로세스는 어플리케이션의 요구사항을 충족하는 노드그룹을 선택/정의하고 이어서 동적 클러스터를 생성함으로써 시작합니다. 동적 클러스터는 정적인 WebSphere ND 에서 정적 카운터 파트인 정적인

클러스터를 확장하는 컨테이너 개념입니다. 동적 클러스터는 단일 노드그룹에서 구성됩니다. 어플리케이션(.ear 파일)은 그 후 WebSphere ND 의 정적 클러스터에 배포되는 것과 동일한 방식으로 동적 클러스터에 배포됩니다. 그러나 WebSphere XD 는 자신의 상위 노드그룹 내에서 동적 클러스터의 자율적인 확장과 축소를 가능하게 합니다. 그래서 어플리케이션을 위한 서버 자원의 수요에 있어서 주기적인 과부하는, 요청을 처리하기 위한 어플리케이션을 위한 서버 자원 증가를 가져옵니다. 이러한 자원의 증가를 위한 전략은 운영 정책(각각의 어플리케이션 그룹이 가지는 비즈니스 목표)에 따라 운영됩니다.

## 1.2. 운영 정책

### 1.2.1. 운영 정책은 무엇인가?

운영정책은 명시적으로 구성이 가능한 명령이고 WebSphere 운영 시 적용되는 운영 지침입니다. 운영 정책을 통해서, WebSphere XD 는 서로 다른 요청 형태의 비즈니스 중요도를 표시하는 기능을 제공합니다. 이 기능은 비즈니스 목표를 달성하지 못하는 동안에 정의된 방식으로 목표를 달성하는 것을 가능하게 합니다. WebSphere XD 의 운영 정책 지원은 두 개의 새로운 개념을 통해서 나타냅니다.: *트랜잭션 클래스 & 서비스 클래스*

### 1.2.2. 서비스 클래스 & 트랜잭션 클래스

트랜잭션 클래스는 몇몇 논리적인 “요청 형태(Request Type)”를 정의합니다. 트랜잭션 클래스는 URI 필터의 목록으로서 표현됩니다. 예를 들면, 관리자는 기업에서 모든 “구매하는 것”을 나타내는 “구매”라는 이름이 붙은 트랜잭션 클래스를 정의할 수 있습니다. 관리자는 트랜잭션 클래스를 서비스 클래스로 매핑함으로써 비즈니스 중요도를 트랜잭션 클래스에 할당합니다.

서비스 클래스는 “목표 성능”과 “비즈니스 중요도 평가”로 구성됩니다. 서버 자원의 수요가 적을 때에, 운영 정책은 중요하게 보이지 않습니다. 왜냐하면 모든 요청은 즉시 서비스와 될 수 있기 때문입니다. 즉 모든 요청이 즉시 응답될 수 있기 때문입니다. 컴퓨팅 파워를 초과하는 요청이 있을 때 성능 목표는 중요해집니다. 이러한 일이 발생하면, 미션 크리티컬한 요청이 높은 QoS 를 유지하고 있는 동안(중요도가 높은 요청의 비즈니스 목표를 충족하고 있는 동안), 중요도가 낮은 서비스 클래스는 성능 목표를 달성하지 못하기 시작하고 점진적이고 성능이 제어된 상태로 저하되기 시작합니다.



URI 를 직접 서비스 클래스로 매핑함으로써 요청을 서비스 클래스로 분류합니다. 그러나 이것은 통계가 서비스 클래스 또는 URI 범위에서만 사용 가능하기 때문에 리포팅과 시각화 분야에서 문제를 가져올 수 있습니다. 트랜잭션 클래스는 각각의 엔터프라이즈에 특화된 요청의 어떤 논리적 그룹화를 나타내는 리포팅 통계를 위한 편리한 범위입니다. 이러한 관점에서 WebSphere XD 는 사용자 요구에 맞추어 수정할 수 있는 시각화 기능과 성능 통계 리포팅을 제공합니다. 요청을 트랜잭션 클래스로 분류하는 것은 On Demand Router 에 의해 서비스 클래스에 따라 요청을 큐에 저장함으로써 수행됩니다. On Demand Router 섹션에서 이 프로세스에 대해 더 자세히 설명합니다.

### 1.2.3. 성능 목표의 형태

WebSphere XD 는 현재 두 가지 형태의 성능 목표를 제공합니다. : **평균 응답 시간**과 **임의의 목표**. 평균 응답 시간 목표는 완전히 자기-해석 적입니다. 이 목표는 가장 높은 평균 응답 시간을 나타내는 특정 시간 숫자를 요구합니다. 어플리케이션의 계산된 평균 응답 시간이 서비스 클래스의 목표 응답 시간을 초과하면, 자을 관리자는 문제를 완화하는 행동을 취합니다. 임의의 목표 성능 목표는 WebSphere XD 실행환경으로부터 요청 서비스로 "최대 작업"이라고 단순히 지시하면 됩니다. 그러나 절대적인 응답 시간 목표는 이러한 형태의 성능 목표에 포함되지 않습니다.

### 1.2.4. 조기 경보 메커니즘으로서의 성능 목표

초과 사용, 부족한 컴퓨팅 자원 또는 문제 상황 같은 원치 않는 많은 상황들은 비즈니스 성능 목표를 달성하지 못하게 할 수 도 있습니다. 이러한 모든 원인은 WebSphere 운영에 있어서 응답 속도가 느려지는 것 같은 현상에서 동등하게 고려되어서 이러한 상황을 바로잡기 위해 조치를 취해야 합니다. 이러한 관점에서 WebSphere XD 의 운영 정책 기능은 조기 경보 메커니즘으로서 행동합니다. 관리자가 요청 트랙이 문제를 야기시킬 정도로 많지 않다고 인식한다면, 관리자는 에러 조건, 메모리 부족 또는 다른 문제 상황을 조사하도록 합니다.

## 2. 자동화된 엔터프라이즈 오케스트레이션

이전 섹션에서는 On Demand 기능을 위한 기반으로 WebSphere XD 의 가상화 인프라스트럭처에 대하여 소개하였습니다. 이 섹션에서는 인프라스트럭처에 의해 사용 가능한 기능-자동화된 엔터프라이즈 오케스트레이션에 대하여 소개하겠습니다. 자동화는 On Demand Router(ODR), Tivoli Intelligent Orchestrator/Tivoli Provisioning Manager(TIO/TPM) 및 몇몇 자을 관리자 같은 새로운 구성 요소를 통해 제공됩니다. ODR 은 WebSphere XD 로의 작업 흐름을 제어하고 TIO/TPM 은 자원 풀 사이의 자원 프로비저닝을 관리합니다. 그리고 자을 관리자는 서버 인스턴스로의 자동화된 어플리케이션 배포를 수행합니다. 이러한 새로운 구성요소들은 아래 자세히 설명됩니다.

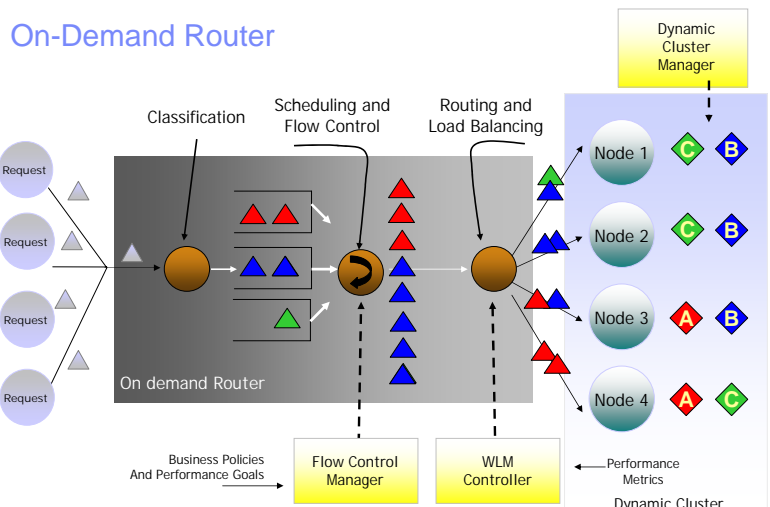
### 2.1. ODR(On Demand Router)

#### 2.1.1. ODR(On Demand Router)은 무엇인가?

ODR(On Demand Router)는 WebSphere XD 엔터프라이즈 앞 단에 위치한 컴포넌트 입니다. 이것은 Backend 로의 요청 흐름을 제어하는 WebSphere XD 구성의 입구에 자리 잡습니다. 특히 ODR 은 운영 정책에 따라 요청을 큐에 넣는 것과 분배하는 것을 조정하는 기능을 가지고 있습니다. 큐 길이와 분배 율의 최적화에 있어서 요청의 동시성(노드 그룹당), 운영 정책, 서비스 클래스 가중치 및 로드 밸런싱 같은 몇 가지 요소가 고려됩니다. ODR 아키텍처 개요도는 아래 있습니다.

#### 2.1.2. 요청의 분류

ODR 의 목적은 구성된 비즈니스 목표를 반영하는 지능적인 방식으로, WebSphere XD 뒷 부분으로 보내기 위해 "원본" 요청을 수신 네트워크와 프록시로부터 받아들이는 것입니다. 이 컴포넌트는 각 요청자의 상대적 비즈니스 중요도를 비교하는 것 같이 들어오는 요청의 특징에 따라 다릅니다. 이렇게 구분되는 것은, 요청이 "요청 타입" 또는 트랜잭션 클래스의 한정된 세트로 분류되는 것으로부터 시작합니다. 운영 정책 섹션에서 설명한 것 같이 트랜잭션 클래스는 URI 리스트와 일치합니다. 그래서 요청의 분류는 이러한 URI 리스트의 세트에 매칭되는 패턴과 일치합니다.



### 2.1.3. 클래스 기반 작업 큐잉

요청이 트랜잭션 클래스에 매핑되면 트랜잭션 클래스의 서비스 클래스에 링크됩니다. 그 후 요청은 그 서비스 클래스와 일치하는 큐에 보내집니다. 각각의 서비스 클래스는 서로 다른 큐를 가집니다. 자을 관리자는 도착 율과 서비스 시간의 평가에 따라 비즈니스 목표를 달성하기 위해서 때때로 각 큐의 분배 가중치를 조정하기도 합니다.

### 2.1.4. 동시성 제한

예상되는 과부하를 막기 위해서, ODR 은 WebSphere XD 뒷 부분으로 동시에 서비스 되는 요청의 수를 제한합니다. 동시성 제한은 구성된 최대 쓰레드 풀 사이즈에 따라 계산됩니다. 이러한 한계에 접근하면, 요청은 요구가 낮아질 때까지 ODR 의 큐에 쌓이게 됩니다. 만약 큐가 구성 가능한 최대 길이에 도달하게 되면 추가적인 요청에 대해서는, 서버가 너무 바빠서 들어오는 요청을 처리할 수 없다는 응답 메시지를 보내게 될 것입니다.

## 2.2. 자을 관리자

---

WebSphere XD 의 자을 기능은 자을 관리자로 알려진 컴포넌트 세트를 통해 제공됩니다. 이러한 컴포넌트는 여러 개의 센서를 통해서 성능과 건강도 통계를 모니터링 하고 시스템 성능을 최적화 하기 위해서 다양한 내부 제어 스위치를 켭니다, 이 섹션에서는 WebSphere XD 의 구성에 있는 각각의 자을 관리자를 보도록 하겠습니다.

### 2.2.1. 흐름 제어 관리자(Flow Control Manager)

흐름제어 관리자(Flow Control Manager)는 ODR 에 있는 큐 세트로부터의 요청 분배를 감시합니다. 흐름제어 관리자는 가중치 라운드 로빈 스케줄링 알고리즘을 사용합니다. 흐름 제어 관리자는 비즈니스 목표를 달성하기 위한 흐름 제어를 할당하기 위해 큐의 가중치를 조정합니다. 흐름 제어 관리자에 의해 작업이 분배되면, 작업은 동적 워크로드 관리자(Dynamic Workload Manager)로 전달 됩니다.

### 2.2.2. 동적 워크로드 관리자(Dynamic WorkLoad Manager, dWLM)

동적 워크로드 관리자는 전체 엔터프라이즈 Back-end 의 작업 로드 밸런싱을 다룹니다. 동적 워크로드 관리자는 작업이 전달되는 서버 테이블을 관리합니다. 이 테이블에서, 작업을 수행하는 상대적 능력과 일치하는 가중치가 각 서버에 동적으로 할당됩니다. WebSphere XD 용어로, dWML 은 각각의 동적 클러스터 위한 액티브 서버 인스턴스 리스트를 관리하고 각각 서버에 관찰된 성능 추이에 따라 라우팅 가중치를 할당합니다. 다음에 동적 클러스터 내의 노드들의 워크로드를 조정하기 위해서 요청은 후보 서버 인스턴스로 라우팅 됩니다.

### 2.2.3. 동적 클러스터 관리자(Dynamic Cluster Manager)

동적 클러스터 관리자는 어느 때든 하나의 노드그룹에서 단 하나의 인스턴스만을 가지는 WebSphere XD 구성 상의 각각의 노드 에이전트 프로세스에 존재합니다. 동적 클러스터 관리자의 주요 임무는 노드 그룹 안의 어플리케이션을 관리하는 것입니다. 특히 동적 클러스터 관리자는 주어진 동적 클러스터에서 액티브 인스턴스의 위치와 중요성을 지시합니다. 주어진 동적 클러스터를 위한 요청이 증가/감소함에 따라 동적 클러스터는 운영 정책에 의해 확장/축소됩니다. 동적 클러스터는 비어 있는 세트를 제외하고(비록 이러한 특별 케이스가 어플리케이션의 "Swapping out"을 가능하게 하는 기능이 차후 버전에서 지원되지만) 노드그룹에서 어떠한 노드 서브세트에도 구성될 수 있습니다.

## 2.3. TIO (Tivoli Intelligent Orchestrator) / TPM (Tivoli Provisioning Manager)

---

WebSphere XD 엔터프라이즈에서 요청 분배가 노드 그룹 사이에 옮겨지면, 보상하는 능력이 WebSphere XD 자을관리자의 영역을 넘어서 확장됩니다. TIO(Tivoli Intelligent Orchestrator)/TPM(Tivoli Provisioning Manager)을 사용하여 WebSphere XD 구성은 노드그룹 외부로부터 추가적인 물리적 컴퓨팅 자원을 확보할 수 있습니다.

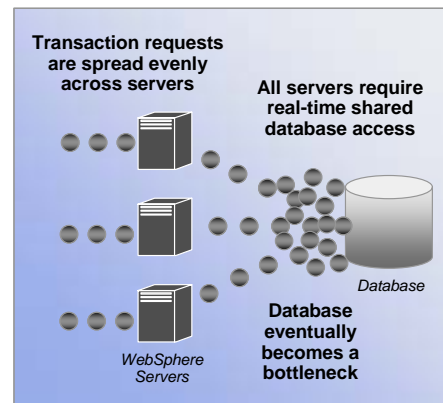
TIO/TPM 은 프로비저닝을 위한 IBM 의 표준 툴입니다. 일반적인 경우에, 비즈니스 목표에 따라 할당된 프로세싱 파워(기계, LPAR, CPU 형태로)을 보장하기 위해 TIO/TPM 은 이 기종 자원 풀 세트(예를 들면 WAS, MQSeries, SAP, e-Mail 등)를 모니터 하고 관리합니다. WebSphere XD 의 프로비저닝 엔진으로서 특화된 어플리케이션에서 TIO/TPM 은 WebSphere XD 구성의 범위 안에서 사용 가능한 자원의 최적화 할당을 계산하는데 사용되는 정보를 제공합니다. TIO/TPM 은 다음에 후보 워크로드 상이에 자원 분산을 조정합니다.

### 3. WPF(WebSphere Partitioning Facility)

WPF(WebSphere Partitioning Facility)는 J2EE 어플리케이션의 높은 성능, 확장성, 가용성에 있어서 더 높은 수준을 달성하는데 필요한 필수 기능을 제공합니다. 지난 5년간의 WebSphere Application Server 개발을 통해서 성능을 높이는 분야에서 분명하게 검증된 디자인 패턴의 시리즈가 발견되었습니다. 이러한 디자인 패턴들이 인식되고 문서화 됨에 따라 고객 어플리케이션에 이러한 패턴의 적용을 공식화 하는 새로운 기능들이 WebSphere XD 안에 포함되었습니다. WebSphere XD 팀이 초점을 맞춘 첫 번째 디자인 패턴은 WPF 를 사용하여 수행되는 어플리케이션에 특화된 파티션 개념을 포함하는 *파티셔닝 패턴*입니다. WPF 를 활용한 초기 사용자의 경우에는 전체 시스템 성능이 20 배 향상되었습니다. 다음 섹션에서는 파티셔닝 패턴과 어떻게 WebSphere XD 가 WPF 기능을 통해서 이 패턴을 사용하는 지를 설명합니다.

#### 3.1. 파티셔닝 패턴

파티셔닝 패턴은 데이터베이스에 대하여 집중적으로 데이터를 읽고 쓰는 대용량 OLTP 어플리케이션에서 발생하는 병목현상을 막아줍니다. 이러한 시스템의 예는 주식거래, 인터넷 뱅킹, 예약 및 온라인 경매 시스템 등을 들 수 있습니다. 오늘날 J2EE 서버는 Back-end 데이터베이스 시스템의 부하를 줄이는데 사용되는 데이터 캐싱 기능을 사용하는 상거래 시스템처럼 대부분 데이터를 “읽는” 환경으로 최적화 되었습니다. 그러나 시스템이 데이터를 업데이트하는 것이 증가하는 것을 발견하면(예를 들어 데이터베이스 삽입, 삭제, 생성 및 업데이트), 이러한 캐싱 시스템은 캐시와 데이터베이스 사이의 일관성을 유지하기 위해 적극적으로 워크로드를 낮추는 작업을 시작합니다. 이러한 구성은 빠르게 응답 시간을 줄이는 시점에 도달하고, 모든 트래픽을 데이터 베이스로 전송하고 데이터베이스가 데이터 일관성을 관리하도록 합니다. 이러한 전략은 대규모 SMP 기계 위에서 실행되는 비용이 많이 드는 대규모 데이터베이스 구성을 필요로 합니다. 초대용량 환경에서, 데이터베이스 서버는 필연적으로 비용과 성능의 병목이 됩니다.



파티셔닝 패턴은 어플리케이션 서버가 버퍼로서 행동하도록 어플리케이션 서버 계층을 활용하는 것으로서 이로 인해 데이터베이스 워크로드를 낮추는 것을 목표로 합니다. 이것은 또한 데이터베이스와 더욱 효율적으로 상호 작용 하도록 합니다. 파티셔닝 패턴에는 5 가지의 핵심 요소가 있습니다. 다음 테이블은 이러한 요소와 이 요소들이 어떻게 WebSphere XD 에서 사용되는지 설명하고 있습니다.

#	요소	WebSphere 기능
1	파티셔닝	- WebSphere Partitioning Facility (WPF)
2	파티션 인식 워크로드 관리	- WebSphere XD – On demand Router (for HTTP) - EJB 클라이언트 – IIOP 라우팅 - JMS – 풀 모델
3	파티션 활용	- WebSphere Dynamic Caching Service - JDBC Batching
4	고 가용 파티션	- HA 파티션을 위한 High Availability Manager
5	파티션 재조정	- WPF Management Bean

다음 섹션은 파티셔닝 패턴의 요소와 J2EE 어플리케이션 성능과 확장성에 있어서 이 요소들이 아주 많은 개선을 가져오기 위해서 WebSphere XD 에서 어떻게 사용되는지를 설명합니다.

#### 3.2. 파티셔닝

이름에서 제시되었듯이, 파티셔닝은 파티셔닝 패턴의 필수 요소입니다. 자신의 파티션 상의 성능과 가용성을 둘러싼 요구사항을 인식하지 못하는 사이에, 파티션은 이러한 이점이 실현되도록 기반을 구축할 것입니다. WPF 는 어떤 파티션이 WebSphere XD 에서 생성되고 관리되는 지에 관한 기능입니다. WPF 의 용어에서, 파티션은 여러 방식으로 설명될 수 있습니다. 가장 간단하게 표현하면, 파티션은 요청될 때마다 어플리케이션(또는 어플리케이션 안에서의 선언되는 메타데이터)에 의해 작성되는

라벨들의 리스트입니다. 예를 들어, 어플리케이션이 경매가 진행되는 동안 경매에 붙여지는 아이템의 카테고리를 나타내기 위해서 추상적인 파티션 이름 세트를 제공할 수 있습니다. :

*스포츠 용품, 자동차 용품, 장난감 및 골동품 등*

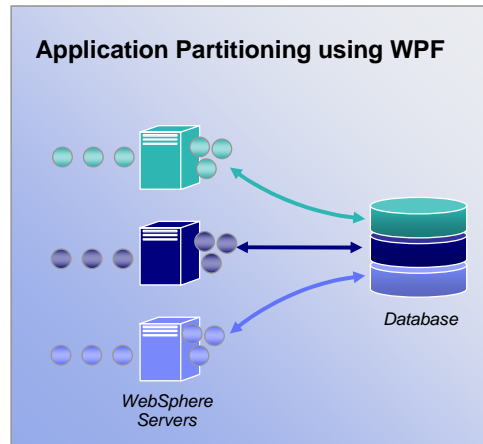
파티션은 HTTP 요청 URI 의 세트로서 표현될 수 있습니다. :

- /stock\_app/stocks/ibm/\*
- /stock\_app/stocks/xyz/\*
- /stock\_app/stocks/abc/\*

파티션은 해쉬 기능을 제공하는 어플리케이션을 사용하여 계산될 수 있습니다. 예를 들어 어플리케이션은 패턴에 대응되는 수를 끌어내기 위해 Arbitrary Token(HTTP 헤더 또는 쿠키 등)에 해쉬 기능을 적용 할 수 있습니다.

어플리케이션이 자신의 파티션을 만들기 위해 앞에 이야기한 방식 중 하나를 사용할 때, WPF(WebSphere Partition Facility)는 각각의 파티션을 WAS 클러스터 안의 서버 프로세스에 할당합니다.

위에서 언급한대로, 파티션은 파티셔닝 패턴의 기반입니다. 다음 섹션에서는 이러한 파티션을 가지고 무엇을 할 수 있는지 설명할 것입니다.



### 3.3. WLM 을 인식하는 파티션

파티션이 생산적이 되도록 하는 첫 번째 스텝은 파티션에서 요청(잠재적으로 변경된 프로토콜의 요청, 예를 들면 HTTP, IIOP 및 메시징 같은)을 결합하는 것입니다. 이러한 경우에서 요청은 파티션과 결합된 것으로서 분류되고 작업은 이러한 특정 파티션이 할당된 서버-프로세스로 전송됩니다. 예를 들면, 온라인 경매의 경우, 경매 참가자가 *스포츠 용품* 카테고리에 있는 아이템에 입찰을 하면, 입찰 요청이 *스포츠 용품* 파티션에 결합된 서버-프로세스로 전송될 것입니다.

WebSphere XD 에서, 요청을 파티션으로 전송하는 여러 가지 방식이 지원됩니다.:

**HTTP 를 위한 On Demand Router (ODR) –** WebSphere XD 의 ODR 컴포넌트는 HTTP 요청을 자신이 결합된 파티션으로 전송합니다. 배포 정책은 이러한 파티션에 매핑되는 HTTP 요청에 의해 파티션과 규칙을 정의하는 WebSphere XD 어플리케이션에 포함될 수 있습니다. 어플리케이션이 시작 될 때, WPF 는 파티션을 결정하고 이 파티션을 WAS 클러스터의 액티브 클러스터의 서버 프로세스에 할당합니다. Partition-to-Server 프로세스 할당은 들어오는 요청을 분류하고 일치하는 파티션에 전송하기 위해 정보를 사용하는 ODR 로 보내집니다.

**IIOP 를 위한 EJB 클라이언트 –** 리모트 EJB 클라이언트는 파티션을 인식하도록 생성될 수 있습니다. 이러한 EJB 가 WAS 클러스터에서 실행되고 있는 리모트 메소드를 호출하면, 유사한 분류 프로세스가 워크로드를 관리하고 라우팅 하는 EJB Stub 안에서 시작합니다. 만약 메소드 호출이 파티션에 결합된다면, 이 리모트 메소드에 대한 요청은 해당 파티션에 대응하는 서버 프로세스로 전송됩니다. 이러한 행동은 작업이 클러스터링된 **Stateful Session EJB** 로 전송되는 것과 유사합니다. 그러나 WPF 를 사용하면, Affinity Process 는 더 정확한 방식으로 제어될 수 있습니다.

**메시지 풀 모델 –** WPF 가 메시징 프로토콜을 위한 명시적인 지원을 가지고 있지 못하다면, 개발자는 Pub/Sub(Publish & Subscribe) 기반 메시징 어플리케이션을 작성할 수 있습니다. 이 시나리오에서, 주어진 서버 상의 어플리케이션은 해당 서버에 할당된 파티션에 대응하는 토픽만을 읽을 수 있습니다. 파티션이 서버에 할당/해제될 때, 서버는 작업을 선택하는 Subscription 세트를 업데이트 합니다. 이러한 어프루치를 통해, 서버는 자신의 파티션에 대응되는 큐/토픽에 메시지를 보낼 수 있습니다.

### 3.4. 파티션 활용

어플리케이션 파티션을 정의하고 이러한 파티션에 메시지를 전송하는 능력은 적절히 활용하기만 하면 매우 강력한 기능이 될 수 있습니다. 어플리케이션은 고성능, 고 가용성의 새로운 수준을 확보하기 위해 이러한 기능을 활용 할 수 있습니다. 파티션과 전송을 통해 WebSphere 클러스터 이 외에서는 할 수 없는 작업을 수행하는 서버 영역을 구축합니다. 파티션의 독점성은 아주



강력한 자산을 만듭니다. 파티션 활용에서 가장 잘 알려진 세가지 방식은 캐싱(Caching), 배치(Batching) 및 싱글톤 서비스(Singleton Services)입니다.

**캐싱(Caching)** – 어플리케이션은 적극적으로 데이터를 캐싱 함으로써 파티션의 독점적인 속성을 활용할 수 있습니다. 예를 들어, 온라인 경매에서, 주어진 카테고리-스포츠 용품-를 위한 입찰은 WebSphere Dynamic Caching Service 를 사용하여 서버의 메모리에 로컬로 캐싱 될 수 있습니다. 어플리케이션은 스포츠 용품 입찰이 하나의 특정 서버에서만 일어나고 클러스터 안의 다른 프로세스에서는 일어나지 않는다는 것을 보장됩니다. 그래서 어플리케이션은 각각의 경매를 위한 입찰을 메모리 상에 캐싱 함으로써 이러한 이점을 얻을 수 있습니다. 이것은 현재 하나의 아이템을 위한 많은 입찰이 데이터베이스로 전송되는 것이 아니고 메모리 상에서 검색되도록 하도록 합니다. 어플리케이션 작성자는 새로운 대규모 입찰이 데이터베이스로 전파되도록 선택할 수 있습니다.

전통적인 캐싱 구조에서, 각각의 프로세스는 모든 카테고리를 위한 모든 경매를 캐싱 하도록 할 것입니다. 파티션을 활용함에 따라, 경매 카테고리는 WAS 클러스터의 모든 메모리에 분산합니다. 이것은 시스템 자원을 더욱 효율적으로 사용하도록 합니다.

**배치(Batching)** – 위의 사례에서, 주어진 아이템을 위한 새로운 대규모 입찰은 로컬 메모리 상의 큐에 저장된 후 Back-end 서버나 데이터베이스 에 한번에 모두 배치로 보낼 수 있습니다. 대용량 환경에서 이러한 배치 처리는 데이터베이스와의 리모트 인터랙션을 아주 많이 절약할 수 있습니다.

**싱글톤 서비스(Singleton Services)** – 파티션은 단일 기수성의 특정 기능 또는 서비스가 있는 WAS 클러스터를 지정 할 수 있습니다. 이 경우에, 이 기능이 실행되는 클러스터의 한 장소에서만 있기 때문에 서비스는 싱글톤 (Singleton)이라고 말할 수 있습니다. 싱글톤은 어플리케이션이 나뉘어져서 클러스터의 여러 서버 상에서 실행될 수 있게 합니다. 싱글톤에 의해 처리될 필요가 있는 작업이 도착하면, 이 작업은 현재 HTTP/IIOP 또는 메시지의 풀 매커니즘을 사용하여 싱글톤을 호스팅하고 있는 서버로 라우팅 됩니다.

파티션을 활용함(캐싱 과 배치)에 따라 WAS 개발팀은 전형적으로 확장하기 어려운 어플리케이션(예를 들어 쓰기 기능 비율이 읽기 기능 비율보다 높은 경우)에서 괄목할 만한 성능 향상을 볼 수 있습니다.

### 3.5. 파티션 고 가용성

---

파티션 활용은 분명한 성능 향상을 가져올 수 있습니다. 그러나 파티션은 예방 활동 스텝을 거치지 않는다면 Single Point of Failure 를 나타낼 수 있습니다. 예를 들어, 데이터 캐싱을 위해 파티션을 활용하는 경우에, 파티션에 Fail 이 발생하면 다음에 클러스터 안에서 Fail 이 발생한 서버 상의 파티션을 위해 요청을 처리할 서버는 없습니다. 싱글톤 서비스를 운영하는 경우에도 동일합니다. 만약 파티션이 문제가 생기면, 서비스를 잃어버리게 되기 때문입니다. 이것은 WPF 가 고 가용성이 가능한 서비스로서 실행되기 때문입니다. WPF 는 WAS 클러스터 내의 프로세스, 서비스 및 파티션을 동적으로 모니터링 하는 WAS 내의 새로운 HA Manager 를 사용합니다. WAS 클러스터 내의 각각의 서버 프로세스는 클러스터 내의 다른 프로세스의 가용성을 인지합니다. 만약 서버 프로세스가 죽으면, 살아있는 프로세스가 Fail 이 발생한 프로세스, 서비스 및 파티션을 회복하기 위해 함께 작업합니다. 예를 들어, 주어진 온라인 경매 카테고리를 위한 파티션을 포함한 서버 프로세스가 Fail 이 발생하면, 클러스터 내의 살아있는 서버는 이것을 감지하고 이 카테고리를 서비스하기 위해 새로운 서버 프로세스를 선택합니다. WPF 및 HA Manager 는 Fail 로부터 아주 빠르게 회복하도록 디자인되었습니다.

### 3.6. 파티션 재조정

---

시스템이 최적으로 실행되고 있다는 것을 확신하기 위해, 때때로 워크로드를 재조정하기 위해서 서버들 사이에서 옮겨질 필요가 있습니다. 예를 들어 스포츠 용품 카테고리가 대용량 요청이 발생한 적이 있으면, 용량이 더 큰 서버로 옮기거나 현재 서비스 되고 있는 서버 대신에 다른 파티션으로 옮기는 것이 유용할 수 도 있습니다. WPF 는 이러한 운영이 가능하도록 해주는 JMX MBean 을 가지고 있습니다. 서버 프로세스 Fail 에 대한 파티션 재조정도 비슷합니다. 그러나, 재조정 작업은 전형적으로 프로그램 또는 운영자의 제어 아래서 계획되고 실행 될 수 있습니다.

### 3.7. WPF 요약

---

WebSphere XD 의 WPF 는 J2EE 어플리케이션이 더 높은 수준의 성능, 확장성 및 가용성을 달성하도록 하여줍니다. WPF 는 엔지니어가 *파티셔닝 패턴*을 사용하도록 합니다. 파티셔닝 패턴은 최대한의 데이터 일관성과 가용성을 요구하는 대용량 OLTP 에서 발생하는 병목을 해소하여 줍니다. 이 패턴은 *파티셔닝, 파티셔닝 인식 WLM, 캐싱, 고가용성 및 재조정*의 5 개의

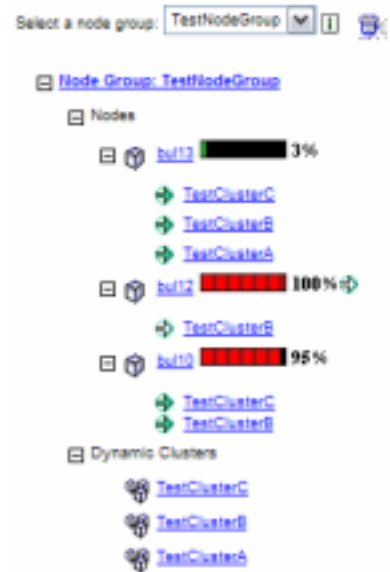
핵심 요소를 확인합니다. 적절히 활용된다면 WPF 는 이전에는 J2EE 환경에서 운영할 수 없었던 거의 선형적인 확장성과 아주 높은 기능을 제공합니다.

## 4. 시각화

제품에서 선택적으로 완전 자동화된 방식으로 복잡한 운영 관리에 제공되는 사용 용이성을 유지하기 위해서, WebSphere XD 는 확장된 관리 기능을 제공합니다. 이것은 동적 WebSphere XD 구성의 그래픽 화면, 성능 통계 보고 및 관리를 위한 운영 실행으로 구성됩니다. 이러한 기능들은 WAS ND 관리 콘솔의 확장된 기능으로서 제공됩니다.

### 4.1. 운영 뷰

WebSphere XD 관리 콘솔은 관리자가 WebSphere XD 구성의 내부 작업을 시각화하는데 도움이 되는 여러 가지 툴을 가지고 있습니다. 이러한 툴은 관리자가 WebSphere XD 환경 안에서 선택할 작업들과 관련되어서 정보를 잘 제공 받도록 합니다. 운영 뷰는 건강도, 성능 및 잠재적으로 자율 결정과 관련된 정보의 직관적이고 중앙 분배 포인트를 나타냅니다.



Runtime Topology

#### 4.1.1. 실행환경 구성

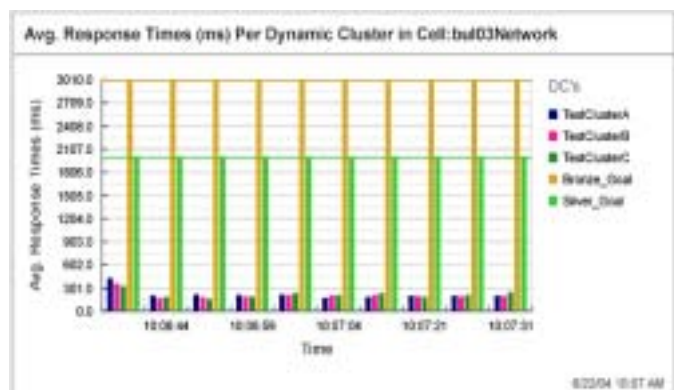
이 같은 뷰의 하나는 WebSphere XD 환경의 한 순간의 상태 묘사인 실행환경 구성 뷰입니다. 이 뷰는 최신 정보를 제공하기 위해 지정된 시간 간격을 가지고 보여줍니다(이 시간 간격은 변경이 가능합니다.). 실행환경 구성은 다음을 포함하는 많은 유용한 정보를 포함합니다.

- 어플리케이션 프로비저닝 활동
- 동적 클러스터 인스턴스의 분배
- CPU 사용량 (노드 당)
- 노드 그룹에 대한 노드의 소속
- 노드 그룹 멤버십에 대한 동적 클러스터
- dWLM 가중치 (어플리케이션 서버 인스턴스 당)
- 프로세스 ID (어플리케이션 서버 인스턴스당)

#### 4.1.2. 차트 생성

WebSphere XD 는 관리자가 WebSphere XD 환경에 전반에 걸쳐 관찰되는 실행 정보를 요구에 따라 변경이 가능한 그래프를 볼 수 있도록 하는 차트생성 기능을 제공합니다. 이 뷰는 실행환경 구성 뷰와 동일한 시간 간격으로 보여집니다. WebSphere XD 는 6 가지의 서로 다른 스타일의 그래프로 아주 다양한 통계의 차트 구성을 지원합니다. 지원되는 통계는 다음과 같습니다.:

- 평균 응답 시간
- 동시 요청 수
- 평균 처리 량
- 평균 큐 대기 시간
- 평균 서비스 시간
- 평균 큐 길이
- 평균 실패 비율



차트생성

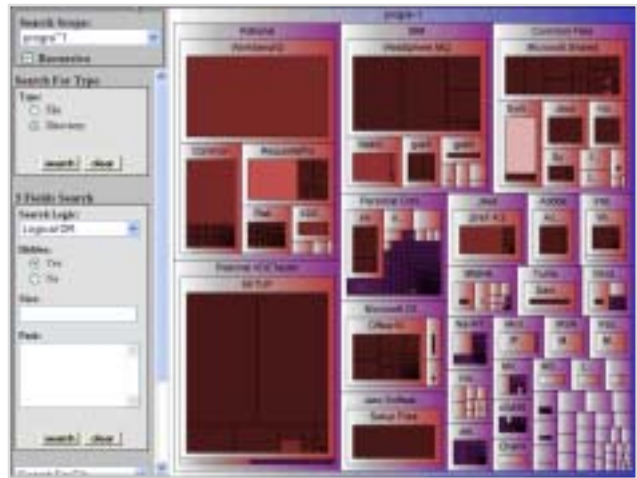
차트는 여러 개의 서로 다른 시각으로부터 만들어 질 뿐만 아니라 관찰되는 통계의 범위에 있어서 유연성을 관리자에게 제공합니다. WebSphere XD 는 셀, 노드 그룹, 동적 클러스터, 서비스 클래스, 트랜잭션 클래스, J2EE 모듈 및 프록시(ODR) 관점으로 부터의 차트 생성을 지원합니다.

WebSphere XD 의 차트 생성 기능은 또한 통계 데이터에 대하여 과거부터의 간단한 로그를 제공합니다. 새로운 데이터 관점이 차트의 오른쪽에 추가됨에 따라, 과거 데이터는 차트의 외쪽으로 옮겨가면서 보여집니다. 물론 어느 시점에서의 차트 상에 보여지는 과거 데이터의 길이는 샘플링 기간당 데이터 포인트 수와 보여지는 차트의 형태에 따라 다릅니다.

#### 4.1.3. 실행환경 맵

WebSphere XD 는 *트리맵*이라 불리우는 혁신적인 시각화 기술을 제공합니다. 트리맵은 계층적인 데이터를 보여주는 방식입니다. 이것은 단순히 반복적으로 더 작은 사각형으로 다시 나누어지는 사각형을 보여주고 각각의 사각형은 보여지는 데이터의 트리 구조에서 어떤 레벨 상의 노드의 집합을 나타냅니다. 각각의 사각형의 크기와 색깔은 의미를 가지고 있습니다.

WebSphere XD 관점에서 최 상위 레벨의 사각형은 셀을 나타냅니다. 이 사각형은 셀 안의 노드 그룹을 나타내는 사각형으로 다시 나누어 집니다. 어떤 트리맵이 보여지는가에 따라 사각형의 크기와 색깔은 특정 통계와 관련이 있습니다. 색깔은 전형적으로 건강도 또는 비즈니스 목표의 성취를 나타내고, 크기는 전형적으로 동시 요청수, 서비스 인스턴스 수 등과 같은 수를 나타냅니다. 기본적인 구성에서는 실행 맵은 전체 WebSphere XD 구성을 나타냅니다. 그러나 관리자는 트리맵의 사각형을 단순히 클릭함에 따라 원하는 대로 더욱 자세한 내용을 확인할 수 있습니다. 트리맵의 예는 옆에 보여집니다.



실행환경 맵

WebSphere XD 의 실행 환경 맵 기능은 아주 대규모 구성을 가지는 관리자에게 특히 도움이 됩니다. 이것은 간결한 방식으로 대규모 데이터 세트를 표현하도록 구성되어 있습니다. 예를 들어, 1000 개의 어플리케이션 세트의 사용은 동적 클러스터의 트리맵을 보는 것과 사각형의 크기를 비교하여 신속히 알 수 있습니다. 많은 서비스 클래스 세트의 목표 달성은 서비스 클래스의 맵의 사각형의 색깔을 보고 알 수 있습니다.

실행환경 맵은 관리자가 전체 맵에서 데이터 서브세트를 선정하고 응답 시간을 기준으로 Top 10 수행 어플리케이션을 강조하거나 5 개의 가장 낮은 동시 요청 수를 가지는 동적 클러스터를 선정하도록 하는 등의 강력한 검색 기능을 제공합니다.

#### 4.1.4. 커맨드 센터

WebSphere XD 의 다음 버전은 관리 콘솔로부터 관리 운영작업을 수행하도록 하는 기능을 제공할 것입니다. 이 기능에는 관리적인 기능에 대한 편리한 접속과 관리자에게 요구되는 WAS Admin 지식이 적어도 되는 두 가지의 이점이 포함되어 있습니다.

### 4.2. 이벤트 통지

관리 콘솔은 자율 관리자에 의해 만들어지는 결정을 관리자에게 통지하도록 확장되었습니다. 통지는 계획된 이벤트와 계획되지 않은 이벤트 모두에게 적용 가능합니다.

#### 4.2.1. 계획된 이벤트

계획된 이벤트는 WebSphere XD 가 액션 플랜을 가진 예상된 이벤트입니다. 동적 클러스터의 범위의 확대를 유발하는 구성된 한계를 넘는 평균 응답 시간이 한 예가 될 수 있습니다. 구성된 자동화 레벨에 따라 이러한 이벤트는 여러 가지 방식 중 한가지

방식으로 관리자에게 통지됩니다. 만약 WebSphere XD 가 “실행” 모드로 운영되고 있다면, 액션 플랜이 실행되고 간단한 통지가 관리자에게 보여집니다. “승인”모드에서는 관리자는 액션 플랜이 나타나고 승인을 기다린다는 메시지가 뜹니다. “수동” 모드에서는 플랜은 권고사항을 따를 지 무시할 지를 결정하는 관리자에게 보여줍니다.

#### 4.2.2. 계획되지 않은 이벤트

액션 플랜에서 할당되지 않는 이벤트는 어떤 예측하지 못한 상황이 발생했을 경우 관리자가 알도록 경고하기 위해서 보여집니다. 이것이 문제가 있다면, 그 후 관리자는 이 상황을 바로잡을 플랜을 마련합니다.

### 4.3. 서비스 가능성 뷰

WebSphere XD 는 다양한 실행 컴포넌트에 대한 “Under-the-covers” 뷰를 제공합니다. 이러한 뷰는 WebSphere XD 환경에서의 문제를 디버깅하도록 하고 구성에서 어떤 일이 벌어지고 있는지 알기를 원하는 관리자를 위한 뷰를 제공합니다. 현재 제안된 뷰의 세트는 다음의 시각화를 가능하게 합니다.

- ODR(On Demand Router) 큐 콘텐츠
- 어플리케이션 위치 결정
- dWLM 라우팅 테이블 및 가중치
- 운영 뷰에서 보여지는 통계적 데이터
- WebSphere XD 의 내부 구성 데이터

## 5. 요약

WebSphere XD 는 확장성, 사용가능성 및 적용가능성을 향상시킨 WAS ND 위에 구축되는 확장 패키지입니다. 이러한 확장 기능은 추가로 현재의 트랜잭션 양의 한계를 확장하면서 더 크고 더 복잡한 환경의 관리를 용이하게 합니다. WebSphere XD 의 새로운 가상화 환경은 고객을 On Demand 세계로 인도하는 자율 컴퓨팅 기능을 제공합니다. 이러한 자율 컴퓨팅 기능은 서로 다른 형태의 작업들 사이의 동일하지 않은 사용 패턴의 장점을 가지기 위해 컴퓨팅 자원이 더욱 지능적으로 관리됨에 따라 더 낮은 총 소유 비용(TCO)을 제공합니다. 자율 컴퓨팅은 사용 패턴의 변화에 더 높은 레벨의 적응성을 제공합니다.

On Demand 를 쉽게 적용하기 위해, WebSphere XD 는 확장성 있는 시각화와 서비스 제공에 대한 그래픽 인터페이스를 제공합니다. 이러한 툴들은 기업의 자율화 수준이 높아짐에 따라 고객이 쉽게 적용할 수 있는 WebSphere XD 의 자율 컴퓨팅 기능에 대한 가치 있는 뷰를 제공합니다. 이러한 뷰는 WebSphere XD 의 이벤트를 보고하고 추가로 자세한 건강도 정보와 성능 정보를 제공합니다. 이러한 모든 기능은 관리자에게 도움이 되도록 중앙제어 패널에 모두 통합되어서 사용 가능성을 강화하였습니다.

WebSphere XD 는 미션 크리티컬 어플리케이션을 위해 확장성 있고 신뢰성 있는 플랫폼을 제공하는 WebSphere 를 위하여 혁신적인 기능을 계속 제공하고 있습니다.