

# 1부 Oracle Exadata 소개

1장 Exadata 개요

2장 Exadata 제품

3장 Exadata 아키텍처

## 1장 Exadata란 무엇인가?

1.1 Exadata 배경

1.2 Exadata 관점

1.3 Exadata 차별성

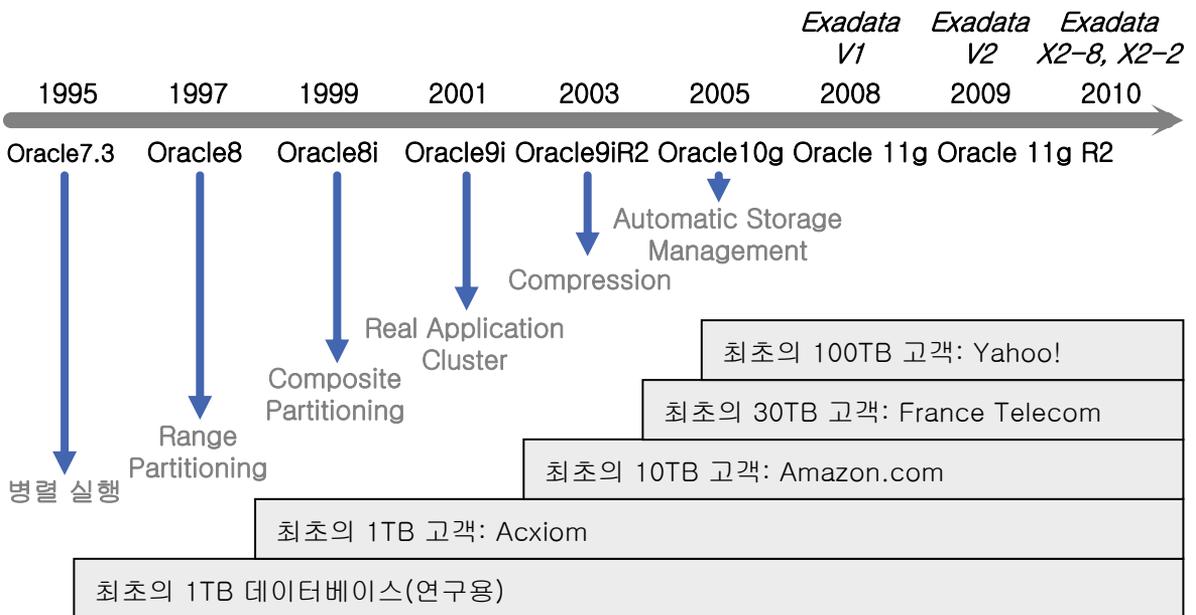
# 1장 Exadata란 무엇인가?

## 1.1 Exadata 배경

### 1.1.1 DW 및 BI 경향

2008년 오라클은 자체 Exadata(엑사데이터, exa: 10<sup>18</sup>) 기술(Exadata Version 1)에 근거한 새로운 데이터 웨어하우징 제품인 HP Oracle Exadata Storage Server와 HP Oracle Database Machine을 발표했다. 그리고 일 년이 지난 2009년 9월, 오라클과 썬(Sun)은 Exadata Version 2를 발표하면서, 그것이 세계 최초의 OLTP 데이터베이스 머신이고 DW와 OLTP를 동시에 지원하는 가장 빠른 머신이라고 소개했다. 그리고 다시 일 년이 지난 2010년 9월, 오라클은 Exadata Database Machine X2-8, X2-2를 발표함으로써 Exadata 제품군을 확장했다(그림 1-1).

<그림 1-1 Oracle의 진화>



출처: Oracle

## 6 1장 Exadata란 무엇인가?

오늘날의 데이터 웨어하우징 및 비즈니스 인텔리전스를 주도하는 몇 가지 큰 흐름이 있다. 먼저 비즈니스 관점에서 기업들은 과거를 분석하고 미래를 예측하기 위해 또 현재의 적시적 비즈니스 기회를 포착하기 위해 더욱 빠르고 심도 있는 분석으로 경쟁하고 있다. 이에 따라 사실에 근거한 신속한 의사결정 및 관리가 더욱 필요하게 되고, 이는 다시 보다 정확한 분석을 보다 빨리 실행하는 한편, 보다 상세하고 보다 많은 데이터를 보다 새로운 데이터(비구조화된 데이터(unstructured data)를 포함해)와 함께 보다 오래 유지하도록 요구한다. 따라서 실시간(real-time) 데이터 웨어하우징이나 운영(operational) BI, EII(Enterprise Information Integration), unstructured DW, 프로세스 내 분석(in-process analytics), 데이터베이스 내 분석(in-database analytics), 고급 분석(advanced analytics) 등으로 분석의 기술과 영역이 확대된다. 아울러 이러한 분석 요구를 지원하기 위한 시스템 환경으로, 플랫폼(platform) 구성 및 설정의 오류를 방지하고 성능을 획기적으로 향상시키기 위해 고도로 최적화된 고성능 대용량 플랫폼이 더욱더 많이 사용된다. 여기에 클라우드 컴퓨팅(cloud computing)과 오픈 소스(open source)의 발전에 힘입어 한층 더 접근과 사용이 용이하고 관리가 단순하며 비용 효과적인 새로운 시스템(환경) 및 서비스가 제시되고 있다.

DW는 이제 옵션이 아니라, 미션 크리티컬(mission-critical: 중요한 업무를 수행하는) 시스템으로 점점 더 인정된다. 따라서 지금의 DW는 엔터프라이즈 와이드(enterprise-wide) 차원의 통합되고 공유되는 하나의 데이터 소스 역할을 포함해 혼합 작업부하(mixed workload: 대량의 데이터 적재, 복잡한 분석 쿼리, 트랜잭션성 데이터 처리와 같이 상이한 작업부하들이 동시에 복합적으로 일어나는 작업부하 유형)를 처리하는 능력을 더 많이 보유하도록 요구된다(심도 있는 마이닝(deep mining)을 실행하는 복잡한 쿼리(query)와 빠른 처리를 요하는 전술적 쿼리의 서비스 레벨은 서로 다른 기대치를 가진다. 이 상이한 작업부하들은 CPU, 메모리, 디스크, 네트워크를 두고 서로 경쟁한다). 또한 데이터 신선도 혹은 데이터 대기 시간(data latency)의 요구 역시 일괄 처리에서 지속적인 적재로 나아간다.

Exadata는 DW 및 BI의 발전적 경향에 부응하고 기존 Oracle 시스템의 주요 단점들(I/O 병목, 과도하고 불충분한 시스템 설정 및 튜닝 등)을 극복하기 위한 노력의 일환으로, 또 IBM, Teradata, Netezza(현재의 IBM)와 같은 경쟁 업체에 대응하기 위한 전략의 일환으로, 나아가 OSFA(One Size Fits All) 머신을 가지고 새로운 오라클 시장을 주도하고자 하는 계획의 일환으로 등장했다. 그런데 Exadata 등장 기술적 배경에는 이미 무르익은 일련의 흐름 — 데이터베이스 성능을 획기적으로 개선하기 위한 데이터베이스와 스토리지의 결합, 차세대 데이터베이스 스토리지 대안인 SSS 기술, 시스템 사전-구성 및 최적화를 위한 DW appliance 접근 방식 등 — 이 또한 있다.

### 1.1.2 데이터베이스와 스토리지의 결합

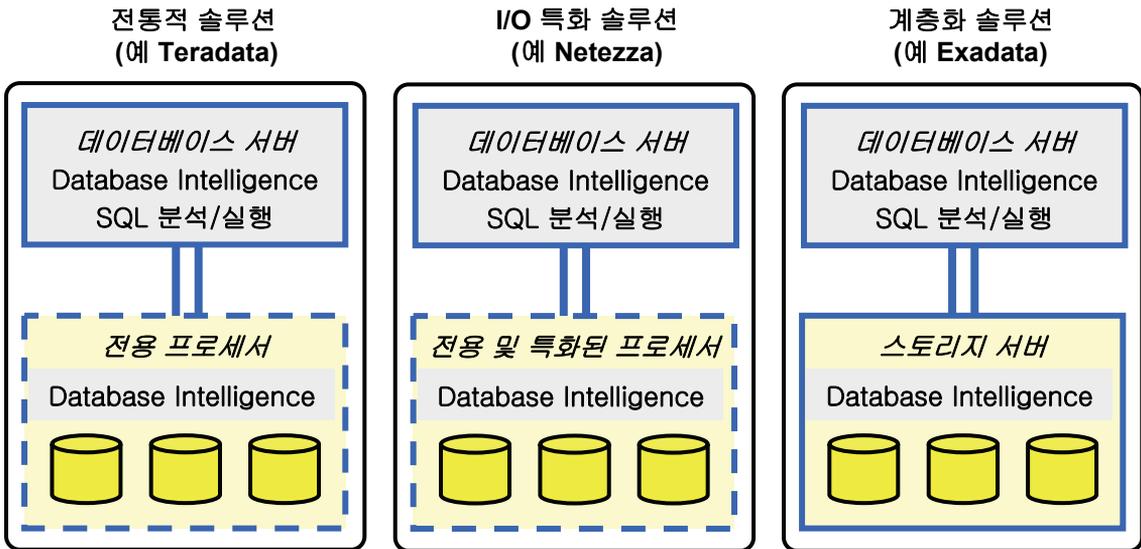
하드웨어 및 소프트웨어 기술의 획기적 발전에도 불구하고 현재의 데이터베이스의 전반적 성능과 응답 시간은 여전히 데이터베이스 크기에 영향을 받는다. 성능 향상과 용량 관리를 위해 다양하고 새로운 ILM 솔루션을 적용하고 데이터 아카이빙(archiving), rolling window, 데이터 압축과 같은 기술로 데이터베이스를 잘라내고 압축할 수는 있지만, 그렇다고 그것만으로 필요한 모든 데이터가 반드시 온라인으로 유지되고 충분히 쿼리 가능하면서 만족할만한 성능이 보장되는 것은 아니다. 여전히 대형 데이터베이스 구현의 근본 과제는 기존 스토리지 시스템의 적정 한도를 넘어선 엄청난 I/O 발생과 작업부하를 어떻게 잘 처리할 것인가이다. 스토리지 하부구조의 전반적 처리 효율과 처리 시간은 최종 사용자 관점의 쿼리 응답 시간에 직접적 영향을 미친다.

더군다나 급격한 데이터 성장은 서버와 스토리지 사이에 거대한 기술적 불균형을 만들었다. 그 불균형이란 다름아닌 프로세서와 디스크 사이의 성능 갭이다. 듀얼 코어(dual core), 쿼드 코어(quad core) 프로세서의 등장에서 보듯, 일반적으로 무어의 법칙(Moore's Law: CPU 처리 능력이 매 18개월 혹은 매 2년마다 2배씩 증가한다는 법칙)은 계속 유효하다고 입증되는 반면, 디스크 처리 속도 및 관련 소프트웨어 성능은 여전히 이에 훨씬 못 미치는 수준으로 향상되고 있다. 회전하는 디스크에 접근하는 속도의 발전은 CPU 처리 능력의 향상을 따라가지 못하는 것이다. 예를 들어, CPU 성능은 무어의 법칙, 멀티-코어 프로세서(multi-core processor) 및 스레딩(threading) 기술의 발전에 따라 획기적으로 향상된 결과, 1987년 이래 2,000,000배가 증가된 반면, 디스크 액세스 타임(disk access time) 성능은 겨우 11배가 향상된 것이다. 오늘날의 시스템에서 디스크 드라이브(disk drive)는 유일한 기계 장치를 대표한다(1956년 IBM에 의해 최초로 소개된 디스크 드라이브는 분당 1,200번 회전했다. 현대 드라이브가 분당 15,000번 회전한다면, 52년 동안 단지 12.5배 향상된 것이다 — 컴퓨터의 다른 대부분의 성능은, 측정 결과, 대략 7년마다 12.5배 증가한다고 한다. 한편 디스크 집적도(density)는 1956년 이래 100,000,000배 증가했고, 칩(chip)당 트랜지스터(transistor) 수는 1971년 이래 100,000배 증가했다). 드라이브 회전을 통해 데이터를 찾는 물리적 제약으로 인해, 각 디스크는 수적으로 제한된 I/O를 제공할 수 밖에 없다. 당연히 디스크 I/O 속도는 데이터를 처리하는 CPU나 그것을 전송하는 네트워크 속도에 비해 너무 느리다. 결과적으로 디스크 스토리지는 대형 데이터베이스 구현에서 반드시 해결해야만 하는 심각한 병목현상(bottleneck)을 초래한다. 이런 상황은 결국 새로운 데이터베이스 스토리지 대안을 고려하게끔 만든다.

그 하나의 결과가 데이터베이스와 스토리지의 일체화 지향이고 다른 하나는 SSS 대안이다. 데이터베이스와 스토리지가 더욱더 하나로 합쳐져 구성된다. 데이터베이스와 스토리지를 더욱더 단단히 결합하기 위한 방법으로, 가상 스토리지나 자동 볼륨 관리와 같은 스토리지 관리 기능을 데이터베이스 내에 포함시키기도 하고, 데이터베이스 인식 기술을 스토리지 내에 혹은 가까이 구현하기도 한다. 이러한 수렴 내지 최적화는 네트워크 통행을 최소화하고 성능을 확보하기 위한 하나의 근간이 되어 데이터베이스 인텔리전스(database intelligence)를 극대화하고 물리적 스토리지에 더욱 다가선다. 수작업이 가능한 배제된 데이터베이스 성능의 최적화 요구가 커져감에 따라 데이터베이스와 스토리지가 더욱더 단단히 밀결합되는(tightly-coupled) 경향이 일어나고 있다.

(현재) 데이터베이스와 스토리지의 결합에는 세 가지 유형이 있다(그림 1-2). 첫 번째는 전통적 MPP 솔루션으로, 여기서는 프로세서 역할별 분산 처리와 특화된 interconnect를 근간으로 한 최적화 기반 상에서 고도로 정교한 데이터 파티셔닝(주로 hashing 알고리즘 사용)과 병렬처리의 극대화(parallel everything)가 이루어진다. 데이터베이스 I/O 또한 고유의 파일 시스템을 통해 최적화된다. 이 유형은 가장 오랜 기간 동안 그 유효성이 입증된 것으로, Teradata가 대표적인 예이다. 두 번째 유형은 보다 I/O 특화된 솔루션으로, MPP 및 병렬처리의 기반 상에서 데이터베이스 인텔리전스의 일부(데이터 필터링 등)를 보다 스토리지 가까운 곳에다 옮겨놓음으로써, 디스크로부터

<그림 1-2 데이터베이스와 스토리지의 결합>



출처: 오라클 엑사데이터 데이터베이스 머신

데이터가 흘러나오면서 바로 해당 데이터만 걸러지도록 한다. 이러한 스트리밍 및 순차 처리에 기반한 시스템 구조 및 기술은 또한 시스템 전반의 운영관리를 고도로 단순화시킨다. 대표적인 예로 Netezza가 있다. 세 번째 솔루션은 데이터베이스 작업부하를 데이터베이스 계층과 스토리지 계층으로 이원화해 처리하는 구조로, 스토리지 계층에서는 데이터 집약적 처리(주로 사전 I/O 제거 및 데이터 필터링)를, 데이터베이스 계층에서는 계산 집약적 처리(집계, 정렬 등)를 각각 수행한다. 결과적으로 작업부하 분산(workload offloading)과 아울러 보다 적은 데이터가 고속의 interconnect를 통해 이동됨으로써, 데이터베이스 I/O 및 작업부하 최적화가 실현된다. 오라클 Exadata 시스템이 여기에 해당된다(Teradata, Netezza, Exadata는 각 유형의 원조이기도 하다).

데이터베이스와 스토리지의 결합은 데이터베이스 I/O를 최적화하기 위한 하나의 대안이다. 그런데 데이터베이스 I/O나 스토리지 최적화를 포함해, 데이터베이스 최적화 차원의 새로운 특화 기술들이 밀려들어 오고 있다. 이 “제3의 데이터베이스 물결” 속의 주요 기술로는 칼럼 오리엔테이션(column orientation) 및 칼럼 기반의 데이터 압축, 맵리듀스(MapReduce, Hadoop), 인-데이터베이스(in-database) 분석, OLAP 큐빙(cubing), 인-메모리(in-memory) 데이터베이스, 하이브리드(hybrid) 데이터베이스, 상관관계형 데이터베이스(correlation database), 지능적 index, 지능적 스토리지, FPGA 등을 들 수 있다. 예를 들면 어떤 DW 어플라이언스(appliance) 제품의 핵심 기술은 두 개인데, 하나는 폰 노이만(von Neumann)의 병목을 깨는 SQL Chip이고, 다른 하나는 디스크 I/O 병목을 해결하는 칼럼 오리엔테이션이다(그림 1-3, 1-4).

<그림 1-3 폰 노이만 구조 기반의 현재 솔루션>



출처: Kickfire

<그림 1-4 폰 노이만 병목을 제거하는 SQL Chip>

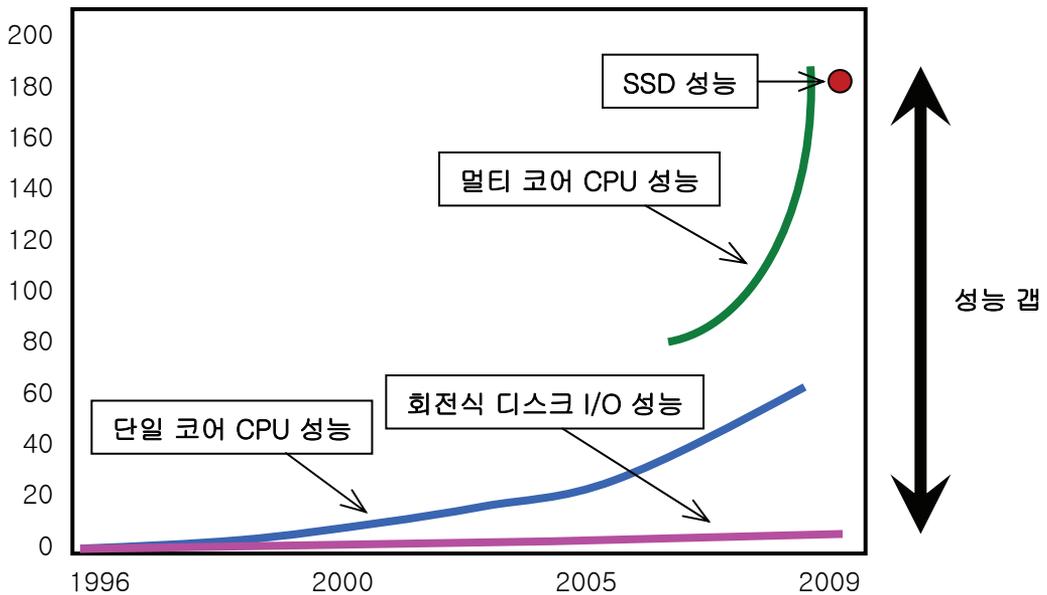


출처: Kickfire

### 1.1.3 차세대 데이터베이스 스토리지 — SSS

데이터베이스 성능과 관련해 SSS(solid-state storage)라는 소위 차세대 저장 장치를 통해 스토리지 I/O의 물리적 한계를 보다 근본적으로 제거하려는 경향이 있다. 이는 디스크 I/O 병목 현상을 (물리적으로) 해결하기 위해 전기기계식 디스크 스토리지를 전자식 반도체 스토리지로 대체하는 것이다. (주지하다시피) 현재의 프로세서 능력은 RAID controller와 디스크 드라이브의 성능(회전 속도와 IOPS)을 훨씬 능가한다. 결과적으로 서버와 스토리지 시스템 간의 불균형은 CPU와 스토리지 사이의 성능 갭으로 인해 더욱 심화된다(그림 1-5, 1-6, 1-7).

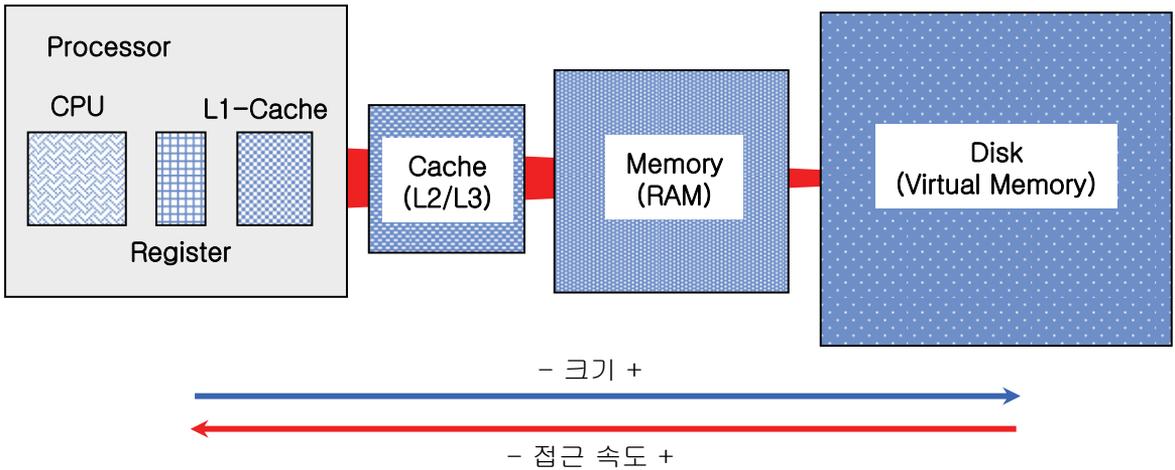
<그림 1-5 디스크 성능 병목 현상>



출처: Intel

만일 이 디스크 성능 문제가 근본적으로 해결될 수 있다면, 그래서 CPU와 디스크의 엄청난 성능 갭을 메울 수 있다면, 데이터베이스 성능 향상을 위해 인덱스를 만들고 요약 테이블(summary table)을 만들고 데이터베이스(I/O)를 튜닝하고 더 많은 드라이브를 추가하고 더 큰 하드웨어를 도입하는 등등의 일들이 상당 부분 불필요해질 수 있다. SSS는 메모리와 디스크 사이의 갭을 메우는 역할을 통해 디스크 I/O로부터 비롯되는 제반 문제를 해결하고자 한다(그림 1-6, 1-7).

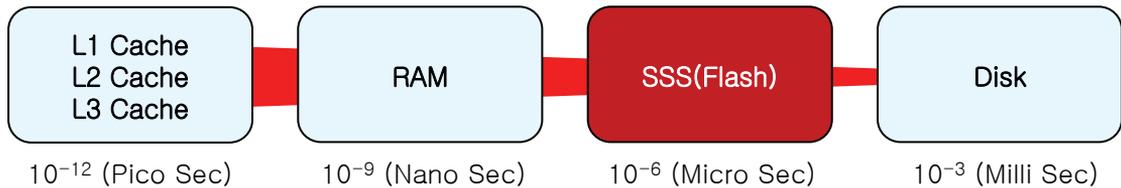
<그림 1-6 종래의 메모리/스토리지 계층>



메모리			
액세스 타임	<1ns	1-5ns	5-50ns
프로세서 속도	3GFlops	600MFlops	60MFlops

※ 캐시는 데이터가 재사용(data locality)될 때에만 작동한다. 메모리에 대한 랜덤 액세스(예: pointer chasing)는 프로세서를 메모리의 속도로 낮출 수 있다.

<그림 1-7 새로운 메모리/스토리지 계층 구조>



출처: Sun Oracle

“solid-state”란 solid state physics(고체물리학: 고체의 구조와 성질 그리고 그와 관련된 현상을 규명하는 물리학의 한 분야)에서 유래된 용어로, 원래는 진공관(electron tube)이 아닌 반도체 장치를 가리키는 의미로 사용되었으나, 현재는 고형의 전자식(electronics) 장치를 전기기계식(electromechanical: 전기에 의해 작동되거나 통제되는 기계 장치 혹은 시스템 방식) 장치와 구별하기 위해 사용된다. SSS는 종래의 자기식(magnetic) 혹은 광학식(optical) 매체와는 상당히 다른 IC(Integrated Circuit)를 채택한 비휘발성 스토리지 매체로, 대용량의 비휘발성 메모리와 동등하

게 가능하다. SSS 장치는 움직이는 부품을 가지지 않고, 데이터와 정보는 종래와 같이 전자적으로 저장되고 또 이 장치로부터 직접 검색된다. 대중적으로 잘 알려진 SSS 매체로는 USB(Universal Serial Bus) 플래시(flash) 드라이브(플래시 메모리 USB 장치), JumpDrive, CompactFlash, SmartMedia, Memory Stick, PCMCIA Type I & Type II, 그리고 기업용의 다양한 제품(업체 독자적)이 있다.

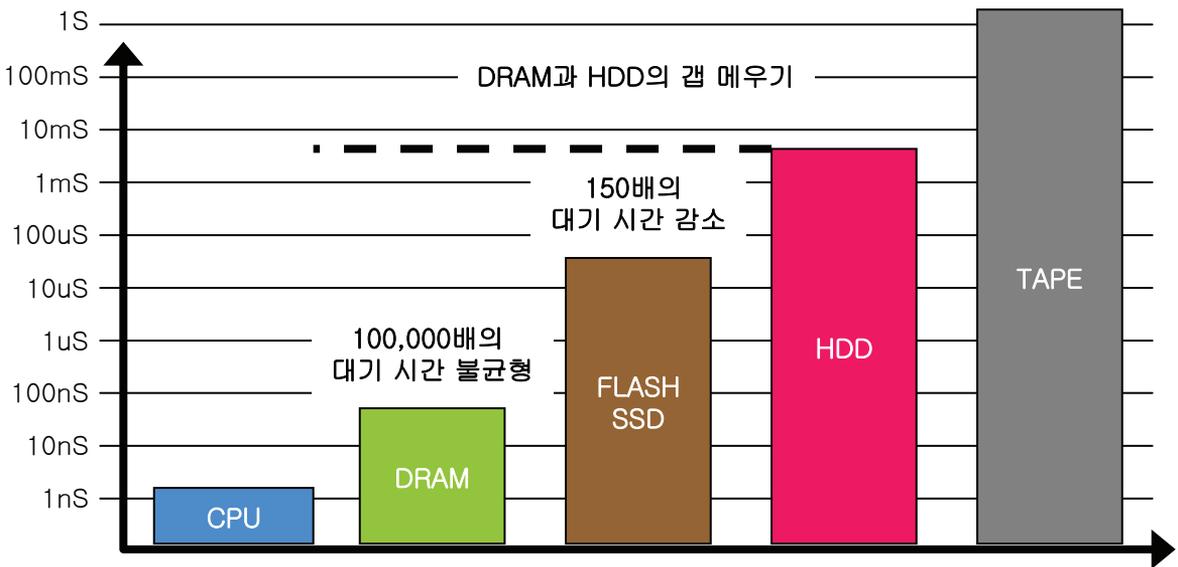
SSS 장치의 주요 이점은 SSS가 기계 부품을 가지고 있지 않기 때문에, SSS에서는 모든 것이 전자적으로 처리된다는 사실이다. 결과적으로 SSS를 통하는 데이터 전송은 전기기계식 디스크 드라이브보다 훨씬 더 고속으로 수행된다. 또한 움직이는 부품이 없다는 것은 그것의 작동 수명이 더 오래갈 수 있다는 의미이기도 하다(장치가 적절히 관리되고 정전기 방전에 노출되지 않는다는 전제에서). 반면에 단점은 용량 면에서 전기기계식 드라이브에 못 미치고, 비용 면에서 더 고가라는 점이다. 그렇지만 가격 갭이 좁아지는 추세이고, SSS 시장은 그 용량과 함께 성장하고 있다. 현재 차세대 데이터베이스 스토리지로 채택되는 SSS 매체는 대부분 플래시 메모리 기반이다(업계에서는 오라클이 Exadata V2가 최초의 플래시 최적화 데이터베이스라고 주장하는 반면, Teradata는 SSD(solid-state disk/drive)를 채택한 자사 제품이 세계에서 가장 빠른 DW 플랫폼이라고 주장하고, IBM 역시 이 분야의 원조임을 내세운다).

플래시 메모리는 DRAM(dynamic random access memory)이나 SRAM(static RAM)과 달리 전원이 끊겨도 저장된 데이터가 그대로 있을 뿐 아니라, 데이터의 입출력도 자유로운 비휘발성 컴퓨터 스토리지 장치로, 전원 없이 데이터를 보존하는 ROM의 장점과 데이터 입출력이 자유로운 RAM의 장점을 모두 가진다. 현재 플래시 메모리는 주로 메모리 카드(memory card), USB 플래시 드라이브, 일반 스토리지 용의 SSD로 사용된다. 원래 플래시 메모리는 기존 EEPROM(electrically-erasable programmable read-only memory)의 특별한 유형인데(EEPROM cell의 구성과 동작을 변형한 것으로 EEPROM에 비해 큰 블록을 사용한다), 비용 면에서 (byte-programmable) EEPROM보다 훨씬 저가이기 때문에 대용량의 비휘발성 SSS가 필요한 모든 곳의 대표 기술이 되었고, 따라서 휴대폰·디지털 텔레비전·디지털 캠코더·디지털 카메라·MP3 플레이어 등에 널리 쓰이게 되었다(플래시 메모리라는 명칭은 1984년 도시바(Toshiba)가 Flash EEPROM이라는 이름으로 논문을 발표한 데서 유래했다).

SSS의 일종인 SSD는 고체의 메모리를 사용해 데이터를 영속적으로 저장하는 데이터 스토리지 장치이지만, HDD(Hard Disk Drive) 인터페이스를 에뮬레이션하기 때문에 대부분의 애플리케이션

에서 쉽게 HDD를 대체해 사용할 수 있다(그림 1-8, 1-9). HDD처럼 알고 사용하면 된다는 얘기가(이 때 SSD는 디스크 컨트롤러의 통제와 제한을 받는다). 그러나 SSD는 HDD처럼 회전하는 금속 원반이 아닌 고정된 반도체에 데이터를 저장한다. 전자 회로로 읽고 쓰기 때문에 HDD와 달리 기록용 금속 원반과 구동용 모터가 필요 없다. 따라서 HDD보다 더 견고하고 크기는 물론 전력 소모와 소음도 줄일 수 있다(냉각 팬이 사용되지 않는다면 소음도 없다). 또한 기계적 지연이 없기 때문에 액세스 타임(access time)과 대기 시간을 줄이고, 따라서 데이터 기록 및 검색 속도도 당연히 빠른 한편, 온도 적응력도 뛰어난 것으로 알려져 있다. SSD는 SRAM 혹은 DRAM을 사용할 수 있으나, 최근에는 NAND 기반의 플래시 메모리가 대부분의 SSD에서 표준으로 인정된다.

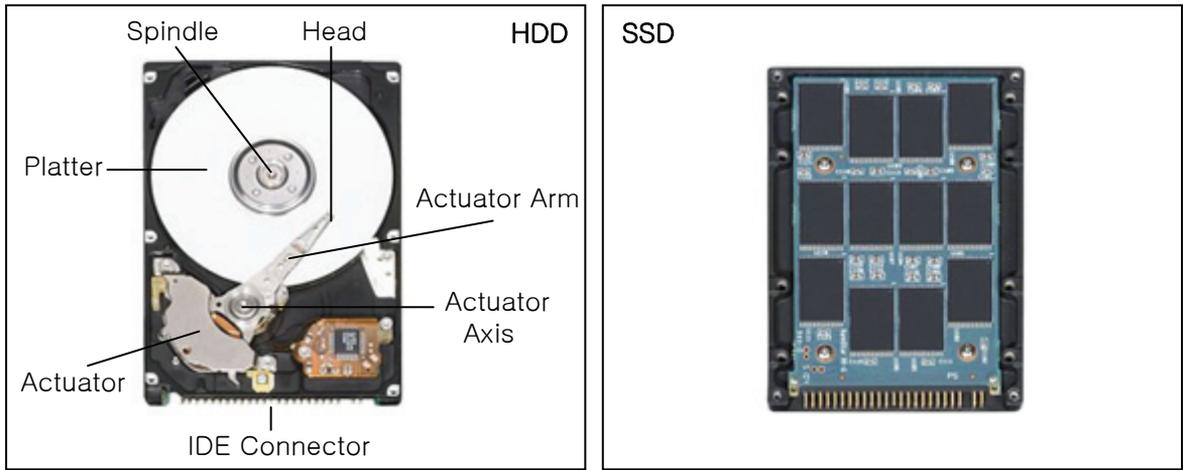
<그림 1-8 대기 시간 비교>



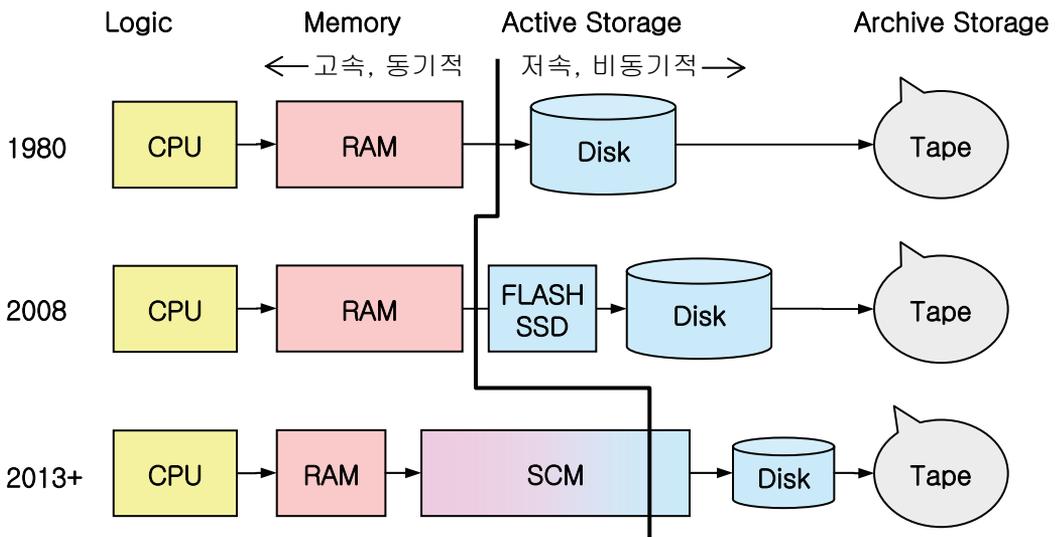
출처: Sun

그런데 스토리지 진화 관점에서 SSD가 회전식 자기 디스크의 대체로 제시되는 반면, 향후의 플래시의 기능적 대체로 SCM(Storage Class Memory)이 제시되기도 한다(그림 1-10).

<그림 1-9 HDD와 SSD>



<그림 1-10 스토리지 시스템 진화 — SCM>



※ SCM은 새로운 유형의 데이터 스토리지 및 메모리 장치를 일컫는 IBM 용어이다. SCM은 움직이는 부품이 없는 비휘발성의 solid state 기반으로, 메모리와 스토리지의 구분을 희석시킨다. IBM에 의하면 플래시는 메모리로 사용될 수 없다는 점에서 SCM으로는 자격 미달이다.

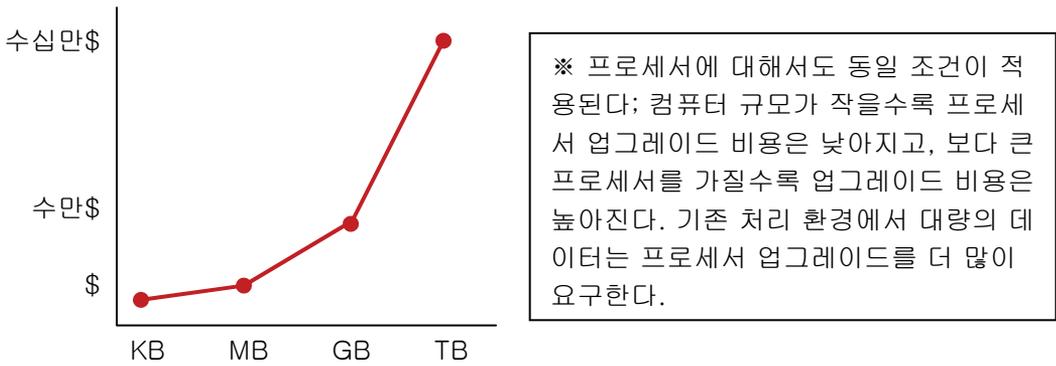
### 1.1.4 데이터 웨어하우스 어플라이언스

지속적인 데이터양의 증가와 신속한 처리의 요구에 따라 더 많은 스토리지, 더 많은 프로세서, 더 많은 관리가 필요 된다. 즉 더 많은 데이터는 더 많은 스토리지를 가지는 것만으로 끝나는 것이 아니라, 그 데이터를 가지고 무언가를 처리할 더 많은 프로세서를 필요로 한다. 스토리지에 대응되는 프로세서를 고려해야 하는 것이다. 대량의 데이터와 제한된 처리 능력이 만나면 필연적으로 성능의 병목 현상을 초래한다. 이에 따라 운영 비용, 하드웨어 비용, 스토리지 비용, 소프트웨어 비용, 유지보수 비용이 상승한다. 그런데 비용 문제의 실제 핵심은 따로 있다. 그것은 데이터 관리 비용의 상승이다. 데이터 크기가 커지면서 빅 데이터 환경을 관리하고 튜닝하고 장애를 처리하는 등의 상황이 악화된다. 데이터를 관리하기 위한 하부구조가 복잡해지고, 데이터를 관리하는 운영적 과부하가 두 가지 차원 — 데이터베이스 관리와 스토리지 관리에서 급격히 증가한다. 이는 데이터양의 증가로 인해, 데이터를 관리하기 위한 하부구조 비용의 상승 정도가 스토리지 비용의 하락 정도보다 훨씬 더 높다는 의미이다. 데이터 양이 작을 경우에는 하부구조 비용 또한 최소한으로 유지되지만, (DW 환경에서) 스토리지가 수TB를 넘어가면 얘기가 달라진다(그 데이터를 관리하기 위한 하부구조의 비용이 TB당 \$500,000에서 \$1,000,000까지 될 수 있다는 얘기다).

스토리지 관리 관점에서 대용량 데이터베이스는 반복적이고 노동 집약적인 작업들을 발생시킨다. I/O 성능을 극대화하기 위해 데이터를 스토리지 시스템 상에 골고루 분산시키고 작업부하의 균형을 주의해서 맞춰야 한다. 이 모든 운영 작업들은 지속적인 반복을 요하고 시간 소모적이며 많은 부분에서 자동화를 벗어나있다. 결국 데이터베이스 관리 관점에서 대용량 데이터베이스를 최적으로 유지하기 위해 요구되는 튜닝 및 관리는 고도로 숙련된 전문가를 필요로 한다. 따라서 데이터베이스 성장에 따라 그것을 유지하고 지원하기 위해 지불해야만 하는 비용은 획기적으로 증가한다. 대용량 데이터베이스의 관리 비용은 데이터베이스 크기에 비례하는 정도가 아니라, 크기가 커질수록 비약적으로 상승한다(그림 1-12). 이 운영 비용의 상승은 곧잘 데이터베이스의 성장과 DW 및 BI의 발전을 가로막는 장애가 될 수 있다.

그간 DW 및 BI 환경이 성숙되면서 DW 운영과 관련된 많은 이슈들(데이터양, 스토리지, 성능, 가용성 등)이 부상했고, 많은 기업들에서 DW 유지 비용은 부담스럽거나 엄청난 것이 되었다. 따라서 보다 비용 효과적인 초점에서, 대용량 데이터베이스를 지원하고 획기적 성능과 확장성을 보장하며 단순한 운영관리를 제공하는 경제적인 DW를 구현하는 것이야말로 IT가 해결해야 할 당면한 최대 과제들 중의 하나이다.

<그림 1-12 스토리지 하부구조 비용>



출처: TDWI

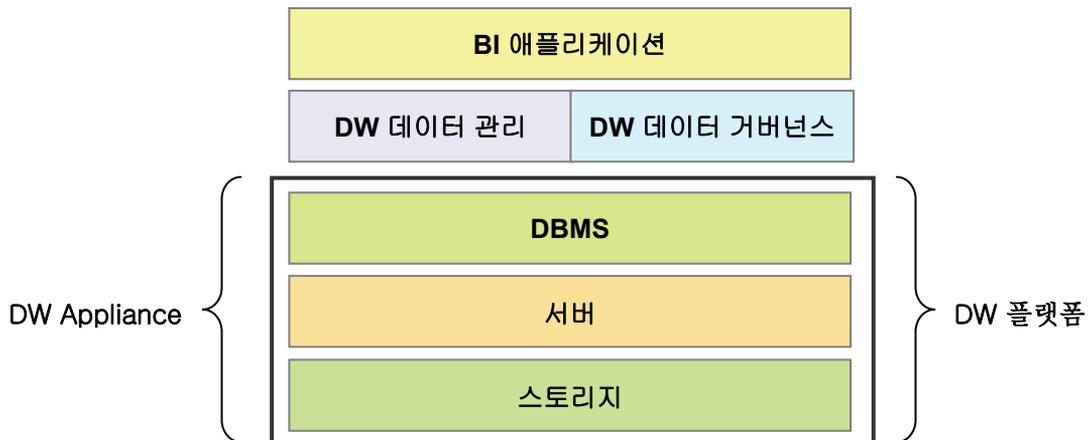
그런데 저가의 컴퓨팅 파워, 전용화된 하드웨어, 새로운 데이터 구조, 지능적이고 정밀한 소프트웨어 등이 등장하면서 과거의 개념, 아이디어, 노력 대신 새로운 시각과 관심으로 새로이 DW에 접근할 수 있는 기회가 생겼다. 이 중 DWA(data warehouse appliance)는 뛰어난 성능 및 확장성과 용이한 관리의 이점을 가지고 DW 구축 및 유지관리 모두에 결코 적잖은 영향을 미친다. DW와 BI를 위해, 저비용의 획기적 성능 향상과 함께 비용 효과적인 확장성과 단순 용이한 관리를 제공하도록 특화된 DWA는 분명 새로운 접근 방식이며 기존 DW 문제를 해결할 수 있는 하나의 솔루션이다.

appliance(어플라이언스)란 사전에 의하면 “특정한 일을 수행하기 위한 장치 혹은 기계(a device or machine for performing a specific task)”로 정의된다. 즉 어플라이언스란 특정 목적을 위해 만들어진 블랙 박스 머신이다(data warehouse appliance라는 용어는 Netezza와 Dataupia의 설립자인 Foster Hinshaw가 2002년 처음 사용한 반면, data appliance란 용어는 1990년대 후반 최초의 data appliance 구현을 주장한 MPP 업체인 WhiteCross가 처음 사용했다). 그런데 과거에도 데이터베이스 머신(database machine)이라는 특정 목적 — 데이터베이스 애플리케이션, 성능, 용이한 관리 등 — 을 위한 전용 플랫폼들이 있었다(그들 중 상업적으로 성공을 거두기는 Teradata가 거의 유일하다). 그런데 왜 이제 와서 데이터베이스 머신 대신에 DW 어플라이언스인가? 그 이유는 초기의 데이터베이스 머신의 실패로 인해 형성된 부정적인 선입견을 피하면서, 새롭게 가능해진 상용 장치와 오픈 소스 그리고 플러그-앤-플레이(plug-and-play)라는 차별적 이점을 강조하기 위한 의도 때문으로 보인다(그런데 아이러니컬하게도 데이터베이스 머신이라는 이고전적 명칭이 최근 Oracle Database Machine으로 다시 나타났다. 이는 아마도 어플라이언스 후

발 업체라는 인식을 피하면서 정통성을 부각시키려는 또 다른 차별적 의도 때문으로 보인다).

2000년대 들어와, 본격적으로는 2000년대 중반 이후, 종래의 DW 기술로 달성하지 못했던 이점들 — 기본적으로 낮은 비용으로 뛰어난 성능을 제공하는 것 — 을 실현함으로써 업계의 상당한 주목을 받고 있는 솔루션으로 단연 데이터 웨어하우스 어플라이언스를 꼽을 수 있다. 범용 RDBMS 업체들이 운영과 분석 환경 모두를 위한 기능과 능력을 계속적으로 향상시키는 가운데, Netezza(현재의 IBM)와 DatAllego(현재의 Microsoft) 같은 업체들은 특별히 분석 환경을 위해 특화된 DWA라는 제품을 제공하는데 집중했다. DWA 제품은 고속의 조회 및 적재 성능을 지원하기 위해 미리 통합된 서버, DBMS, 스토리지로 구성된다(그림 1-13).

<그림 1-13 DW 어플라이언스와 BI 하부구조>



출처: 오라클 엑사데이터 데이터베이스 머신

주로 DW 플랫폼에서 발생하는 DW의 전형적 제약으로 일괄 처리 위주, 시스템 사용 제한, 시스템 구성의 복잡성, 장시간 실행 쿼리, 과도한 요약 데이터(summary data), 최적화 한계, 지속적인 업그레이드 주기, 다수 업체들에 대한 개별 접촉 등을 들 수 있다. DWA는 이러한 종래의 DW의 한계를 극복하고자 한다. 따라서 DWA의 목적은 분석 환경을 위한 성능 및 확장성의 획기적 향상, 신속한 설치 및 용이한 운영관리, 총소유비용(total cost of ownership)의 확실한 차별성을 보장하는 것이다. 즉 보다 빠른 ROI(return-on-investment: 투자회수)를 가능케 하는 보다 낮은 총소유비용(TCO, total cost-of-ownership)으로 보다 고성능의 데이터 웨어하우징을 제공하는 것이다. 그럼으로써 이전에는 기술적인(성능 문제) 혹은 비용적인(투자회수 문제) 이유로 불가능했던 비즈니스 케이스(business case)들을 구현할 수 있게 하는 것이다.

한편 DWA의 성능이 뛰어날 수 있는 이유는 데이터 집약적 처리의 목적에 맞는 하드웨어, 소프트웨어 구조를 채택할 뿐 아니라, 고유의 특화 기술을 통해 데이터베이스 I/O 및 스토리지를 최적화하고 병렬처리를 극대화하기 때문이다. 또 DWA가 총소유비용을 낮게 가져갈 수 있는 이유는 상용 요소 및 오픈 소스의 사용에 따라 하드웨어와 소프트웨어의 비용을 낮출 뿐만 아니라, 단순한 운영 환경과 용이한 사용 덕분에 설치, 관리, 지원의 비용을 줄일 수 있기 때문이다. 이러한 고성능, 고효용도는 프로젝트를 보다 빨리 개발하고 인도할 수 있는 차별적 이점을 다시 제공함으로써, 비즈니스 가치 실현에 걸리는 시간을 대폭 줄인다. 현재의 비즈니스 상황에서는 가능한 빨리 ROI를 달성하는 것이 중요하다. DWA는 신속한 비즈니스 가치 실현으로 총소유비용을 감소시키고, 따라서 비즈니스 ROI를 증가시킨다. 실제로 DWA를 사용한 고객들은 상당한 ROI를 달성했고, DWA 사용에 매우 만족해 하는 편이다. 물론 데이터 웨어하우징 프로젝트의 총비용은 DWA 비용 이상의 것들을 많이 포함한다. 대다수의 프로젝트에서 가장 큰 비용은 소스 데이터를 추출하고 통합해서 DW에 적재하는데 드는 시간과 노력이다. DWA는 이 경우에도 어느 정도 혹은 상당히 속도와 시간의 비용을 줄일 수 있다.

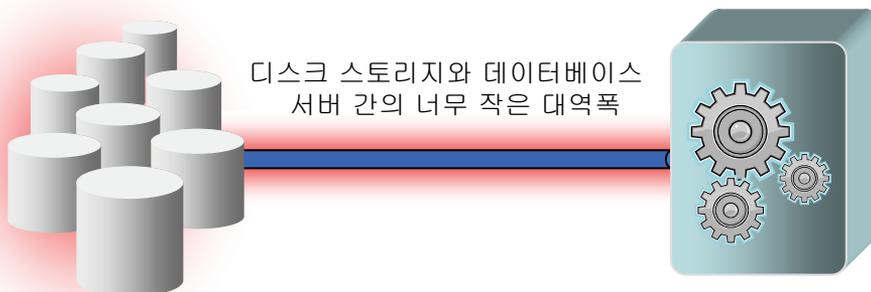
DWA는 DW에 있어 하나의 논리적 진화이자 물리적 진화이다. 더 빠른 정보 및 분석과 함께 더 많은 자원을 원하는 사용자들 그리고 결코 끝나지 않는 백로그(backlog)와 투두(to do) 리스트를 가진 IT 조직 모두에게 DWA는 하나의 신속한 솔루션이 될 수 있다. 더 많은 데이터와 더 빠른 처리를 원하는 요구가 확산되면 될수록 고성능과 확장의 경제성을 제공하는 DWA를 검토할 기회 또한 더욱더 늘어날 것이다. 궁극적으로 DW 구현은 목적이 아니라 하나의 수단이다. 그 목적은 다름 아니라 비즈니스 요구를 즉각적으로 지원하는 솔루션을 제공하는 것이다. DWA 사용과 관련해 대체로 혹은 의심할 여지없이 인정할 수 있는 이점들이 있다 — 낮은 총소유비용, 고성능, 고확장성을 비롯해 용이한 운영관리, 신속한 구현 및 가치 실현, 고가용성이 그것이고, 부록으로 원-스톱-샵(one-stop-shop)을 통한 네 가지 단일화 — 단일 목적, 단일 패키지, 단일 설치, 단일 업체 — 이점이 추가된다.

## 1.2 Exadata 관점

### 1.2.1 스토리지 병목 현상

오라클이 보는 오늘날의 데이터베이스 성능은 결국 스토리지에 의해 제한되는 것이다. 이유는 스토리지 병목 현상 때문이다(그림 1-14); 스토리지 시스템이 스토리지에서 데이터베이스 서버로 가는 데이터 대역폭을 제한하고, 스토리지 어레이(storage array) 내부와 SAN(storage area network) 상에서 데이터 정체(traffic congestion)가 발생하며, 물리적(기계적 회전)의 디스크 속도로 인해 랜덤(random) I/O 병목 현상이 일어난다. 결과적으로 데이터 대역폭 제한은 DW 성능을 제한하고, 랜덤 I/O 병목은 OLTP 성능을 제한한다.

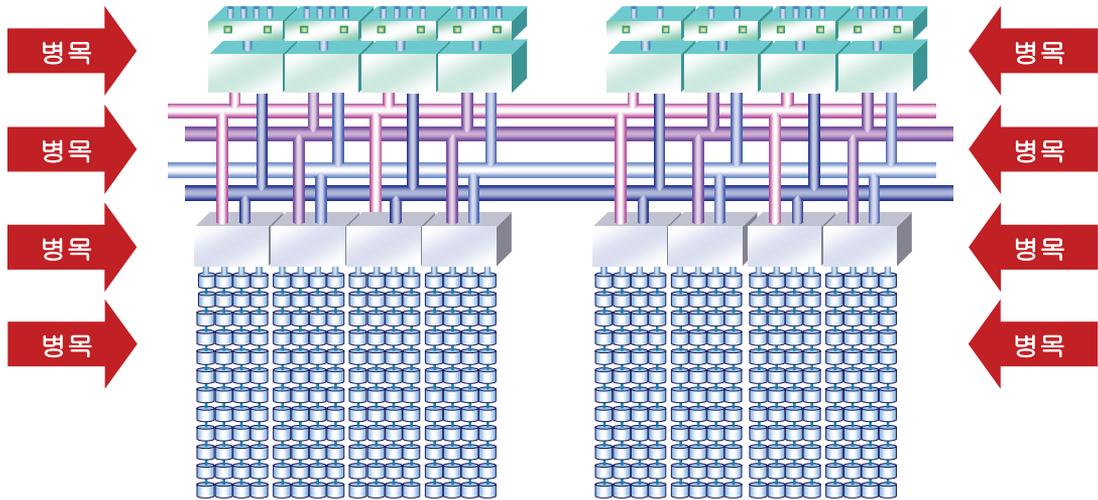
<그림 1-14 데이터 대역폭과 스토리지 병목현상>



데이터베이스 사용에 있어, 종종 디스크에서 데이터베이스 서버로 가는 데이터 이동을 제한하는 병목 현상이 발생한다고 오라클은 지적한다. 오라클에 따르면, 이는 전통적 스토리지 제품들이 좁고 제한된 데이터베이스-스토리지 인터페이스를 제공하기 때문으로, 이로 인해 데이터베이스 I/O 경로 상에서 데이터 대역폭을 제한하는 많은 병목 현상이 발생하게 되고, 따라서 데이터베이스 사용 전반의 성능이 제한 받게 된다. 일반적으로 대용량의 데이터를 처리하는 범용 데이터베이스 서버가 (스토리지에서 데이터베이스까지) 데이터를 적정 속도로 전달하기 위해 필요한 대역폭을 제공하려면 많은 SAN HBA(Host Bus Adapter)를 확보해야 한다는 것이다. 그렇지만, 오라클에 의하면, 적정 성능을 내기 위해 필요한 많은 수의 HBA를 서버가 지원할 수 없거나 제공하기에는 너무 많은 비용이 드는 경우가 허다하고, 필요한 대역폭 및 이중화를 제공하기 위한 SAN switch의 비용과 복잡성 또한 크게 증가한다. 게다가 대형 스토리지 어레이는 (자체의) 수백 개의 디스크를

위한 적정 대역폭을 제공할 수 없기 때문에, 디스크의 실제 성능은 원래 가능한 수준 아래로 끌어 내려진다. 디스크 성능은 디스크로 가는 FCL(fibre channel loop, fibre channel: 엔터프라이즈 스토리지 환경에서 SAN을 위한 스토리지 네트워킹에 주로 사용되는 표준 네트워크 기술) 상에서 또 스토리지 어레이의 처리 능력에 의해 제한된다. 결국 스토리지 병목 현상은 디스크 I/O, 디스크 어레이, FC, SNA HBA 등에서 두루 발생하게 된다(그림 1-15, 표 1-1).

<그림 1-15 스토리지 병목현상>



“전통적 스토리지는 back-end disk loop, cache, controller, front-end channel, SAN link, HBA에서 병목 현상이 발생한다.”

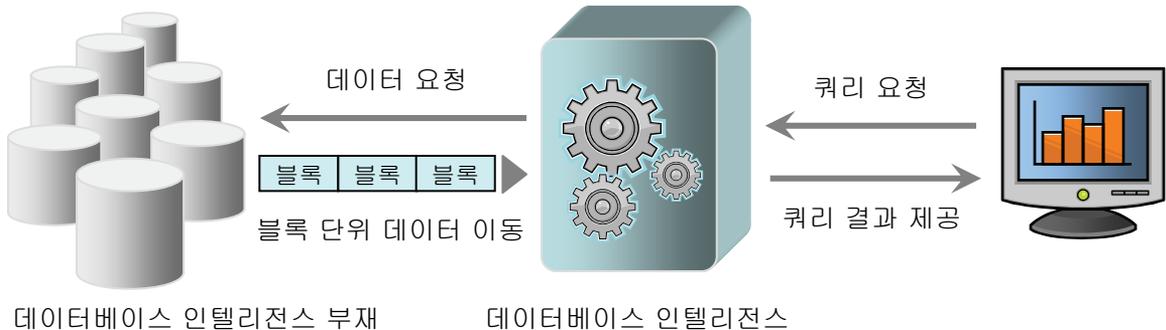
<표 1-1 데이터 대역폭 문제와 스토리지 병목 현상>

데이터 대역폭 문제	스토리지 병목 현상
<ul style="list-style-type: none"> <li>● 디스크 스토리지와 데이터베이스 서버 간의 데이터 대역폭 병목으로 인해 종종 데이터 이동이 제한된다.</li> <li>● 디스크와 서버 사이의 파이프가 데이터 크기에 비해 너무 느리다(10배에서 100배까지).</li> </ul>	<ul style="list-style-type: none"> <li>● 서버 내 병목: 제한된 fibre channel HBA 수</li> <li>● fibre channel loop 상의 병목</li> <li>● SAN 구성의 불충분 및 복잡성</li> <li>● 스토리지 어레이 내부의 병목</li> <li>● 물리적 디스크 I/O 병목</li> </ul>

참조: Oracle

한편, 전통적 스토리지 장치의 또 다른 문제는 데이터베이스가 스토리지 상에 존재하는지를 인식하지 못한다는 것이다. 따라서 데이터베이스 인식(database-awareness) 차원의 어떠한 I/O 혹은 SQL도 처리할 수 없다. 그러니까 데이터베이스가 필요로 하는 데이터를 스토리지에 요청할 때 스토리지로부터 돌아오는 것은 데이터베이스 쿼리의 결과가 아니라 데이터 블록(data block)들이다(그림 1-16). 다시 말해, 전통적 스토리지는 요청 받은 특정 row와 column을 인식하는 데이터베이스 인텔리전스(database intelligence)를 갖고 있지 못하기 때문에, 데이터베이스를 위한 I/O 처리 시 데이터베이스 쿼리가 요구하는 것보다 훨씬 더 많은 데이터를 돌려보내면서 대역폭을 낭비하게 된다(그런데 이 문제는 스토리지 장치와 별개로 데이터베이스 자체의 데이터 저장 구조(row-oriented 혹은 column-oriented)와도 밀접히 관련된다). 그래서 오라클은 스토리지 병목 현상을 극복하기 위해 몇 가지의 근본적 조치가 취해질 필요가 있다는 결론에 이른다. 그것은 바로 데이터 대역폭과 데이터 이동량 그리고 물리적 디스크 I/O에 관한 것이다.

<그림 1-16 종래의 데이터베이스>

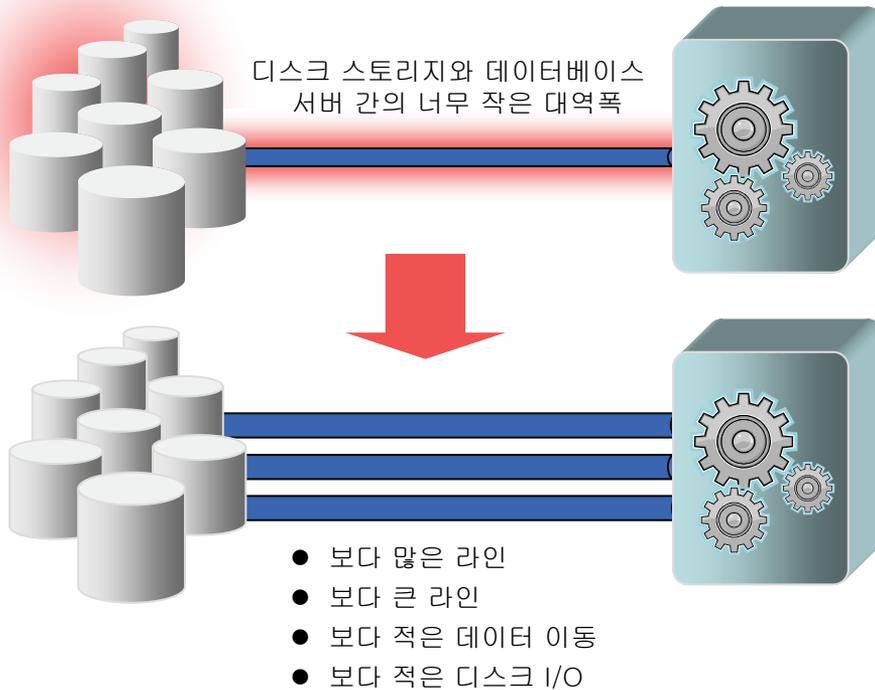


## 1.2.2 Exadata 아이디어

Exadata 아이디어는 기본적으로 데이터베이스 I/O의 최적화에 관한 것으로, 데이터베이스 성능 혹은 데이터베이스 I/O 문제의 근본 원인이 데이터베이스 서버와 스토리지 간의 대역폭이 너무 좁은 데에 기인한다는 인식에서 출발해(Exadata V1), 랜덤(random) 디스크 I/O의 최소화로 그 관점이 확대되었다(Exadata V2). 그런데 이 Exadata 아이디어는 기존 Oracle Database(구조와 기능)를 그대로 유지한다는 전제 하에서 문제를 해결하려는 하나의 대안으로 나온 것이라는 점을 특별히 지적할 필요가 있다(이 기본 전제는 Exadata를 정확히 이해하기 위한 하나의 단초가 된다).

아무튼 Exadata 아이디어란 바로 너무 좁은 대역폭 문제를 해결하기 위해, 보다 많은 파이프를 추가하고, 보다 큰 파이프를 만들고, 파이프에 보다 적은 데이터를 실어 보내고, 그리고 랜덤 디스크 I/O를 최소화하는 장치 및 기술을 제공하는 것이다(그림 1-17, 표 1-2). 따라서 Exadata는 데이터베이스 성능을 제한하는 데이터베이스 I/O의 문제를 네 가지 핵심 차원 — 파이프 수, 파이프 크기, 스토리지와 데이터베이스 서버 간의 데이터 이동량, 랜덤 디스크 I/O — 에서 다룬다.

<그림 1-17 데이터 대역폭 병목현상에 대한 솔루션>



<그림 1-2 Exadata 이이디어>

Exadata 관점		Exadata 아이디어
데이터 대역폭	파이프 수	● 보다 많은 파이프: 스토리지에서 데이터베이스 서버로 데이터를 실어 보내는 파이프 수를 더 늘린다. 또한 데이터베이스 크기의 증가에 비례해 데이터 파이프가 용이하게 확장될 필요가 있다.
	파이프 크기	● 보다 큰 파이프: 스토리지에서 데이터베이스로 가는 개별 파이프들을 더 크게 만든다.

<그림 1-2 Exadata 아이디어>

Exadata 관점	Exadata 아이디어
데이터 이동량	<ul style="list-style-type: none"> <li>● 보다 적은 데이터 이동: 스토리지와 데이터베이스 서버 간에 보다 적은 양의 데이터가 이동하도록 데이터 처리 일부를 스토리지 자체에서 수행한다.</li> </ul>
랜덤 디스크 I/O	<ul style="list-style-type: none"> <li>● 디스크 I/O 최소화: 랜덤 디스크 I/O를 최소화하는 장치 및 기술을 제공한다.</li> </ul>
Oracle Database	<ul style="list-style-type: none"> <li>● Oracle Database 유지: 데이터베이스(구조와 기능)와 옵션이 변경되지 않고 유지된다.</li> </ul>

참조: Oracle

### 1.2.3 Exadata 솔루션

Exadata가 Oracle Database의 구조와 기능을 유지하면서 데이터 대역폭 및 스토리지 병목 현상을 극복하기 위해(네 가지 관점의 Exadata 아이디어를 실현하기 위해) 제시하는 솔루션의 핵심은 바로 지능적 스토리지 서버이다. 오라클은 웹(web) 서버, 애플리케이션 서버, 데이터베이스 서버와 마찬가지로 스토리지 서버 역시 별도로 존재해야 한다고 본다(그림 1-18).

<그림 1-18 Exadata 솔루션>



Exadata 솔루션은 Exadata 스토리지 서버(소프트웨어 포함)와 Oracle Database 11gR2 데이터베이스 서버 그리고 고속의 interconnect를 하나의 데이터베이스 머신으로 구성하는 것이다. 그럼으로써 Exadata 솔루션은 스토리지 서버의 초병렬 구조를 통해 더 많은 파이프를 제공하고, 고속의 interconnect를 통해 더 큰 파이프를 제공하고, 스토리지 서버 내의 데이터 처리를 통해 보다 적은 데이터를 (데이터베이스 서버로) 실어 보내고, Exadata 플래시 캐시(flash cache)를 통해 랜덤 디스크 I/O를 최소화하는 네 가지 방식으로 스토리지 병목 문제를 다룬다(표 1-3).

<표 1-3 Exadata 솔루션>

Exadata 아이디어	Exadata 솔루션	
Exadata는 보다 많은 파이프를 가진다.	초병렬 구조, 초병렬 스토리지 그리드	<ul style="list-style-type: none"> <li>● 초병렬 구조(massively parallel architecture) 기반: Exadata는 데이터베이스 서버와 스토리지 서버 간에 보다 많은 데이터를 보다 빨리 전달하도록 보다 많은 파이프를 제공한다.</li> <li>● 초병렬 스토리지 그리드(massively parallel storage grid): 스토리지 cell(서버)이라는 모듈 단위의 빌딩 블록(building block)들로 구성된다.</li> <li>● 데이터 대역폭은 용량과 함께 확장된다.</li> </ul>
Exadata는 보다 큰 파이프를 가진다.	고속의 Interconnect	<ul style="list-style-type: none"> <li>● 보다 큰 파이프(InfiniBand interconnect) 사용: 데이터베이스 서버와 스토리지 서버 간에 대단히 큰 대역폭을 제공한다(대단히 짧은 대기 시간과 함께 종래의 SAN HBA가 제공하는 것보다 10배가 큰 40Gb/초의 대역폭을 제공한다; fibre channel보다 10배 빨리 데이터를 전송한다).</li> </ul>
Exadata는 파이프를 통해 보다 적은 데이터를 보낸다.	Exadata 스토리지 서버 내 데이터 처리(데이터 필터링, 데이터 압축)	<ul style="list-style-type: none"> <li>● 작업부하 분산: 데이터 집약적 처리는 Exadata 스토리지에서 실행된다.</li> <li>● 보다 적은 데이터 이동: 데이터베이스 서버의 CPU 부하를 덜어내는 한편, 서버로 반환되는 데이터를 획기적으로 줄이기 위해 쿼리 처리의 일부가 Exadata 스토리지 내로 이동된다. 따라서 Exadata는 데이터베이스 인식(database-aware)을 통해 쿼리 처리에 필요한 데이터만 데이터베이스 서버로 보낼 수 있다. 그 결과 데이터베이스 서버와 스토리지 서버 간에 보다 적은 데이터가 이동하게 된다.</li> <li>● 데이터 압축(Hybrid Columnar Compression): 10배의 비용 절감과 상응한 성능 향상을 제공한다(archive 경우는 50배의 압축률을 제공한다).</li> </ul>
Exadata는 랜덤 디스크 I/O 발생을 최소화한다.	Exadata 스토리지 서버 내 플래시 캐시	<ul style="list-style-type: none"> <li>● 플래시 캐시 사용: (지능적) 플래시 캐시를 통해 데이터베이스 처리와 관련된 랜덤 디스크 I/O 병목 현상을 해결하고 빠른 응답시간을 제공한다(랜덤 I/O 성능을 20배 향상시킨다).</li> </ul>

참조: Oracle

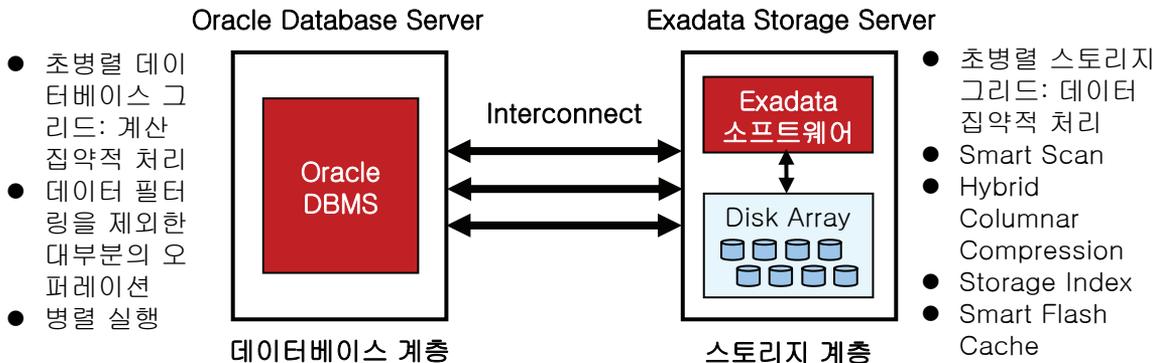
## 1.3 Exadata 차별성

### 1.3.1 Exadata 기술

Exadata는 본질적으로 Oracle Database를 위한 스토리지의 역할을 새로이 부여하고, Oracle Database와 스토리지 간의 관계를 재설정한다. 이 특화된 스토리지(서버)를 오라클은 지능적(intelligent, smart) 혹은 프로그래머블(programmable)이라고 규정한다(이는 Netezza의 FPGA를 연상시킨다). 나아가 Exadata는 고성능 및 대용량의 표준 하드웨어와 병렬 처리를 최대한 활용하고, 곧잘 오라클 성능 문제를 유발시키는 사용자에게 의한 시스템 구성의 오류를 방지하기 위해 사전 구성(pre-configured) 방식으로 시스템을 제공한다. 결과적으로 Exadata 시스템은 이전의 Oracle 시스템에 비해 향상된 데이터 처리 효율 및 성능을 제공할 수 있다.

Exadata 시스템은 Oracle Database를 위해 하드웨어와 소프트웨어가 완전히 통합된 데이터베이스 플랫폼으로, Exadata Storage Server와 Oracle Database Server 그리고 interconnect로 구성된다. Exadata Storage Server는 Oracle Database를 위해 최적화된 지능적 스토리지로, Exadata 시스템의 근간이 된다. 이 Exadata 시스템의 가장 큰 특징은 종래의 스토리지가 데이터베이스 인식 스토리지 서버로 대체됨으로써, 쿼리(트랜잭션) 처리가 데이터베이스 서버와 스토리지 서버로 이원화 되는 2계층 구조라는 점이다(그림 1-14).

<그림 1-19 Exadata 2-계층 아키텍처>



여기서 스토리지 계층은 데이터 필터링, 데이터 압축, 데이터 캐싱(caching)을 포함한 데이터 집약적 처리를 수행하고, 데이터베이스 계층은 필터링을 통해 걸러진 데이터에 대해 다른 모든 오퍼레이션(집계, 정렬 등), 즉 계산 집약적 처리를 수행하며, 각 계층은 서로 interconnect를 통해 연결된다.

Exadata는 데이터 양의 급증에 따른 비용 증가와 성능 저하 등의 근본 문제를 데이터베이스 플랫폼 차원에서 해결하기 위해 개발된 것으로, 무엇보다 대용량의 데이터를 대단히 빨리 처리할 수 있는 아키텍처 구현에 중점을 두었다. 따라서 Exadata는 초병렬 그리드 및 고속의 interconnect를 통해 서버와 스토리지를 연결하는 네트워크 파이프의 크기와 수를 늘리고, 스마트 플래시 캐시(Smart Flash Cache)라는 고속의 대용량 플래시 메모리를 채택한다. 또한 스마트 스캔(Smart Scan)이라는 오라클 고유의 지능적 쿼리 기술, 스토리지 인덱스(Storage Index)라는 불필요한 데이터의 사전 제거 기술, HCC(Hybrid Columnar Compression)라는 보다 효율적인 데이터 압축 기술을 적용하는 등, 새로운 시스템 아키텍처와 기술을 통해 데이터 및 I/O 병목 문제를 해결하고 처리 속도를 향상시키고자 한다.

이 가운데 특히 스마트 스캔은 대용량의 데이터 처리 과정에서 발생하는 네트워크나 프로세서의 병목 문제를 보다 근본적으로 제거하려는 Exadata의 핵심 소프트웨어 기술이다. 종래에는 쿼리 실행 시, 모든 데이터를 데이터베이스 서버로 불러들이는 이유로 해서 곧잘 네트워크나 프로세서의 병목 문제가 불거졌다. 그런데 스마트 스캔을 적용하면 스토리지로부터 필요한 데이터만 가져올 수 있어 대용량의 데이터를 매우 빠르게 처리할 수 있다.

스토리지 인덱스는 스토리지에 저장된 칼럼의 최소값과 최대값을 디스크 용량 전체에 걸쳐 추적함으로써, 쿼리(WHERE 절)에서 지정된 데이터(칼럼 값 혹은 범위)를 포함하지 않는 데이터 블록(Storage Region)을 I/O 대상에서 제외한다. 따라서 불필요한 데이터에 대한 I/O를 사전(데이터 스캔 전에)에 방지할 수 있다; 스토리지 인덱스는 스마트 스캔에 앞서 수행된다.

HCC는 데이터 압축을 최적화하기 위해 (관계형 데이터베이스 데이터에 대한 새로운 유형의 데이터 저장 방식인 “hybrid columnar”의 채택을 통해) 칼럼 기준의 새로운 데이터 압축 기술을 적용함으로써, 데이터가 차지하는 스토리지 공간을 줄이고 상응하는 성능 향상(I/O 감소와 데이터 대역폭 절약) 효과를 제공한다(오라클은 10배의 데이터 압축률을 기준으로 할 때, DW table의 크기는 10배로 줄고 쿼리는 10배 더 효과적으로 실행될 수 있다고 주장한다).

Exadata의 주요 특성인 Exadata 스마트 플래시 메모리는 메모리 카드 형태로 Exadata Storage Server에 추가돼, 새로운 메모리 계층(CPU-캐시-메모리-플래시-디스크)을 구현한다(Exadata 스마트 플래시 캐시는 오라클이 최근 10년 간의 최대 기술들 중의 하나로 꼽는 Sun FlashFire에서 나왔다). 하나의 Exadata 시스템(full-rack 기준)은 총 5TB의 플래시를 표준 용량으로 제공하는데, 여기에 새로운 데이터 압축 기술(HCC)이 적용될 경우, 50TB(5TB x 10배 압축률)로 확장된다고 오라클은 주장한다.

SQL이 Oracle에 전달되면, Oracle Database는 지능적으로 처리 일부를 스토리지 서버로 덜어낸다. 그러면 Exadata Storage Server는 데이터 블록 전체가 아니라 쿼리에 필요한 로(row)와 칼럼만을 데이터베이스로 돌려보낸다. 이 스마트 스캔 기능을 통해 시스템은 데이터 이동량을 크게 줄일 수 있다. 이 줄어든 데이터 흐름은 다시, 쿼리를 실행하고 결과를 제공하는데 걸리는 전체 처리 시간을 줄인다. 여기서 HCC는 높은 압축률(Query Mode는 10배, Archive Mode는 50배)을 제공함으로써 대량의 데이터가 효과적으로 스캔 될 수 있도록 한다. 스토리지 인덱스는 스캔 프로세스에서 적절한 데이터만 스캔 되도록 보장하기 위해 불필요한 데이터(row)를 사전에 제거하는 가장 빠른 방법을 제공한다. 스토리지 인덱스를 통해 불필요한 로를 제거함으로써 스캔 되는 로 수를 크게 줄일 수 있다.

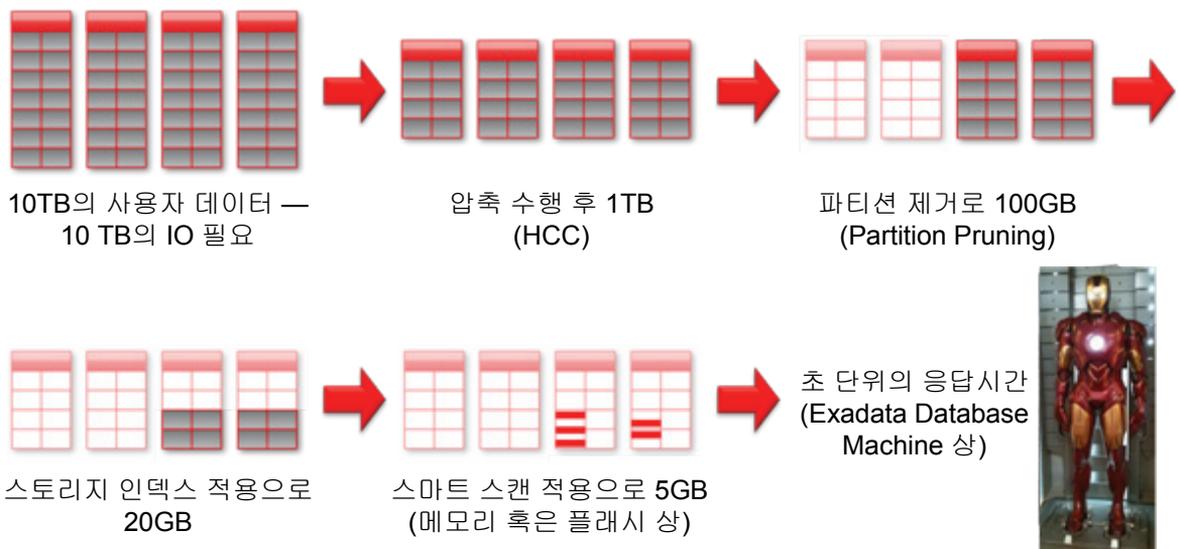
Exadata 스마트 스캔, HCC, 스토리지 인덱스는 각기 Exadata 성능 퍼즐을 맞추는 대단히 중요한 부분들로, 상호 연동을 통해 상승 효과를 만들어내고 공동으로 Exadata 스토리지 솔루션을 완성한다(그림 1-20). 그렇지만 스토리지와 데이터베이스 간의 좁은 대역폭 접속은 이들 이점을 한번에 날려버릴 것이다. 이 문제를 방지하기 위해 Exadata는 Exadata 스토리지 서버와 데이터베이스 서버 간에 인피니밴드(InfiniBand) 접속을 제공한다. 상대적으로 새로운 기술인 InfiniBand는 전통적 이더넷(Ethernet)보다 훨씬 더 큰 대역폭을 제공한다. Exadata는 이 InfiniBand 상에서 스토리지와 데이터베이스 서버 간의 통로를 종래보다 더 많이 추가할 수 있다. 결국 Exadata Storage Server는 불필요한 데이터를 읽고 전송하는데 시간을 소모하지 않고, 필요한 데이터를 더 크고 더 많은 파이프를 통해 전송한다. 여기에 지능적 스마트 플래시가 디스크 I/O의 효율적 제거에 가세하고, 디스크 I/O와 데이터 통행을 더한층 줄이기 위해 Oracle Database 11gR2는 인-메모리(In-Memory) 병렬 쿼리 실행을 제공한다. 이것이 Exadata의 성능 비결, Exadata 시크릿이다.

오라클 RAC(Real Application Cluster)는 여전히, 복수의 서버 노드로 확장하는 스케일-아웃

(scale-out) 컴퓨팅을 위한 기반이다. 오라클 그리드 컴퓨팅(grid computing)은 이 RAC에 의해 가능한 것으로, RAC 구현의 중심에 있다. 또 스토리지 관리를 단순화하고 자동화하는 ASM(Automatic Storage Management)은 견고하고 유연한 스토리지 하부구조의 기반으로, (SMP는 물론이고) RAC를 스토리지 차원에서 지원한다.

한편 Exadata 시스템은 랙(rack) 최적화된 시스템 구성(quarter-rack, half-rack, full-rack) 중에서 하나를 선택할 수 있기 때문에, 서버와 스토리지가 통합된 솔루션을 (캐비닛 내에 장착되는) 보다 작은 단위로 시작할 수 있고, 필요에 따라 단위를 추가함으로써(혹은 새로운 랙을 추가함으로써) 용량과 성능을 함께 확장해 나갈 수 있다. Exadata는 확장 가능한 데이터베이스 및 스토리지 엔진과 함께 RAC 그리드 컴퓨팅 소프트웨어와 ASM 스토리지 관리 소프트웨어, Linux나 Solaris OS를 사용해, 스케일-업(scale-up) 시스템(SMP)이 아닌, 스케일-아웃 시스템을 제공한다. 그런데 한가지 중요한 점은, 이러한 Exadata 시스템 상에서 작동하기 위해 애플리케이션이나 데이터베이스가 수정될 필요는 없지만, Exadata의 특성과 기능을 십분 활용하기 위해 애플리케이션과 데이터베이스 레벨에서 적절한 디자인과 구현이 고려될 필요가 있다는 것이다(“2부 Exadata 구현”참조).

<그림 1-20 Exadata 시너지>



“데이터는 10배 줄어들고, 스캔은 2000배 빨라진다.”

### 1.3.2. Exadata 용도

오라클은 Exadata가 모든 작업부하에 대해 작동하고 확장하는 유일한 데이터베이스 머신이라고 주장한다. 오라클에 의하면, Exadata는 강력한 성능을 기반으로 모든 데이터 관리, 즉 데이터 웨어하우징, OLTP 애플리케이션, DBMS 통합(consolidation) 모두에 적합한 플랫폼이다(표1-5). 오라클 DBMS를 사용하는 곳이라면 누구나 Exadata를 도입해 보다 쉽게 DW를 구축할 수 있고, ERP, CRM과 같은 OLTP 애플리케이션을 운영할 수 있으며, 다양한 DBMS를 단일 환경으로 통합할 수 있다는 것이다. 그러나 Exadata 상에 OLTP와 DW를 통합할 수 있다고 해서, Exadata 상에 OLTP와 DW를 통합해야(만) 하는 것은 아니다. 이는 간단한 일이 아니고, 현재로서는 이러한 통합에 이에 대해 제시할 수 있는 베스트 프랙티스도 없기 때문이다.

<표 1-5. Exadata 용도 별 성능 이점>

<p><b>DW를 위한 최적의 성능</b></p>	<p>데이터 집약적 쿼리 처리와 데이터 마이닝 스코링(data mining scoring) 등을 데이터베이스 서버로부터 덜어내 보다 데이터 가까이로 가져옴으로써, 쿼리 성능과 동시(concurrency) 처리 모두를 향상시킨다.</p>
<p><b>OLTP를 위한 최적의 성능</b></p>	<p>‘hot data’를 Sun FlashFire에 그대로 옮김으로써 디스크 랜덤 I/O 병목 문제를 해결한다(비즈니스 애플리케이션에 대해 10배 빠른 I/O 응답시간과 10배 적은 디스크 사용을 가능케 한다).</p>
<p><b>혼합 작업부하를 위한 최적의 성능</b></p>	<p>초병렬 서버 그리드(massively parallel server grid) 상에 데이터 웨어하우징, 트랜잭션 프로세싱, 혼합 작업부하를 통합함으로써, 제삼자 제공의 특화된 하드웨어, 보안, 관리 솔루션의 비용 부담을 없앨 수 있게 한다.</p>

출처: Oracle

Exadata 시스템에서 데이터베이스 통합이란 Exadata 시스템 내에 복수의 Oracle Database를 둘 수 있고, Exadata 시스템 내의 모든 Oracle Database는 스토리지 서버와 데이터베이스 서버를 공유할 수 있다는 의미이다. 따라서 데이터베이스, 스토리지, 데이터베이스 서버가 Exadata 시스템 내로 모두 통합될 수 있다. 이 때 Exadata는 자원 할당, 작업 우선순위, 작업부하 균형을 관리함으로써, 모든 애플리케이션을 위한 고성능 플랫폼을 제공할 수 있고, 공유 환경 내에서 응답시간을 예측할 수 있으며, 모든 데이터의 요구를 수용할 수 있다고 오라클은 주장한다(표 1-6, 그림

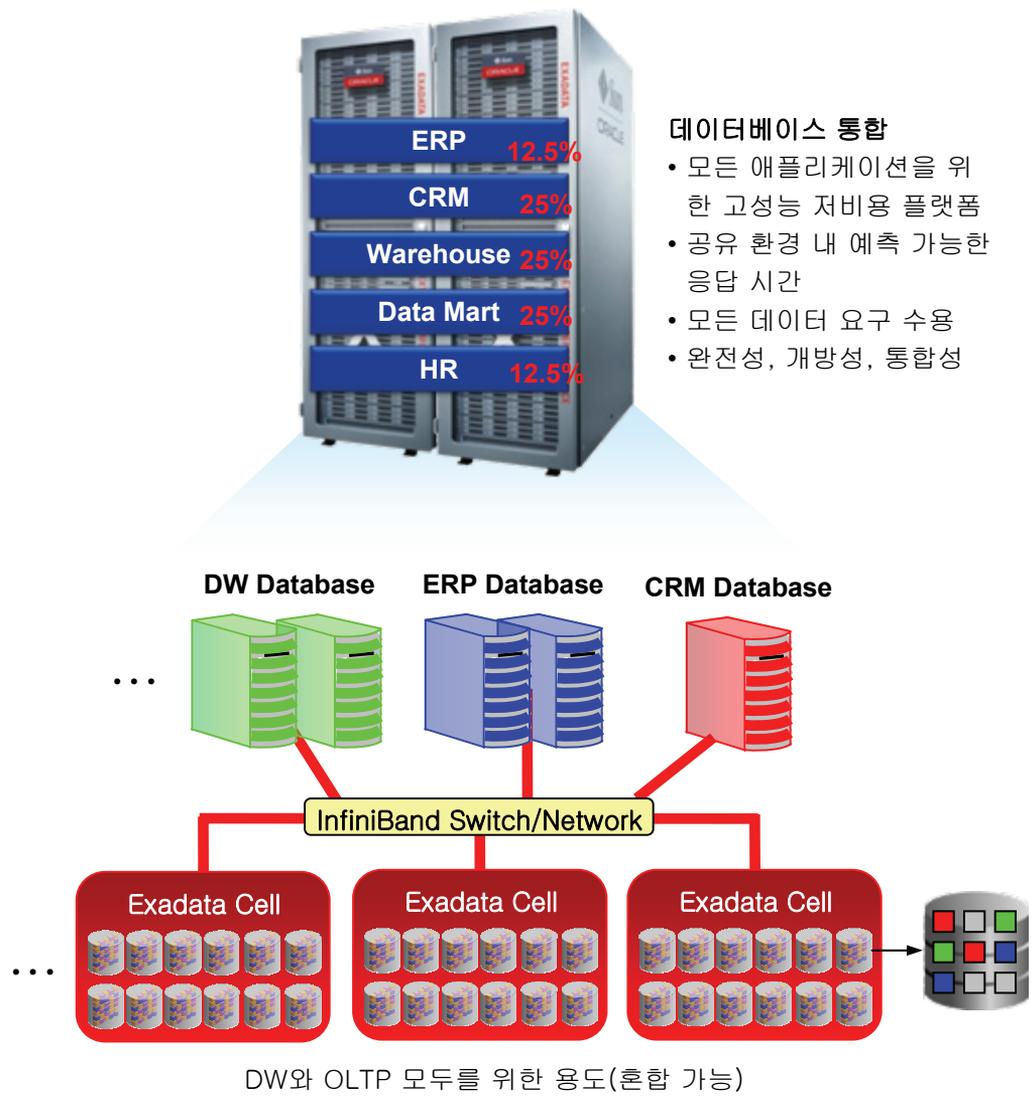
1-21, 표 1-7).

<표 1-6. Exadata 용도>

DW	OLTP	DC
<ul style="list-style-type: none"> <li>● 대량 데이터를 빨리 처리하기 위한 초병렬 대용량 하드웨어                     <ul style="list-style-type: none"> <li>- 스토리지 내 데이터 집약적 처리</li> </ul> </li> <li>● 대부분의 분석 능력 보유                     <ul style="list-style-type: none"> <li>- OLAP, statistics, spatial, data mining, real-time transactional ETL, point query</li> </ul> </li> <li>● 강력한 DW 최적화                     <ul style="list-style-type: none"> <li>- partitioning, bitmap indexing, join indexing, materialized view, result cache</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● 그리드(grid) 상의 실제 비즈니스 애플리케이션 실행</li> <li>● 고유한 장애극복 가능 스케일-아웃 OLTP 데이터베이스                     <ul style="list-style-type: none"> <li>- RAC, Data Guard, Online Operaion(Online Index Rebuild, IOT, Online Table Redefinition 등)</li> </ul> </li> <li>● OLTP에 적합한 장애극복 가능 스케일-아웃 스토리지                     <ul style="list-style-type: none"> <li>- ASM, Exadata</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● 통합 — 하나의 시스템 내 상이한 작업부하들의 혼합                     <ul style="list-style-type: none"> <li>- DW 위주의 대량 데이터 처리</li> <li>- OLTP 위주의 랜덤 업데이트</li> <li>- 멀티미디어 위주의 스트리밍(streaming) 파일</li> </ul> </li> <li>● 고성능에 의한 모든 혼합 작업부하 처리 가능                     <ul style="list-style-type: none"> <li>- 응답 시간 예측 가능</li> </ul> </li> <li>● 모든 작업부하에 대해 작동하고 확장하는 유일한 데이터베이스 머신</li> <li>● 다중-데이터베이스, 다중-애플리케이션, 다중-사용자 환경 내의 응답 시간 예측 가능</li> </ul>

참조: Oracle

<그림 1-21 Exadata 데이터베이스 통합>



출처: Oracle

<표 1-7 Exadata 데이터베이스 통합>

데이터베이스 스토리지 통합	데이터베이스 서버 통합
<ul style="list-style-type: none"> <li>● Exadata와 ASM은 모든 스토리지 서버가 (데이터베이스들에) 공유될 수 있게 한다.</li> <li>● 공유 시스템 구성                     <ul style="list-style-type: none"> <li>- ASM(data striping)은 모든 데이터베이스를 모든 스토리지 상에 분산한다.</li> <li>- hot-spot 저장 공간과 확보되고도 사용되지 않는 저장 공간을 제거한다.</li> <li>- 완전한 스토리지 그리드(성능)가 모든 데이터베이스에 가용하다.</li> <li>- 데이터베이스 혹은 클러스터 수준의 스토리지 보안이 지원된다.</li> </ul> </li> <li>● 예측 가능한 성능                     <ul style="list-style-type: none"> <li>- Exadata I/O Resource Manager는 예측 가능한 성능을 보장하기 위해 I/O의 우선순위를 매긴다(user, job, application, database 레벨에서).</li> <li>- 독립된 스토리지가 불필요하다.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● 복수의 데이터베이스가 복수의 Exadata Database Machine 서버 상에서 운영될 수 있다.</li> <li>● 공유 시스템 구성                     <ul style="list-style-type: none"> <li>- 애플리케이션은 하나 이상의 데이터베이스 서버 상에서 실행되는 하나의 database service에 접속된다 (service는 동적으로 확장, 축소, 이동될 수 있다).</li> <li>- 대형 애플리케이션은 RAC를 통해 노드들 상으로 확장될 수 있다.</li> <li>- 복수의 작은 데이터베이스는 단일 노드 상에서 운영될 수 있다.</li> </ul> </li> <li>● 예측 가능한 성능                     <ul style="list-style-type: none"> <li>- Instance Caging은 복수의 데이터베이스가 동일 노드 상에서 운영될 때, 성능 예측 가능한 CPU 자원(사용)을 제공한다(각 데이터베이스(인스턴스)를 프로세서(CPU core)들의 일부로 제한한다).</li> </ul> </li> </ul>

참조: Oracle

### 1.3.2 Exadata 이점

오라클은 Exadata 사용에 따른 주요 이점으로 성능, 확장성, 비용 절감, 조기 사용, 표준화를 제시한다(표 1-8). 오라클에 의하면, 먼저 빠르고 예측 가능한 성능(DW에 대해 10-100배, OLTP에 대해 20배 성능 향상)은 고속의 인터커넥트를 비롯해 플래시 메모리와 같은 하드웨어 요소, 스마트 스캔, 스토리지 인덱스, HCC와 같은 소프트웨어 기술 그리고 오라클 모범 사례에 근거한 사전 구성(pre-configuration) 등을 통해 실현 가능하다. 전통적 스토리지를 사용하는 전형적 DW에서는 데이터 대역폭 병목 현상으로 인해, 데이터베이스 크기가 증가하면서 대형 table의 스캔 시간

<표 1-8 Exadata 이점>

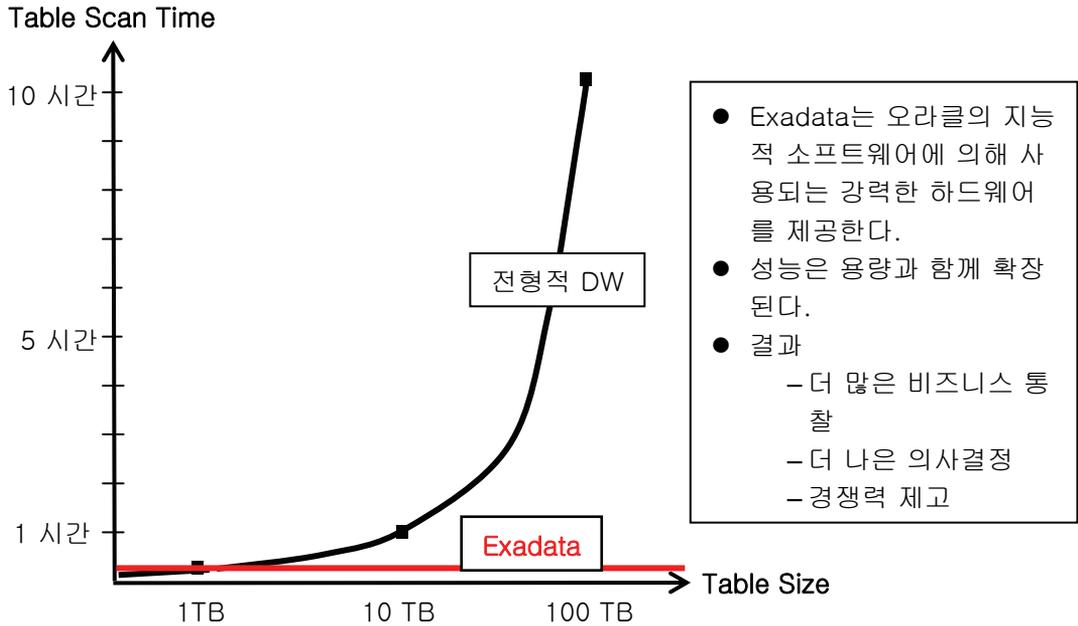
<p><b>성능</b></p>	<ul style="list-style-type: none"> <li>● DW를 위한 획기적 성능 향상(10-100배)                     <ul style="list-style-type: none"> <li>- 보다 많은 파이프: 초병렬 구조</li> <li>- 보다 넓은 파이프: 종래의 스토리지보다 5배 더 빠른 속도</li> <li>- 보다 적은 데이터 전송: 스토리지 내 데이터 처리</li> </ul> </li> <li>● OLTP를 위한 획기적 성능 향상(20배)                     <ul style="list-style-type: none"> <li>- 플래시 메모리: 랜덤 I/O 최소화</li> </ul> </li> </ul>
<p><b>확장성</b></p>	<ul style="list-style-type: none"> <li>● 무제한의 확장성</li> <li>● 데이터 양의 증가에 따른 선형적 성능(데이터 대역폭) 확장</li> <li>● 랙 최적화</li> </ul>
<p><b>비용 절감(낮은 TCO)</b></p>	<ul style="list-style-type: none"> <li>● Oracle의 하드웨어/소프트웨어 통합 솔루션</li> <li>● 기존의 Oracle 투자 보호: 데이터베이스 및 관리 비용 절감</li> <li>● 스토리지 비용 절감</li> <li>● 업계 표준 시스템(구성요소)</li> </ul>
<p><b>조기 사용</b></p>	<ul style="list-style-type: none"> <li>● 사전 구성 시스템: 하드웨어, 소프트웨어가 포함된 하나의 완전한 시스템</li> <li>● 신속한 설치</li> <li>● 단일 지원</li> </ul>
<p><b>표준화</b></p>	<ul style="list-style-type: none"> <li>● 단일 Oracle 환경: OLTP와 DW 시스템 모두를 위한 Oracle 표준화</li> <li>● ETL 복잡성 감소</li> <li>● 기존 애플리케이션에 대한 투명성(변경 불필요)</li> <li>● 기존 방식대로의 데이터베이스 관리(기존 Oracle DBA 스킬 활용)</li> </ul>
<p><b>혼합 작업부하</b></p>	<ul style="list-style-type: none"> <li>● transaction/job 레벨의 품질 서비스</li> <li>● 다용도: DW, OLTP, DBMS 통합</li> </ul>
<p><b>스토리지 절약</b></p>	<ul style="list-style-type: none"> <li>● 데이터 압축</li> </ul>
<p><b>가용성 &amp; 보안</b></p>	<ul style="list-style-type: none"> <li>● 미션 크리티컬</li> <li>● 장애 복구, 백업, Point-in-Time Recovery, 데이터 검증, 암호화</li> </ul>

참조: Oracle

이 급격히 늘어난다. DW는 점점 더 커져만 가기 때문에 성능 문제는 시간이 갈수록 점점 더 악화된다. 대신에 Exadata를 사용하는 DW에서는 DW의 크기가 증가할 때 Oracle Exadata 서버를 더 많이 추가함으로써 데이터 대역폭을 선형적으로 확장할 수 있다(그림 1-22). 이는 table이 더 커

지더라도 쿼리는 여전히 동일 시간 내에 결과를 내놓을 수 있고, 따라서 DW가 계속 성장하더라도 성능 문제를 겪는 대신 더 많은 비즈니스 통찰을 얻을 수 있고 더 나은 의사결정을 내릴 수 있다는 의미이다.

<그림 1-22 Exadata 성능 및 확장성>



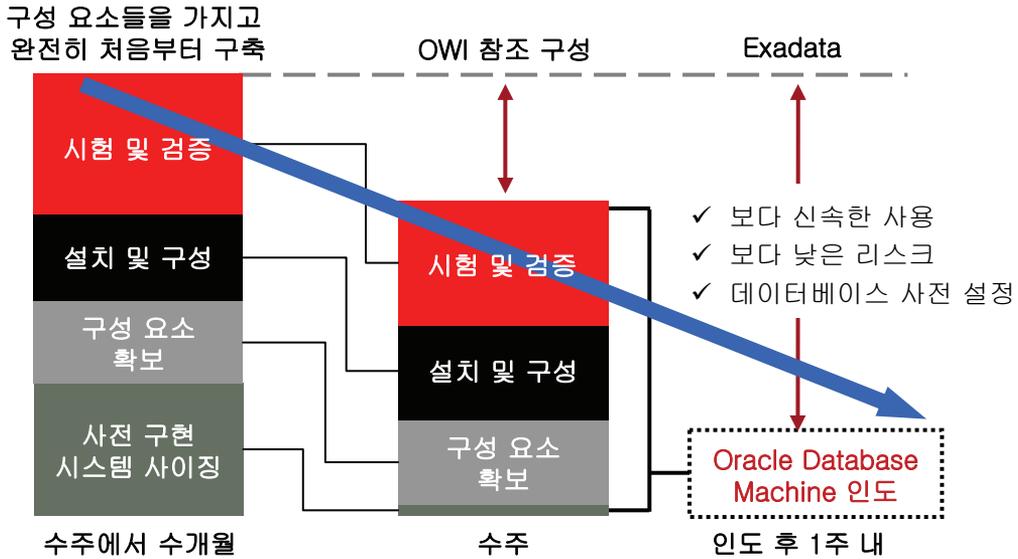
출처: Oracle

확장성은 데이터양의 증가에 따라 성능 및 데이터 대역폭을 선형적으로 향상하기 위해 Exadata 시스템을 스케일-업(하나의 랙 내)과 스케일-아웃(복수의 랙)으로 용이하게 확장해 나갈 수 있다는 것이다. 따라서 속도 X와 용량 Y의 요구를 만족시키는 시스템 Z가 바로 Exadata가 제시하는 것이다. 그것은 일정한 사용자 데이터에 대해 일정한 처리효율을 제공하기 위해 적절한 수의 Exadata Storage Server와 InfiniBand switch를 가지고 Oracle Database가 설치된 시스템이다. 만일 하나의 랙으로 충분치 않으면, 복수의 랙으로 확장해 나간다.

비용 절감은 기존의 Oracle Database 투자를 보호할 수 있고 기존의 데이터베이스 관리 기술을 계속적으로 사용할 수 있는데 따른 운영 비용과 스토리지 비용의 절감 그리고 업계 표준의 아키텍처 채택에 따라 가장 최신 제품(서버, 스토리지, 네트워크 등)을 사용하는 것으로 가능하다. 조기 사용은 하드웨어와 소프트웨어가 포함된 완전한 시스템을 신속히 설치 및 가동할 수 있고, 모든

하드웨어와 소프트웨어에 대해 오라클로부터 단일 지원을 받을 수 있다는 것이다. 이와 아울러 종래의 환경에서 수주에서 수개월이 걸리는 준비 및 설치 기간을 Exadata 환경에서는 불과 일주일 내에 완료할 수 있음으로써, 빠른 투자 회수 및 리스크 최소화 또한 가능하다(그림 1-23).

<그림 1-23 조기 사용 및 리스크 감소>



출처: Oracle

표준화란 Oracle 환경의 표준화를 의미한다. Exadata 시스템은 DW와 OLTP 모두에 대해 Oracle 표준을 적용할 수 있고, 기존의 OLTP 혹은 DW 애플리케이션에 대해 어떤 변경도 요구하지 않는다. Exadata는 업계 표준 하드웨어, Linux(혹은 Solaris), Oracle Database에 기초하기 때문에, 기존의 애플리케이션에 대해 투명하게 작동하고 기존 방식대로 데이터베이스를 관리할 수 있다.

### 1.3.4 Exadata 포지셔닝

Exadata는 주로 Netezza(네티자)와 Teradata(테라데이타)의 특화된 DW 시스템 그리고 확장 가능한 IBM 서버(Power System과 System z 메인프레임)와 경쟁하는 것으로 자리매김된다. 오라클은 특히 (특정 작업부하를 위해) IBM DB2나 Oracle Database를 사용하면서 확장성이 가장 뛰어난 IBM 서버와 가격과 성능으로 경쟁할 의도를 분명히 가지고 있다. 오라클은 또한 적어도 (보

다 오래된 SMP 시스템에서 작동 중인 대형 오라클 데이터베이스(크기가 수TB에서 수PB의 범위에 있는)들 중 일부를 새로운 컴퓨팅 패러다임으로 옮길 강력한 의지를 가지고 있다. 이는 상당히 중요하다 — 왜냐하면 대형 데이터베이스 및 데이터 웨어하우스의 대부분은 현재의 서버 하드웨어를 확장해 나가기를 원하는데, 그 핵심 이유는 확장 가능한 서버, 특히 Unix 서버를 선호하기 때문이다. 따라서 오라클은 이러한 목적에 (대형 데이터베이스를 스케일-아웃 옵션을 가진 표준 서버의 단일 시스템에서 시작할 수 있도록 하는) Exadata를 제공하기를(기존 시스템을 Exadata로 대체해 나가기를) 원한다.

Exadata의 주 고객 대상은 우선적으로 기존의 오라클 고객이 될 것이다. 오라클은 Exadata 고객 확보에 있어, Oracle Database를 사용하며 시스템 포화 혹은 한계에 이르거나 데이터베이스 플랫폼의 표준화에 관심을 가지거나 새로운 플랫폼을 도입할 필요가 있는 대형 고객을 일차적 표적으로 삼을 것이다(오라클은 어떤 특정 DW 플랫폼 고객의 80%는 또한 오라클 고객이기도 하다고 강조한 바 있다). 고객 입장에서는 플랫폼 표준화, 통합의 차원에서 성능과 비용 양쪽의 이점이 보장되기만 한다면 충분히 검토해볼 여건이 마련된 셈이고, 경쟁 업체들 입장에서는 대단히 강력한 경쟁자가 부상한 셈이다. 전형적으로 10TB를 초과해 PB로 들어가는 범위가 Exadata 머신이 성능과 확장성을 확실히 제공하는 영역이라 볼 수 있다.

중요한 점은, 디스크 공유 구조(shared disk architecture) 기반인 오라클이 대형 데이터 웨어하우스에서 그 주요 단점들을 극복하기 위한 노력의 일환으로 내놓은 Exadata가 이제 DW와 OLTP 지원 모두를 겨냥한다는 것이다. 이 두 작업부하는 미션-크리티컬 애플리케이션을 지원하는 확장 가능한 Unix 시스템이나 메인프레임 상에서 곧잘 혼합적으로 나타날 수 있다. 그리고 이러한 유형의 서버는 사용자들이 상당한 관심을 가지고 주시하는 대상이다. Exadata를 위한 최상의 적소로 예상할 수 있는 곳은 데이터베이스 작업부하들을 적극적으로 통합하고 (초)대형 데이터베이스를 보다 빠른 속도로 처리하고자 하는 메가 데이터 센터(mega-data-center)이다. 이에 근거해 오라클은 머지않아 Exadata(류)가 오라클의 주력 제품이 될 것으로 예상한다(그림).

한편 기술 면에서, 얼핏 종래의 단일 구조를 복잡하게 만든 것처럼 보이는 Exadata의 2계층의 구조에 대해, 오라클은 데이터베이스를 스토리지에 보다 가까이 두었다고 주장하는 반면, 오히려 둘 사이의 거리를 더 벌려 놓았다는 비판도 있다. 나아가 Exadata(스마트 스캔)가 전혀 지능적이지도 효율적이지도 않다는 지적도 있고, (그것이 데이터베이스 서버로 보내지는 데이터양을 줄일 수 있는 것은 사실이라 하더라도) 그 구조를 유지하는데 드는 막대한 하드웨어 투입의 효율성에

관한 논란도 있다. 그렇지만 Exadata 시스템 아키텍처, 하드웨어 구성, 비용 등과 관련된 지적이나 논란에도 불구하고, Exadata 시스템의 시장 진입은 일단 성공적으로 평가되는 듯하다. 주요 평가기관에서 Exadata 시장 진입을 포괄적으로 인정하고, 시장(특히 DW 어플라이언스 및 분석 DBMS 업체들)에서는 여러모로 강력한 경쟁자가 부상한 것으로 받아들이는 분위기이며, 차츰 Exadata 고객 사례가 늘고 있기 때문이다. Exadata는 통합 시스템을 제공하고자 하는 오라클 목표의 시작이다. 아마도 앞으로 상당 기간, 막대한 투자가 계속될 것이다.