미래를 준비하는 기업의 스토리지 기술, 퓨어스토리지의 차별성

DEEP LEARNING AND BIG DATA

- ✓ 딥러닝 에서의 FLASHBLADE
- ✓ 빅데이터 에서의 FLASHBLADE
- ✓ FLASHBLADE 아키텍처
- ✓ 도입 사례

PURESTORAGE®

딥러닝 에서의 FLASHBLADE

PURESTORAGE®

DEEP LEARNING TRAINING 에서 STORAGE 요구사항

1. Elastic Performance

Performance grows linearly with data

2. Small to Large Files

Maximum performance for all types of files

3. Random Access

Access to any data with predictable performance

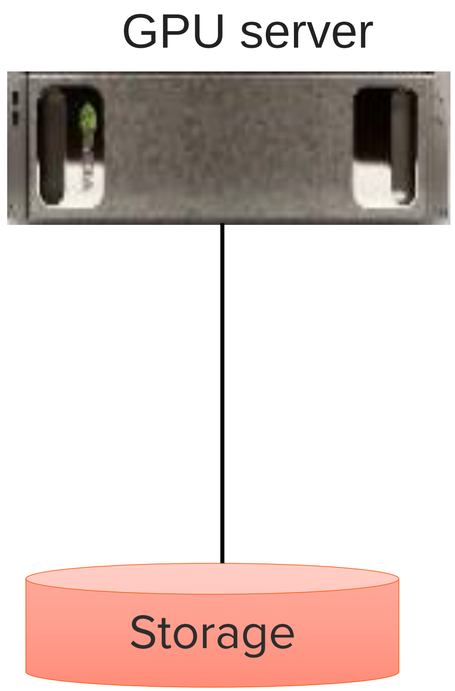
4. Maximum Concurrency

Large number of clients w/o compromising performance

DEEP LEARNING Deploy options

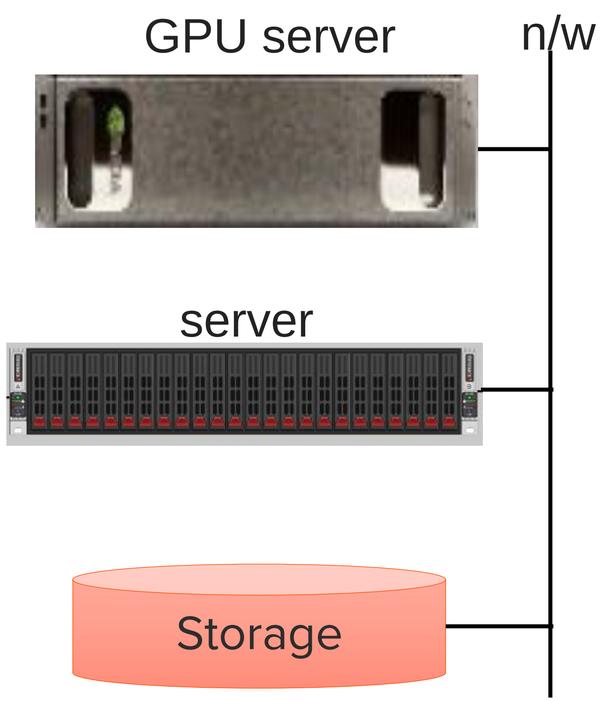
1 초기 단계

- Framework및 모델 개발
- 훈련 데이터: 전처리 안됨, 기성 전처리 데이터 사용(ImageNet)



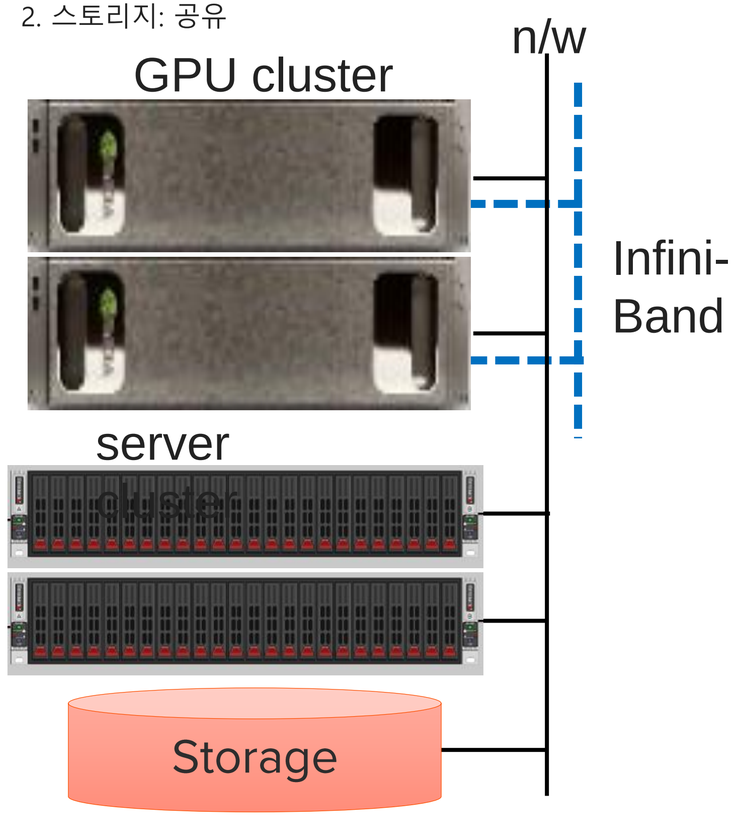
2 활용 단계(표준화)

- 모델 개발, 서비스 개발
- 훈련 데이터: 자체 데이터를 전 처리하여 사용
- 인프라에 대한 표준화 단계
  1. 클러스터 구성: 단일 서버 단위 혹은, 클러스터 단위로 Training (Infini-Band필요 여부 결정)
  2. 스토리지: 로컬 혹은 공유

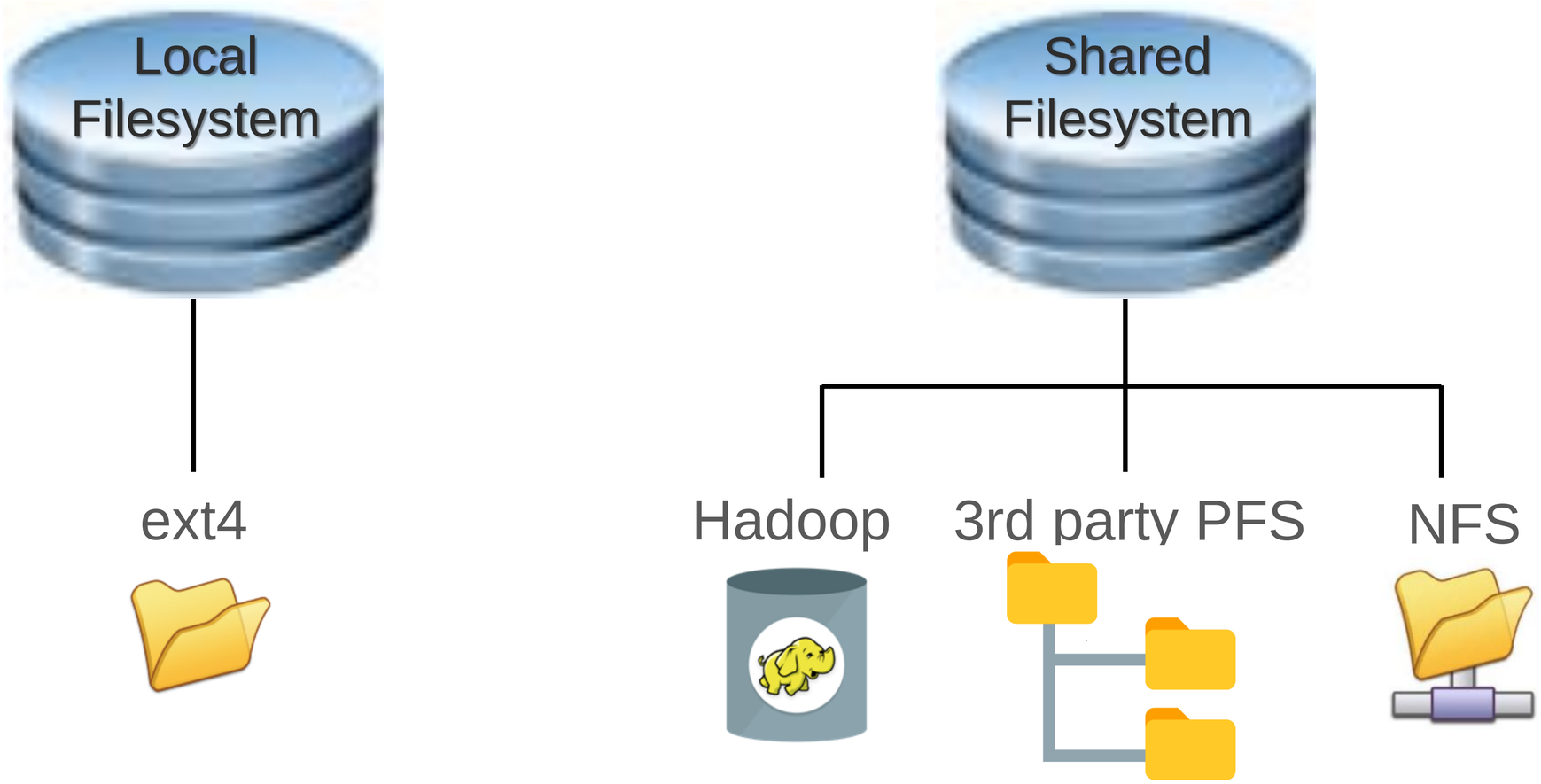


3 활용 단계(엔터프라이즈)

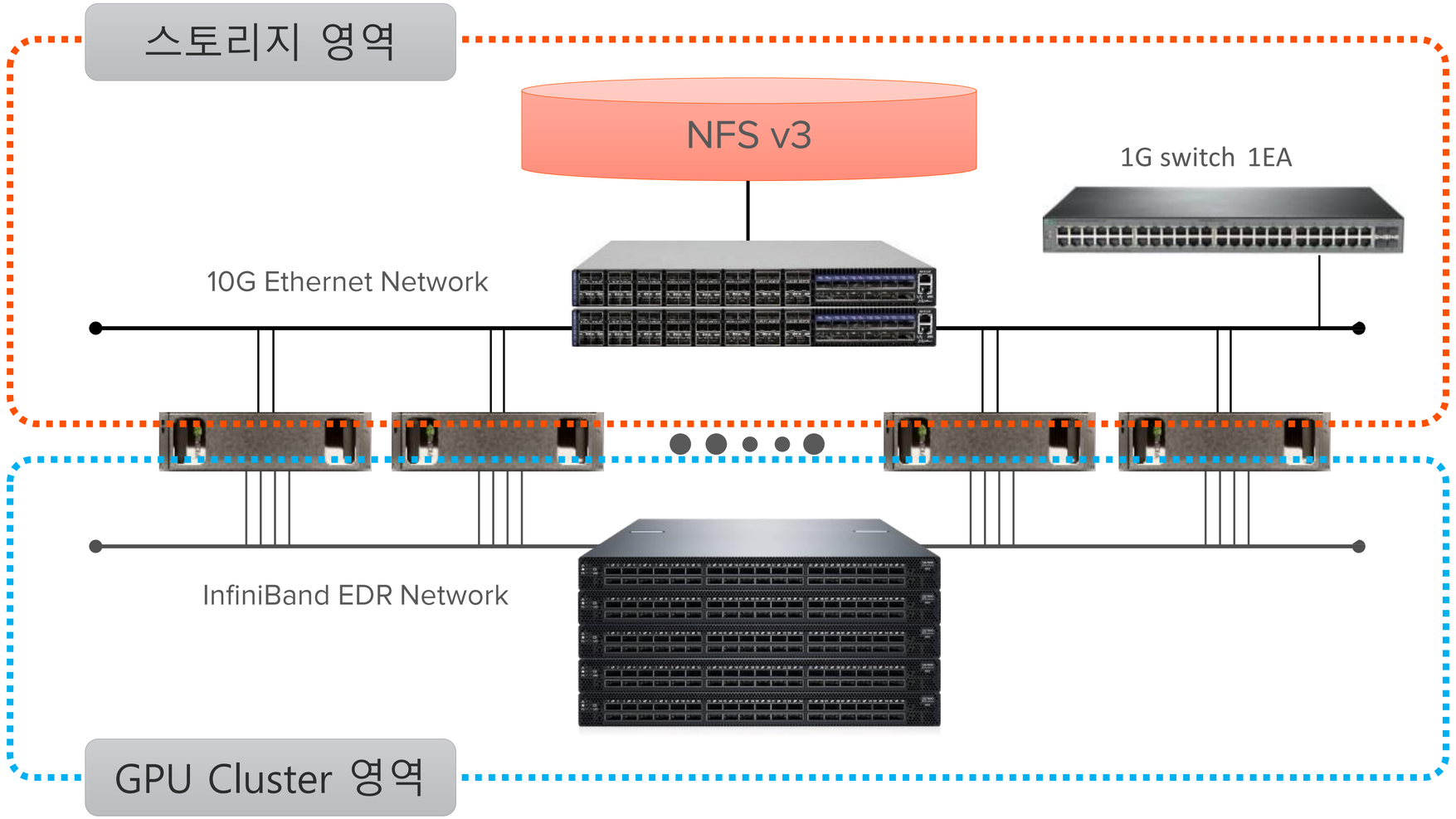
- 모델 개발, 서비스 개발
- 훈련 데이터: 자체 데이터를 전 처리하여 사용
- 엔터프라이즈 인프라 구성
  1. 클러스터 구성
  2. 스토리지: 공유



DEEP LEARNING Storage options



DEEP LEARNING 엔터프라이즈 인프라 구성 참조



DEEP LEARNING 스토리지 구성 가이드

NVIDIA DGX-1 User Guide에 의하면,
Training 데이터 저장을 공유 스토리지로 구성 시 권장 사항은 NFS v3입니다.

3.5. Configuring NFS Mount and Cache

The DGX-1 includes four SSDs in a RAID 0 configuration. These SSDs are intended for application caching, so you must set up your own NFS drives for long term data storage. The following instructions describe how to mount the NFS onto the DGX-1, and how to cache the NFS using the DGX-1 SSDs for improved performance.

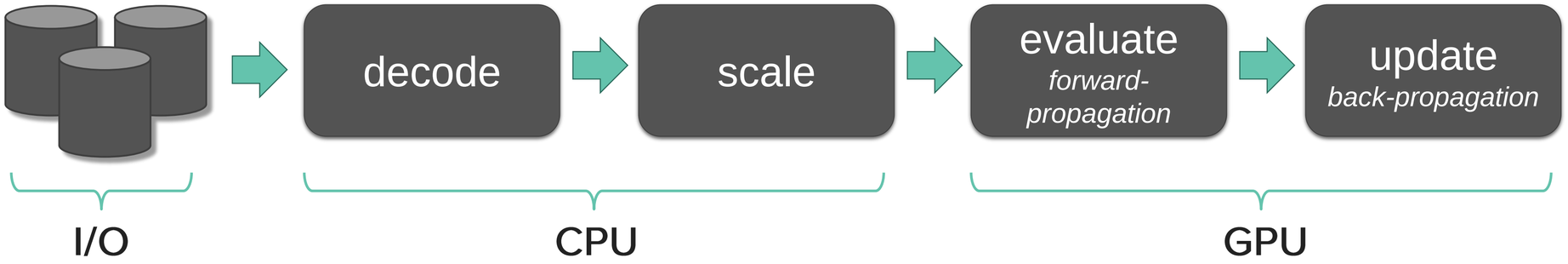
Make sure your DGX-1 is set up in Base OS mode, that you have an NFS server with one or more exports with data to be accessed by the DGX-1, and that there is network access between the DGX-1 and the NFS server.

출처) https://images.nvidia.com/content/technologies/deep-learning/pdf/DGX-1-UserGuide.pdf 25페이지

DEEP LEARNING TRAINING WORKFLOW

PERFORMANCE BENCHMARK SETUP

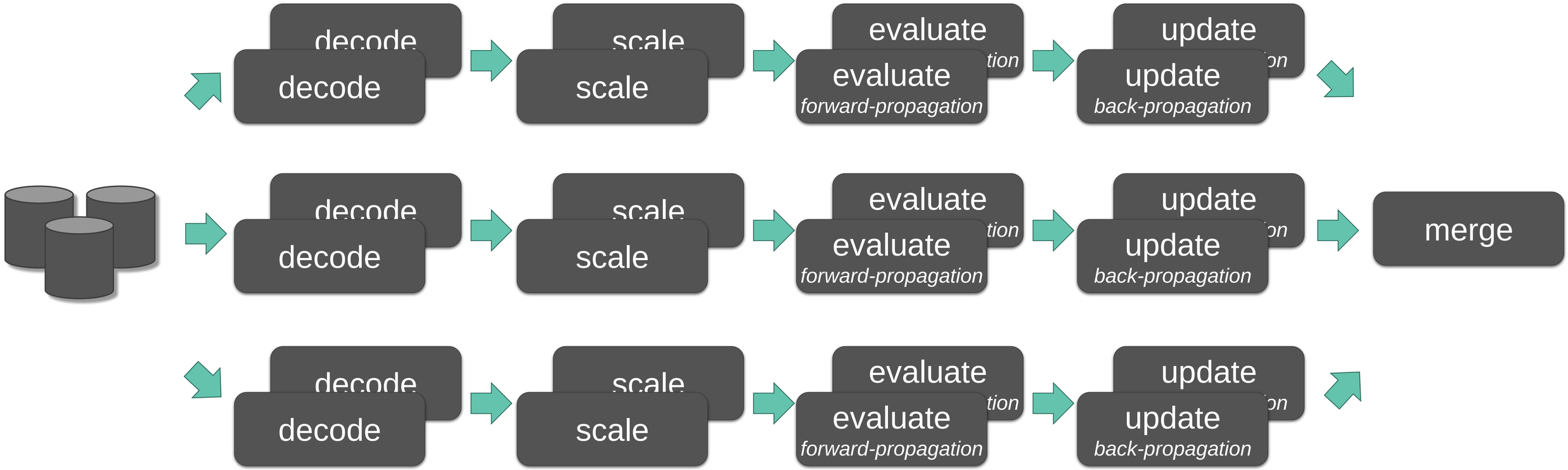
FULL TRAINING WORKFLOW



BENCHMARK SETUP	I/O	CPU	GPU
#1: Synthetic	System RAM	none	8x Tesla P100 GPUs in DGX-1
#2: Local SSD	4x SSDs in DGX-1	2x Intel Xeon CPUs	
#3: FlashBlade	FlashBlade 2x 10GbE in DGX-1	2x Intel Xeon CPUs	

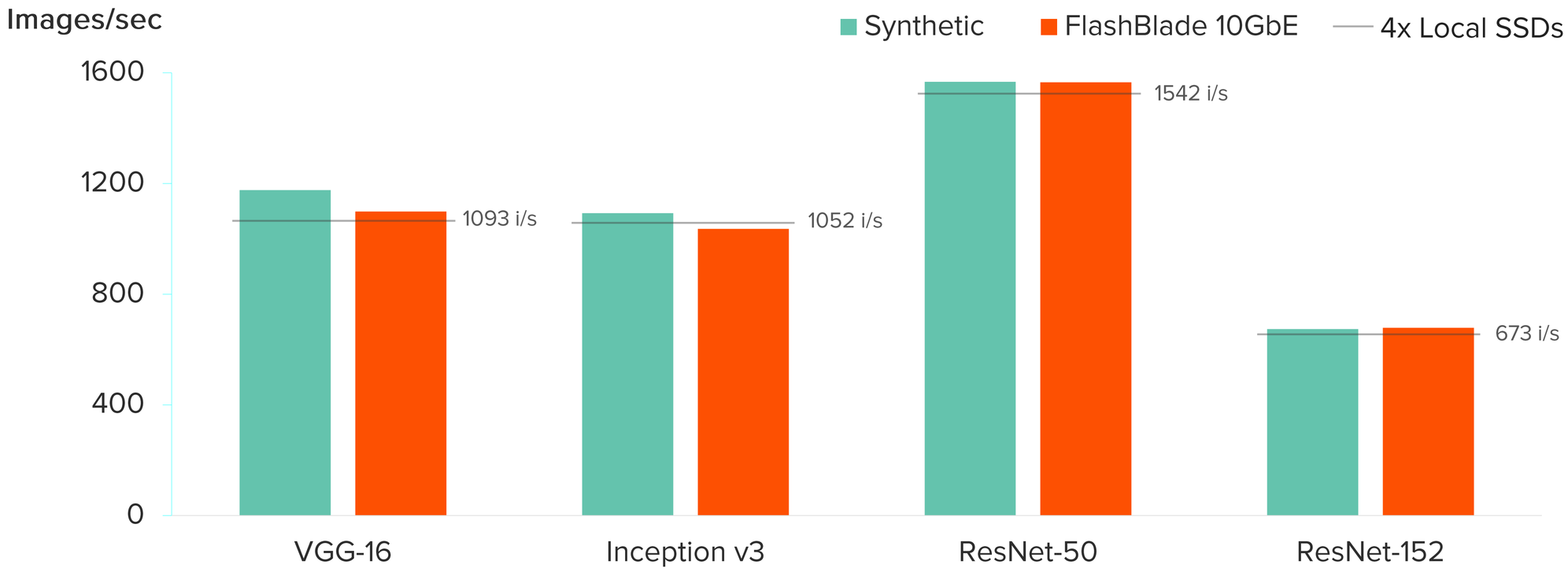
DEEP LEARNING TRAINING IN THE REAL-WORLD

NEED POWERFUL STORAGE TO FEED DISTRIBUTED TRAINING IN LARGE CLUSTER



FLASHBLADE의 LOCAL DATA 성능 제공

8-GPU TRAINING BENCHMARK FOR IMAGENET

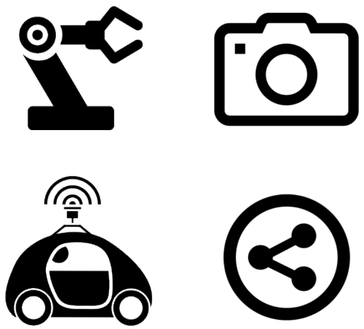


PURESTORAGE®

PRODUCTION에서 DEEP LEARNING의 복잡성

INGEST

From sensors, machines, & user generated



CLEAN & TRANSFORM

Label, anomaly detection, ETL, prep, stage

CPU Servers

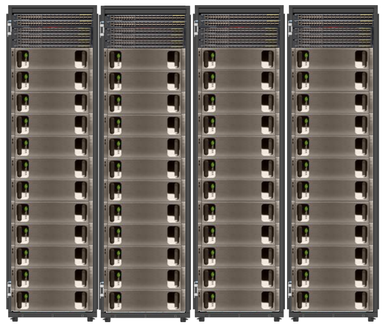
EXPLORE

Quickly iterate to converge on models

GPU Server

TRAIN

Run for hours to days in production cluster



GPU Production Cluster

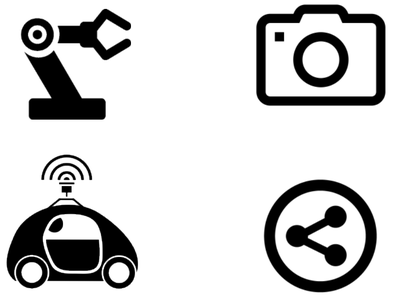
FLASHBLADE가 제공하는 DEEP LEARNING의 PIPELINE

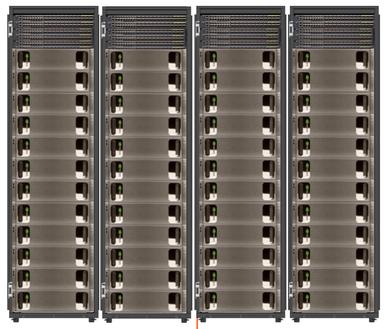
INGEST

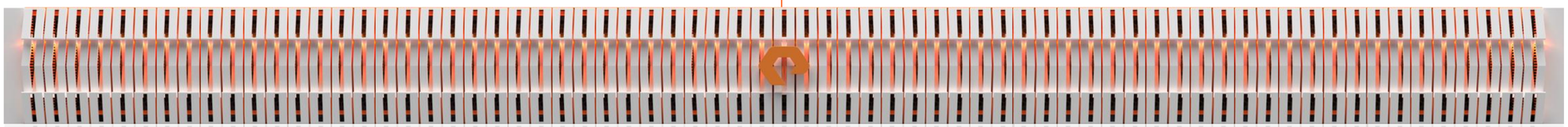
CLEAN & TRANSFORM

EXPLORE

TRAIN







PURESTORAGE®

NVIDIA DGX-1 DATA 전송 테스트 결과

	“cp -r”	Parallel Copy
1 TB	48.5 minutes	25.7 minutes
8 TB	6.5 hours	3.4 hours

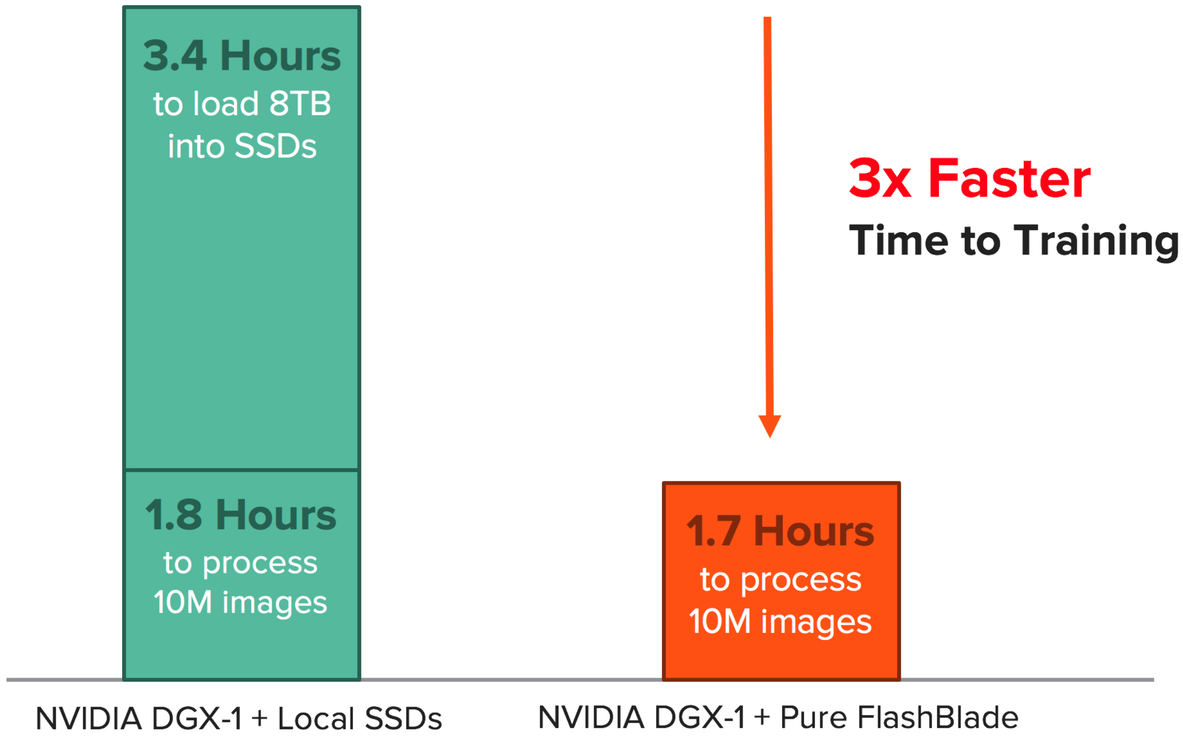
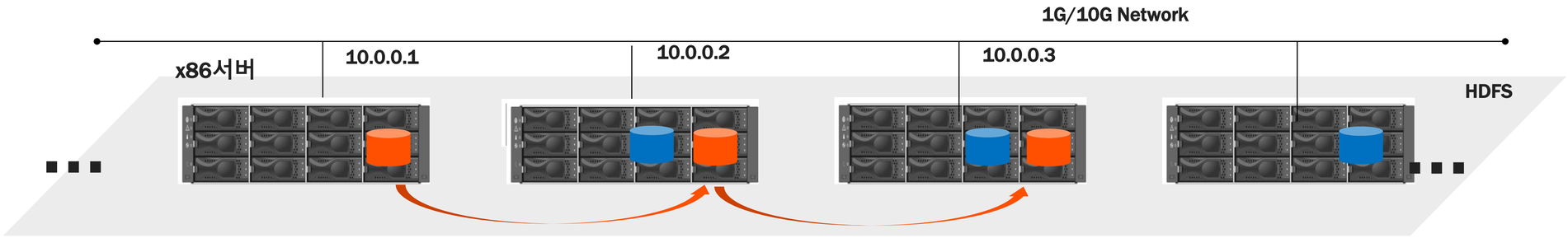


Figure: TensorFlow benchmark for ResNet-50, measuring training time for 10M images, including data transfer time for SSDs

빅데이터 에서의 FLASHBLADE

PURESTORAGE®

FLASHBLADE를 활용한 HDFS의 장점



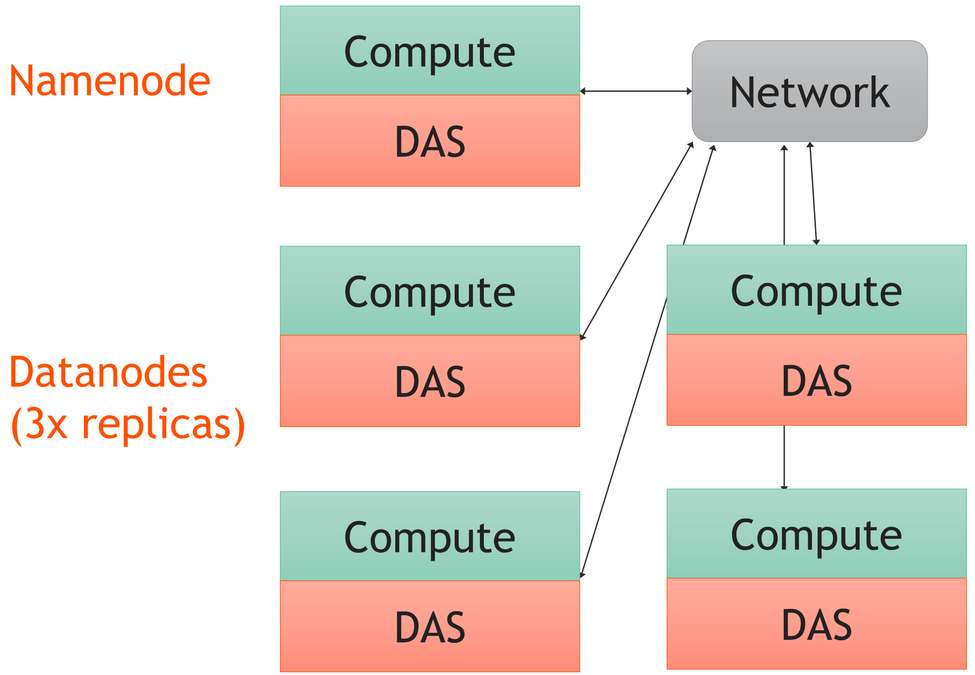
하둡/HDFS 특징	이슈 사항	FLASHBLADE 장점
다중 x86서버	Tightly coupled 구조	연산과 저장을 분리
복제를 통한 데이터보호	부가적인 IO 생성	저장 레이어에서 수행
병렬처리 수행	네트웍 데이터 전송 지연	데이터 이동 최소화
다양한 데이터 타입 저장	IO패턴에 따른 성능불균형	FLASH기반 안정적 성능 제공

PURESTORAGE®

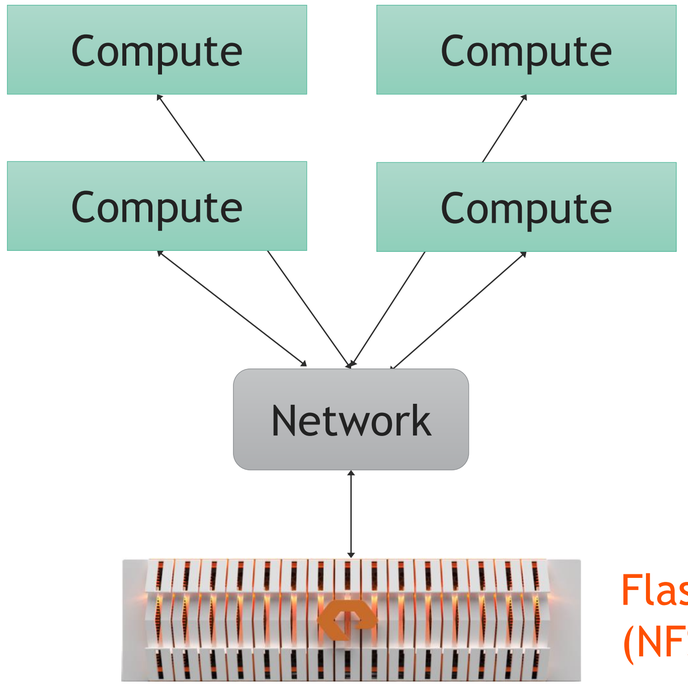
FLASHBLADE 기반 유연성 확보

업무 패턴에 따른 연산(CPU, Memory) 및 저장(HDFS) 분리 및 투자보호

Tightly Coupled 데이터 플랫폼



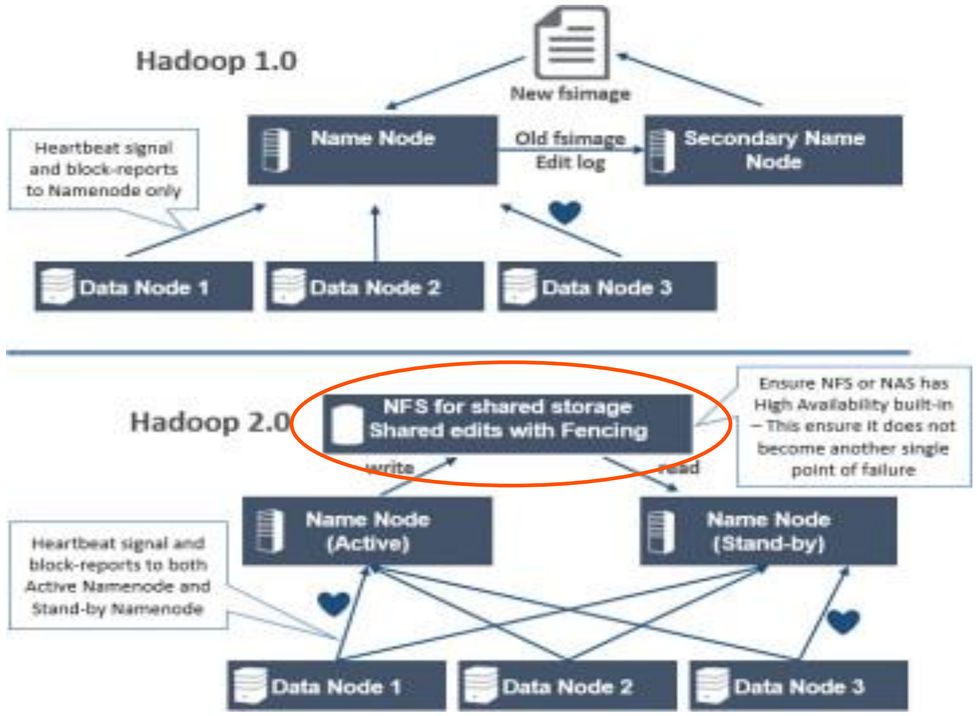
Loosely Couple 데이터 플랫폼



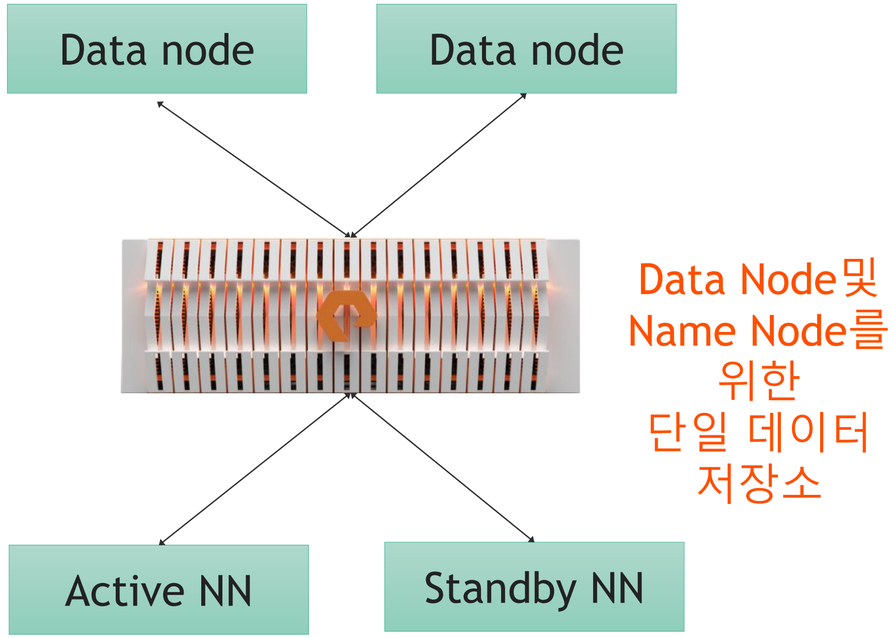
FLASHBLADE 기반 비용보호

NameNode 이중화를 위한 불필요한 비용 발생 방지

Name Node이중화를 위한 별도 NAS구성



단일 데이터 스토리지 구성

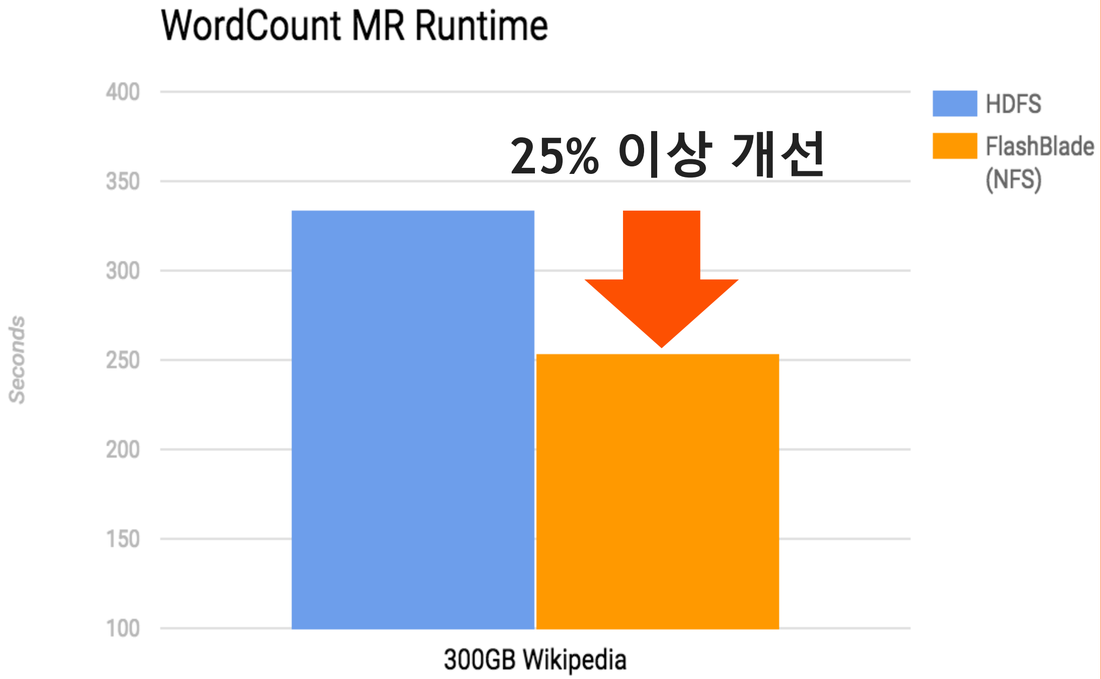


Data Node및 Name Node를 위한 단일 데이터 저장소

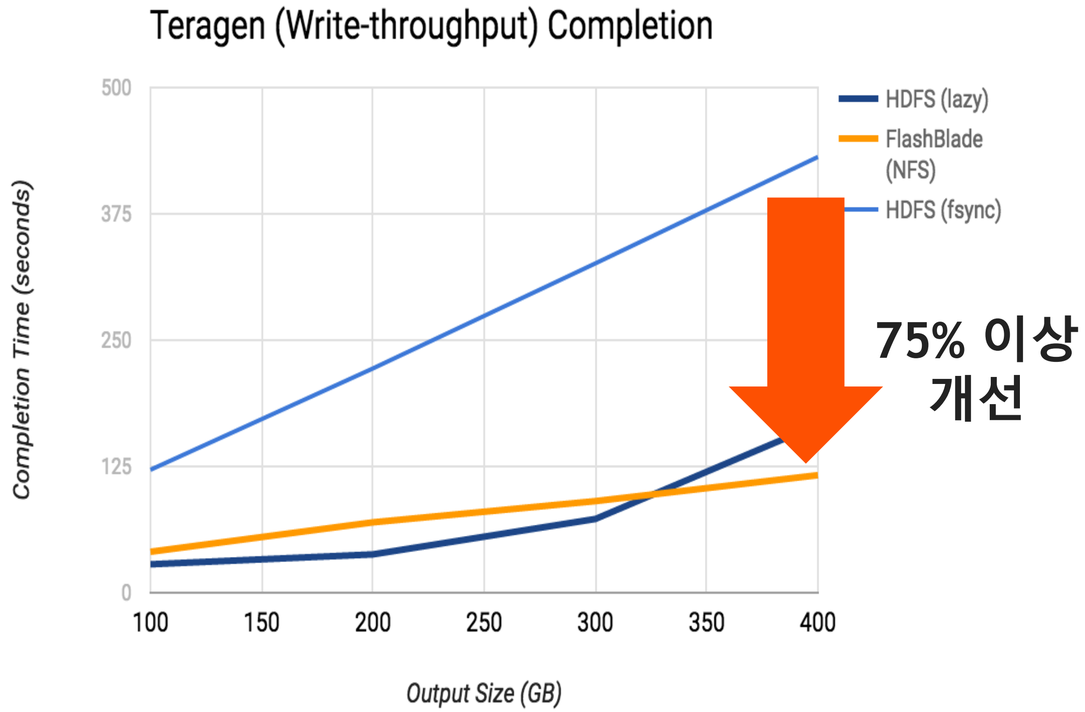
FLASHBLADE를 활용한 HDFS성능

NFS기반 HDFS구성 시 DAS대비하여 보다 빠른 Read/Write성능 제공

Read 성능비교



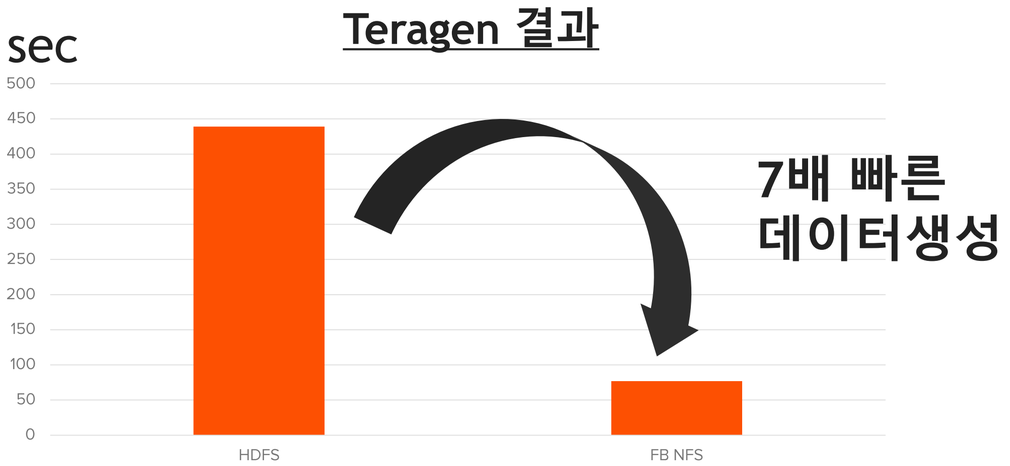
Write 성능비교



* https://blog.purestorage.com/how-to-run-hadoop-on-flashblade/

FLASHBLADE를 활용한 HDFS성능 - Teragen

FB NFS기반 HDFS구성 시 DAS대비하여 보다 빠른 Read/Write성능 제공

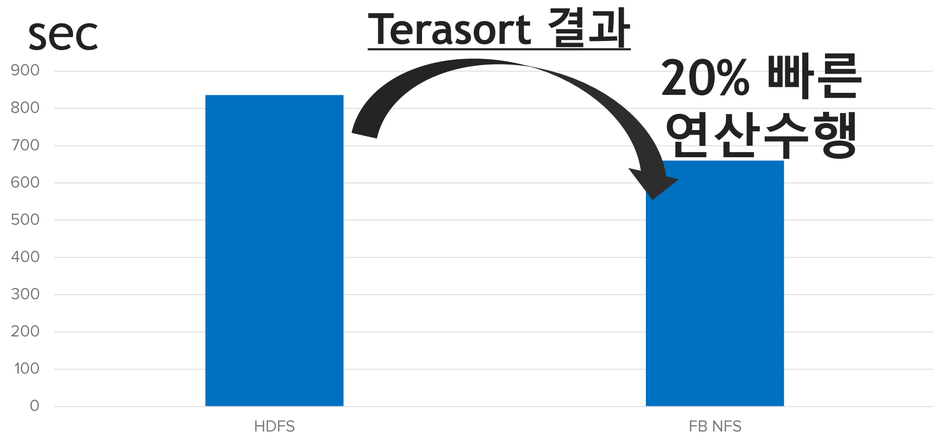


HDFS

User:	rpingarron
Name:	TeraGen
Application Type:	MAPREDUCE
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	FINISHED
Queue:	default
FinalStatus Reported by AM:	SUCCEEDED
Started:	Thu Nov 30 10:19:38 +0000 2017
Elapsed:	7mins, 19sec
Tracking URL:	History
Log Aggregation Status	SUCCEEDED
Diagnostics:	
Unmanaged Application:	false
Application Node Label expression:	<Not set>
AM container Node Label expression:	<DEFAULT_PARTITION>

FB NFS

User:	rpingarron
Name:	TeraGen
Application Type:	MAPREDUCE
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	FINISHED
Queue:	default
FinalStatus Reported by AM:	SUCCEEDED
Started:	Thu Nov 30 20:05:58 +0000 2017
Elapsed:	1mins, 17sec
Tracking URL:	History
Log Aggregation Status	SUCCEEDED
Diagnostics:	
Unmanaged Application:	false
Application Node Label expression:	<Not set>
AM container Node Label expression:	<DEFAULT_PARTITION>



HDFS

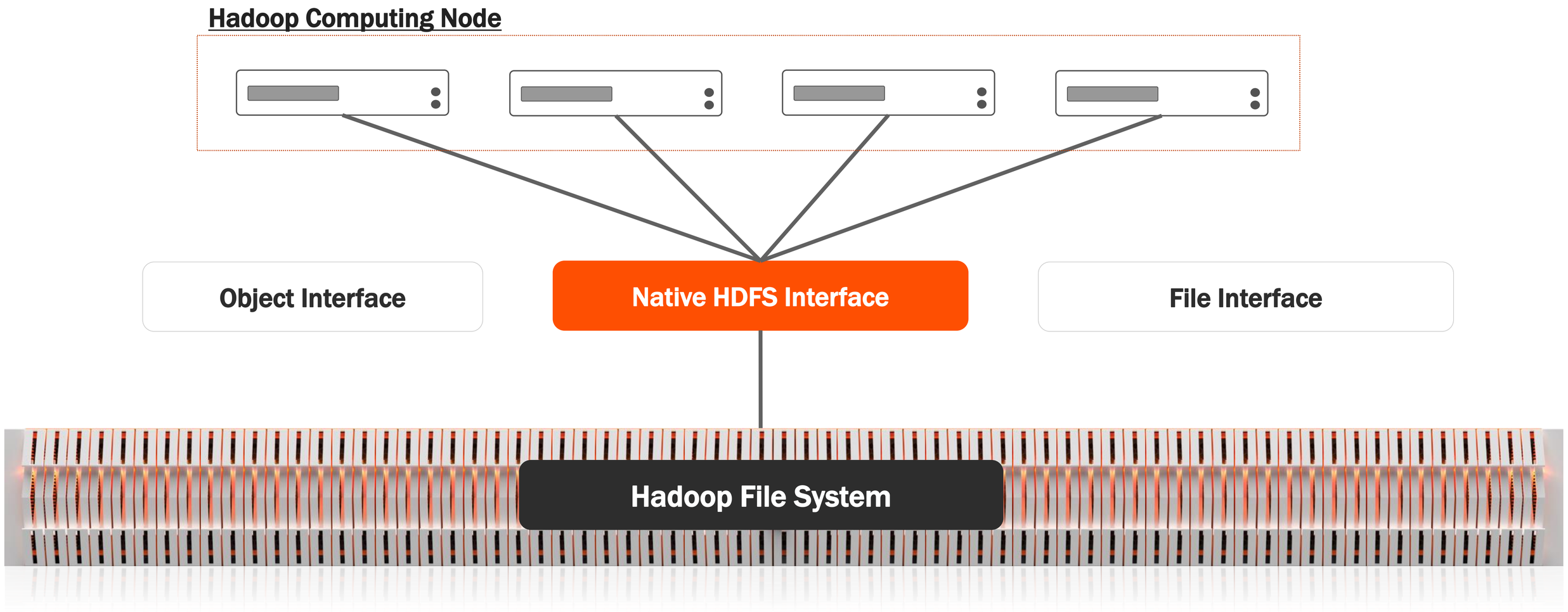
User:	rpingarron
Name:	TeraSort
Application Type:	MAPREDUCE
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	FINISHED
Queue:	default
FinalStatus Reported by AM:	SUCCEEDED
Started:	Thu Nov 30 10:41:21 +0000 2017
Elapsed:	13mins, 56sec
Tracking URL:	History
Log Aggregation Status	SUCCEEDED
Diagnostics:	
Unmanaged Application:	false
Application Node Label expression:	<Not set>
AM container Node Label expression:	<DEFAULT_PARTITION>

FB NFS

User:	yarn
Name:	TeraSort
Application Type:	MAPREDUCE
Application Tags:	
Application Priority:	0 (Higher Integer value indicates higher priority)
YarnApplicationState:	FINISHED
Queue:	default
FinalStatus Reported by AM:	SUCCEEDED
Started:	Thu Nov 30 21:13:06 +0000 2017
Elapsed:	11mins, 0sec
Tracking URL:	History
Log Aggregation Status	SUCCEEDED
Diagnostics:	
Unmanaged Application:	false
Application Node Label expression:	<Not set>
AM container Node Label expression:	<DEFAULT_PARTITION>

PURESTORAGE®

FLASHBLADE Native HDFS ROADMAP

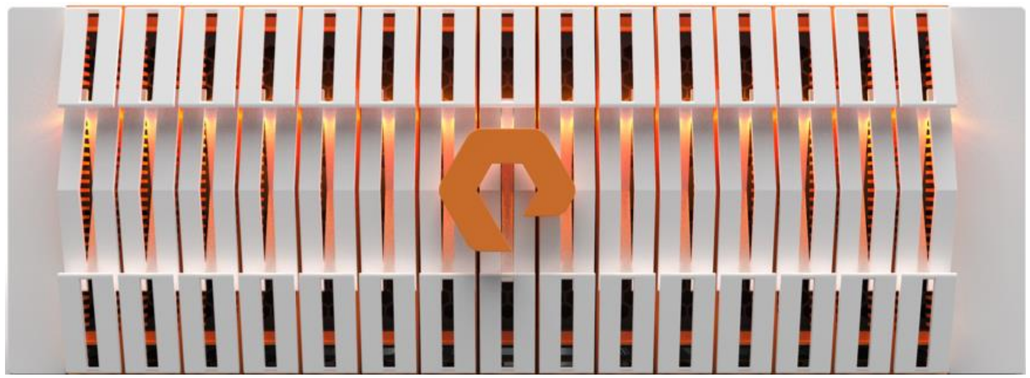


* FB Native HDFS지원은 2018년 상반기에 예정되어 있으며 출시 일정은 변경될 수 있습니다.

FLASHBLADE 아키텍처

PURESTORAGE®

INTRODUCING
FLASHBLADE



BLADE

SCALE-OUT
PROCESSING + FLASH

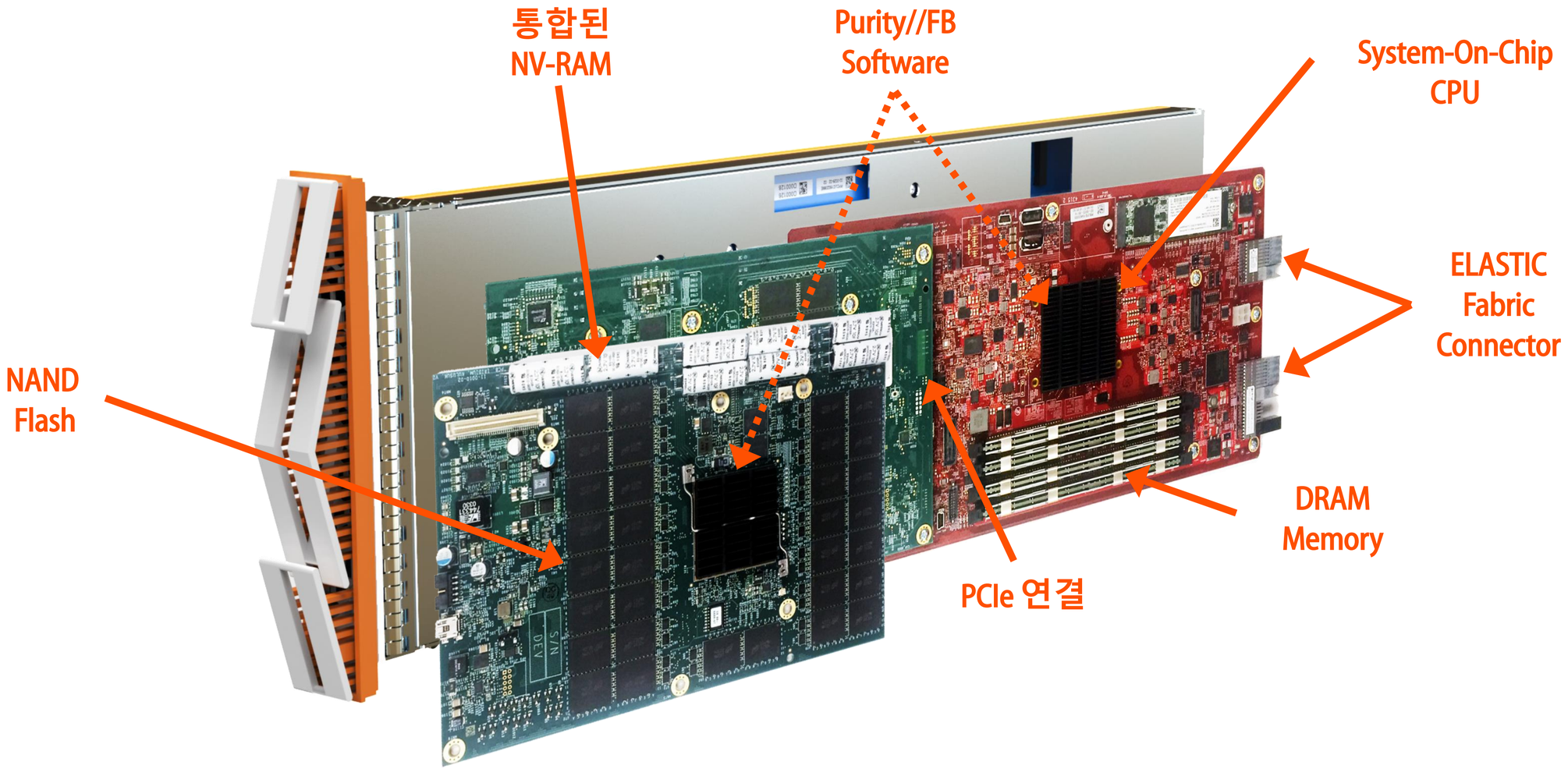
ELASTIC Fabric Module

LOW-LATENCY,
SW-DEFINED ETHERNET INTERCONNECT

Purity//FB2

SCALE-OUT STORAGE SOFTWARE

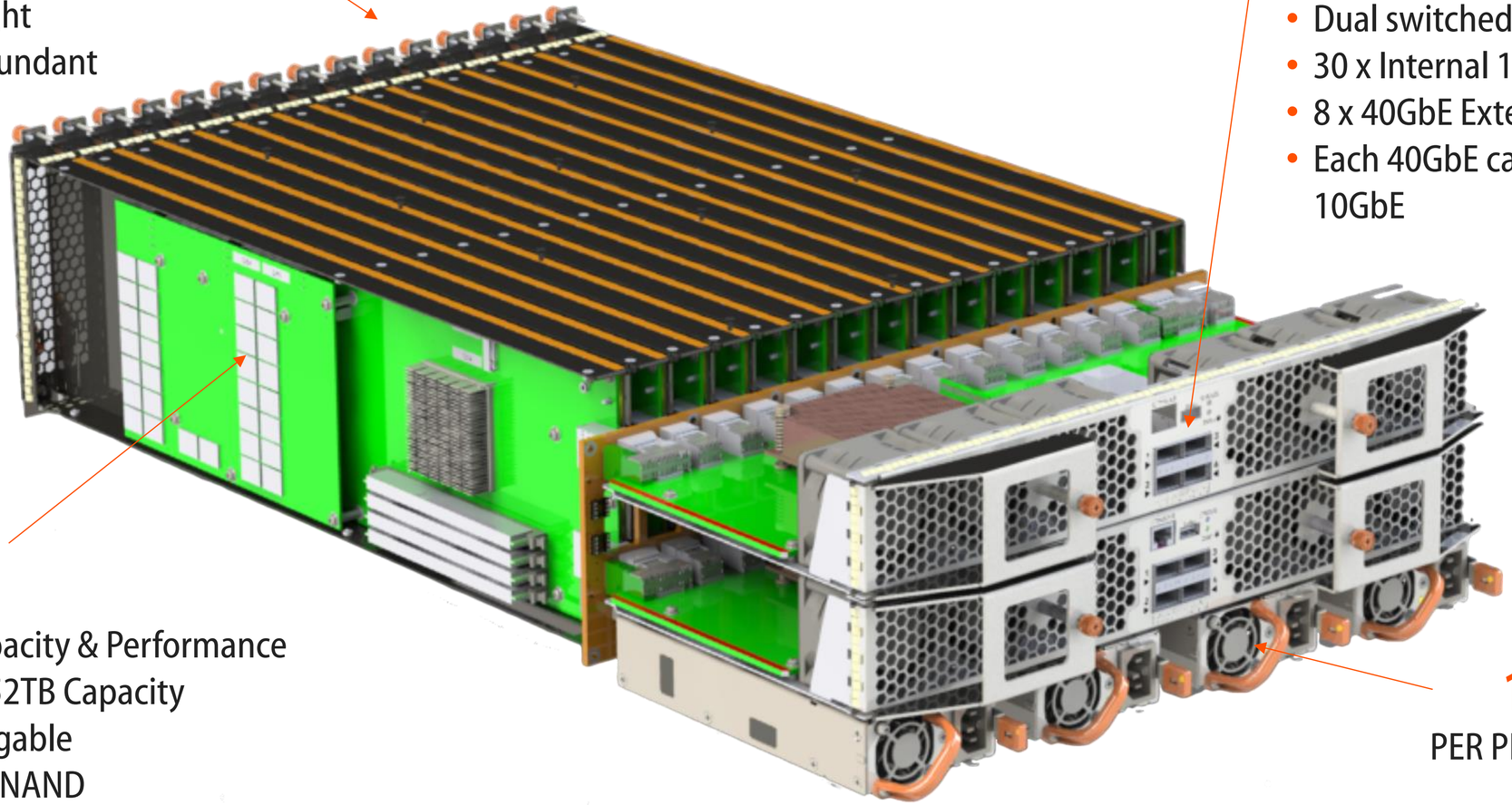
FlashBlade HW > Blade



FlashBlade HW > Chassis

FLASHBLADE CHASSIS

- Up to 15 Blades
- 4RU Height
- N+2 Redundant



FABRIC MODULE

- Dual switched mid-plane
- 30 x Internal 10Gbit ports
- 8 x 40GbE External ports
- Each 40GbE can be broken into 4x 10GbE

BLADE

- Adds Capacity & Performance
- 17TB or 52TB Capacity
- Hot Pluggable
- Non SSD NAND
- Embedded NVRAM

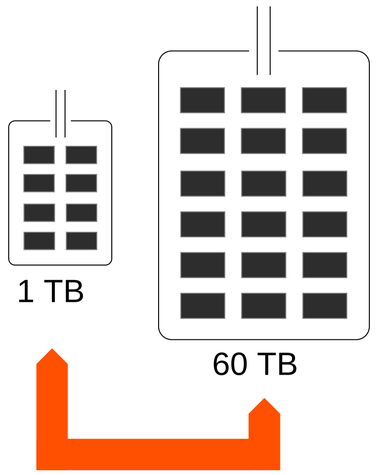
1,850 WATT

PER PETABYTE USABLE

FlashBlade HW > NVMe > SSD의 문제점

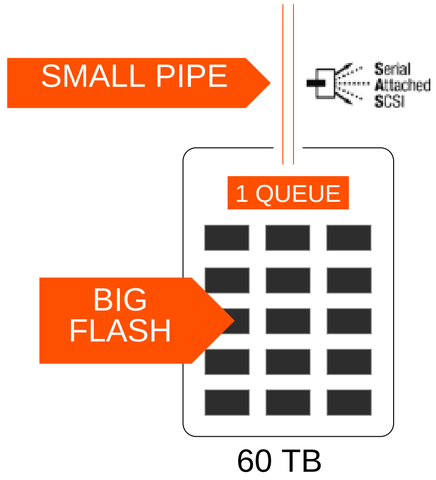
SSD의 용량이 점점 더 커짐에 따라 여러 가지 문제점이 대두되고 있습니다.

점점 커지는 SSD



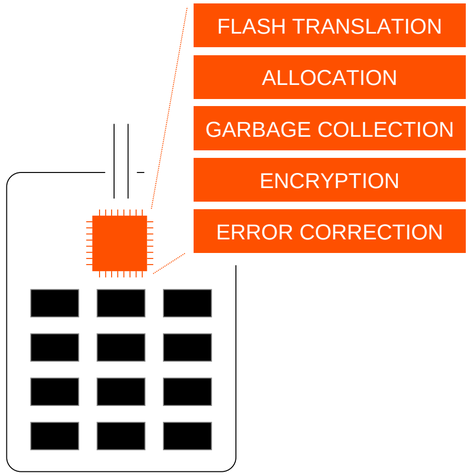
60배 더 많은 FLASH
여전히 같은 성능

작고 단일한 경로



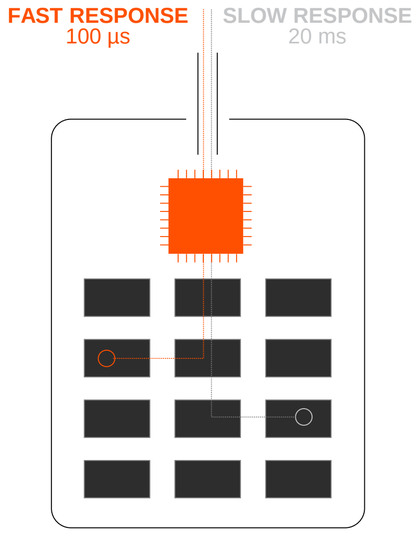
1 SERIAL CHANNEL

SOFTWARE 복잡성



수 백 개의 SSD에서
독자적으로 수행되는 복잡한 작업

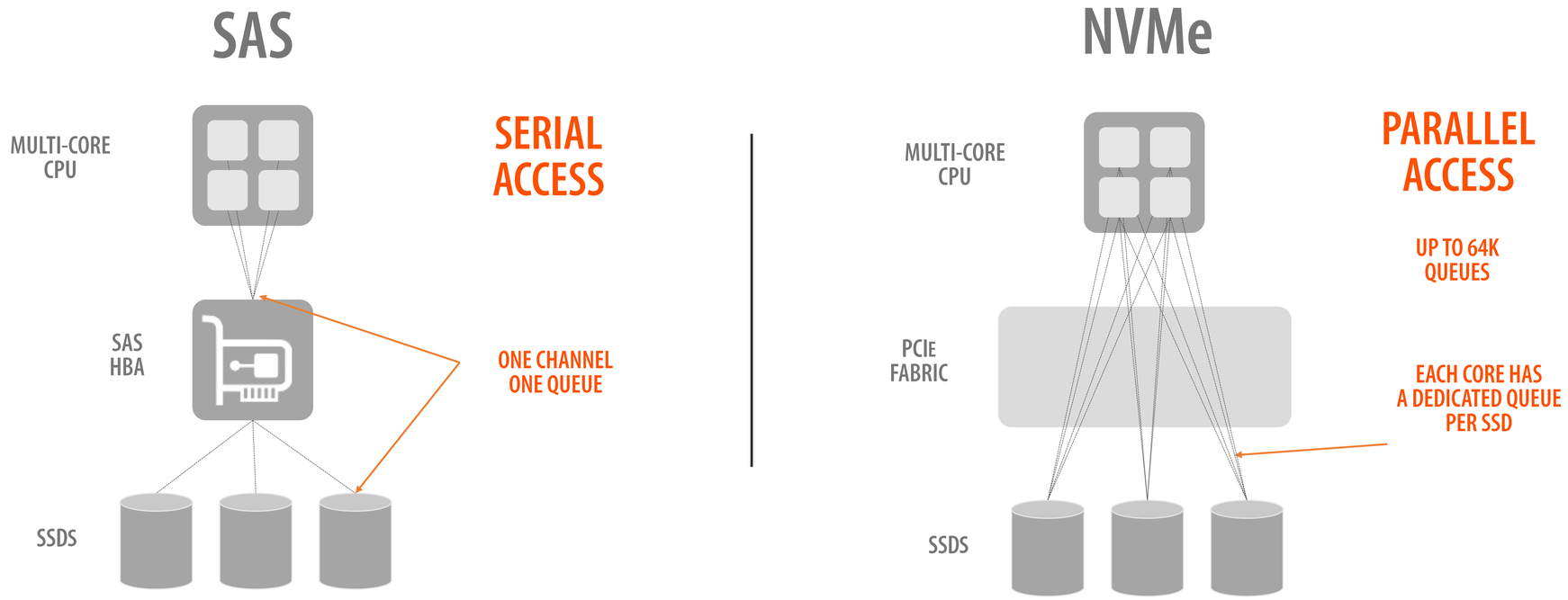
예측 불가한 성능



WHAT'S GOING ON IN THERE?
GARBAGE COLLECTION,
REMAPPING…

FlashBlade HW > NVMe > 필요성

SAS방식의 단일 채널로 인한 병목을 해소하기 위해, PCIe 방식의 NVMe가 필요합니다.

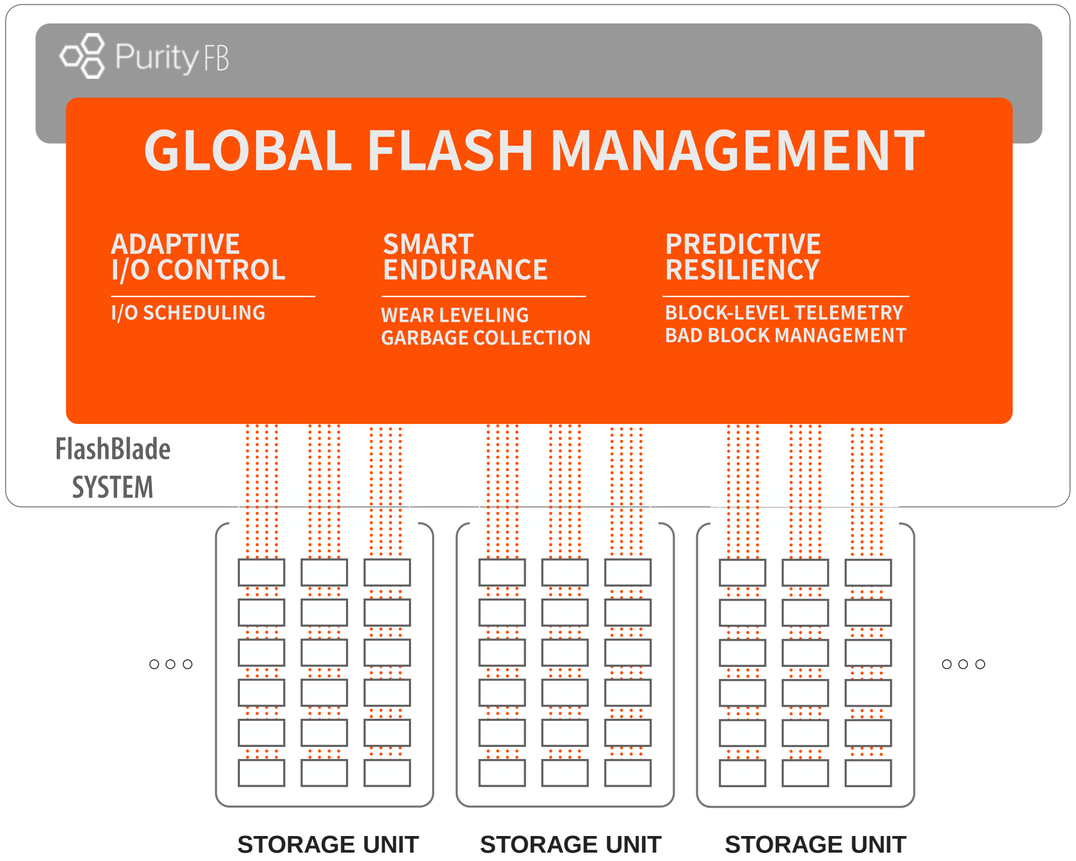


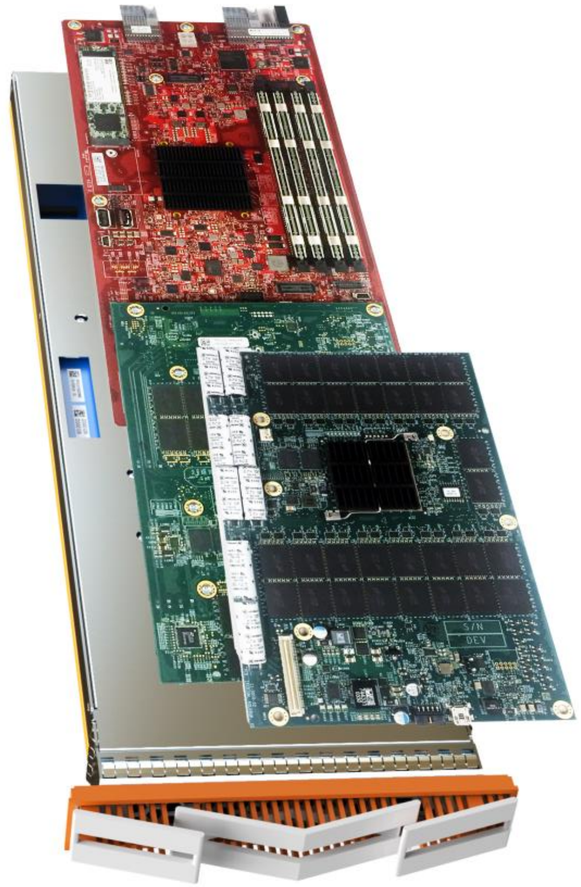
SAS**는 느린** HDD **환경에서 사용하기에는 충분한 아키텍처**

Memory**기반의 아키텍처는** Multiple Device**에 병렬로** Access**하도록 허용**

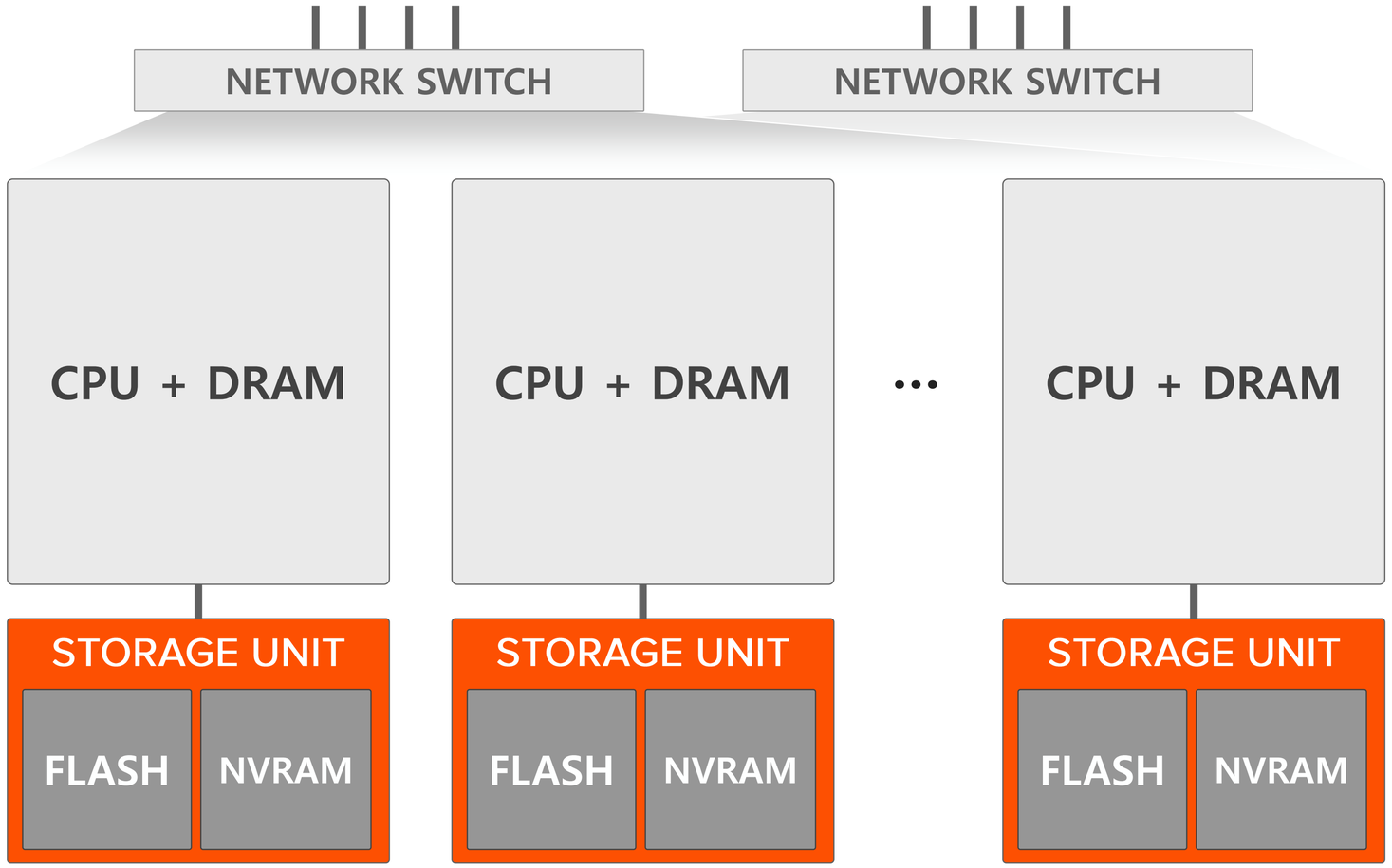
FlashBlade HW > NVMe > Purity//FB의 Global Flash Management

SSD의 개별 Controller에 의존하지 않고, FlashBlade의 Operation Environment인 Purity//FB에 의해 모든 NAND Flash를 전체적으로 관리합니다.





FlashBlade 개념도



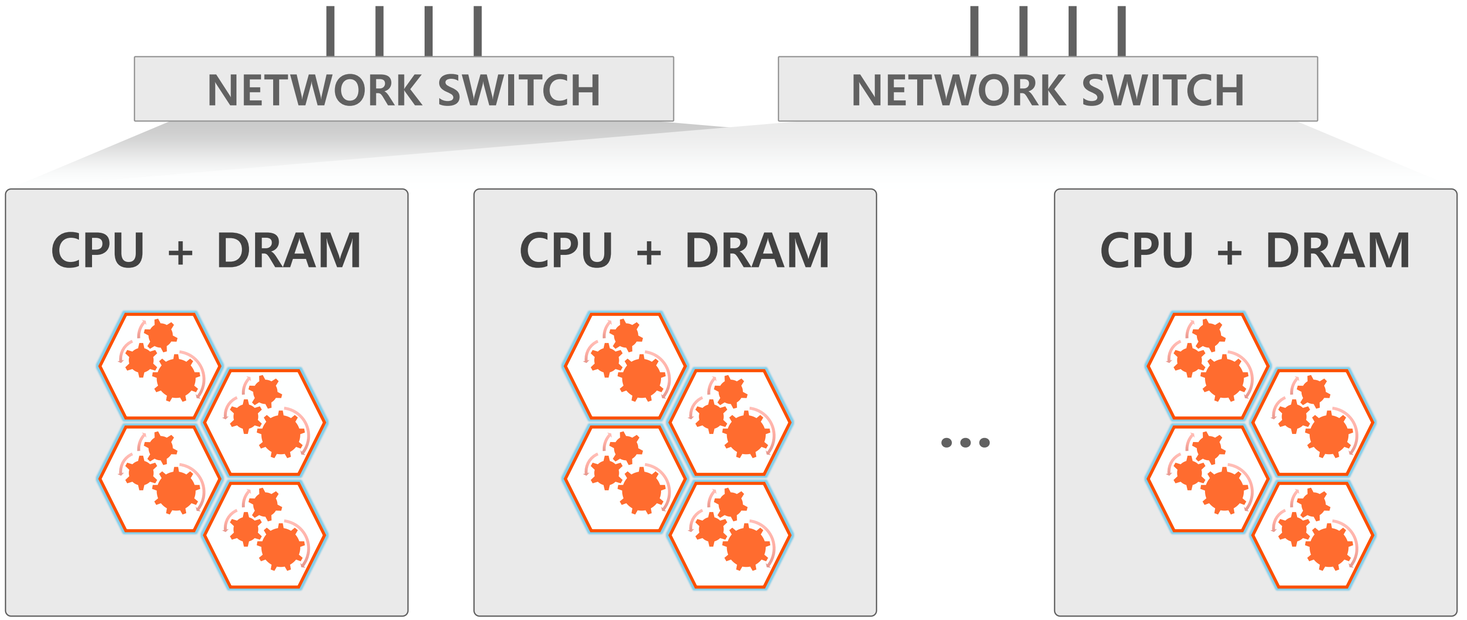
High-performance, integrated fabric

Scalable server node

PCIe-attached flash and NVRAM in a proprietary storage module

FlashBlade SW > VIRTUAL CONTROLLER Concept 1/3

Each one is similar to a mini controller



Split into independent virtual controller

Many virtual controller on each blade

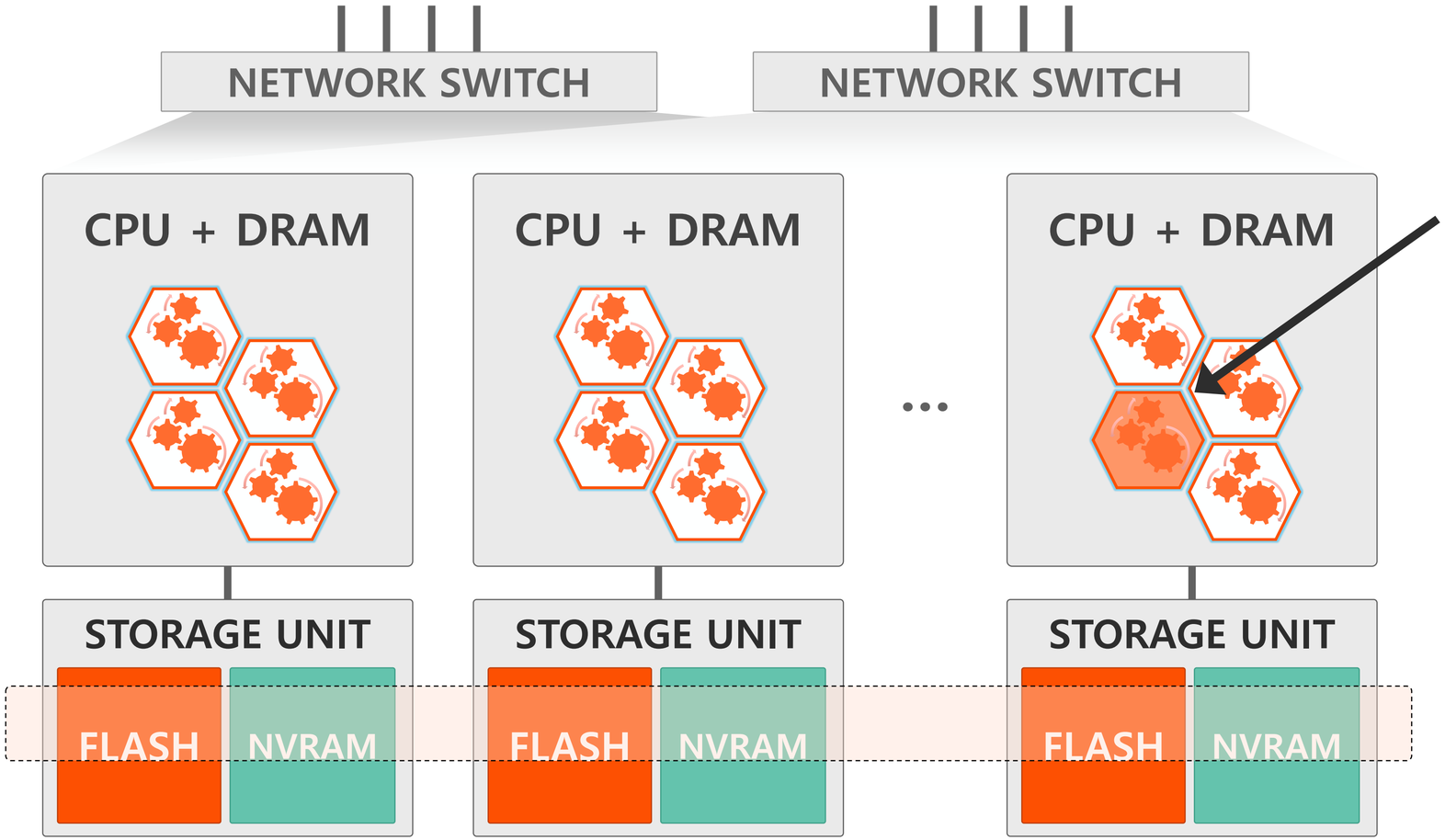
virtual controller migrate to any blade

Each Partition manages its own:

- Namespace Operations
- Data Layouts
- Data Protection
- Maintenance (GC, FTL, etc)

FlashBlade SW > VIRTUAL CONTROLLER Concept 2/3

Each one is similar to a mini controller



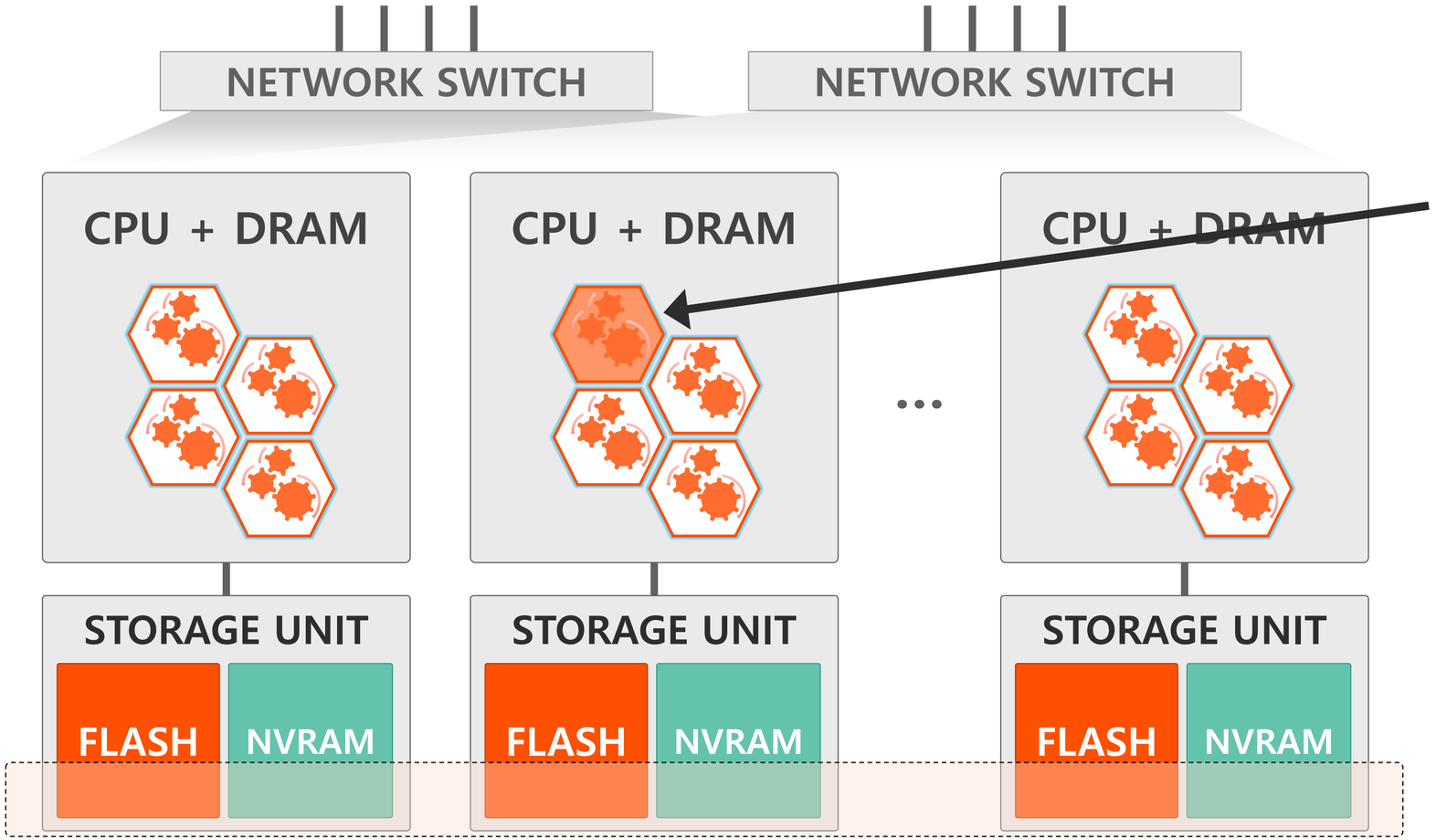
Virtual controller

Virtual controller sees array of storage across each blade

Storage Unit provides some resources for each virtual controller

FlashBlade SW > VIRTUAL CONTROLLER Concept 3/3

Each one is similar to a mini controller



Virtual controller

Virtual controller sees array of storage across each blade

Storage Unit provides some resources for each virtual controller

FlashBlade SW > VIRTUAL CONTROLLERS PARTITIONING SCHEME

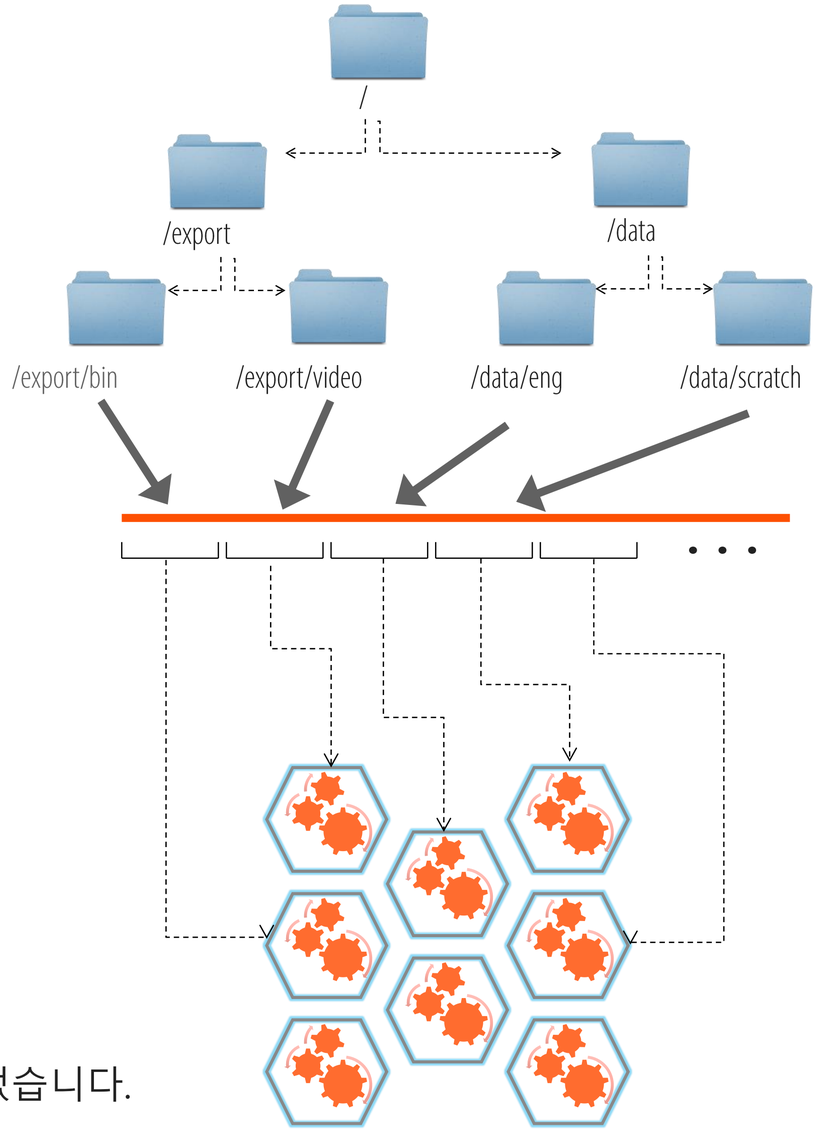
PARTITIONING A NAMESPACE TO BALANCE EVENLY

by hashing algorithm

OBJECT IDS Ranges of object ids (e.g. file/dir inode) map to a partition

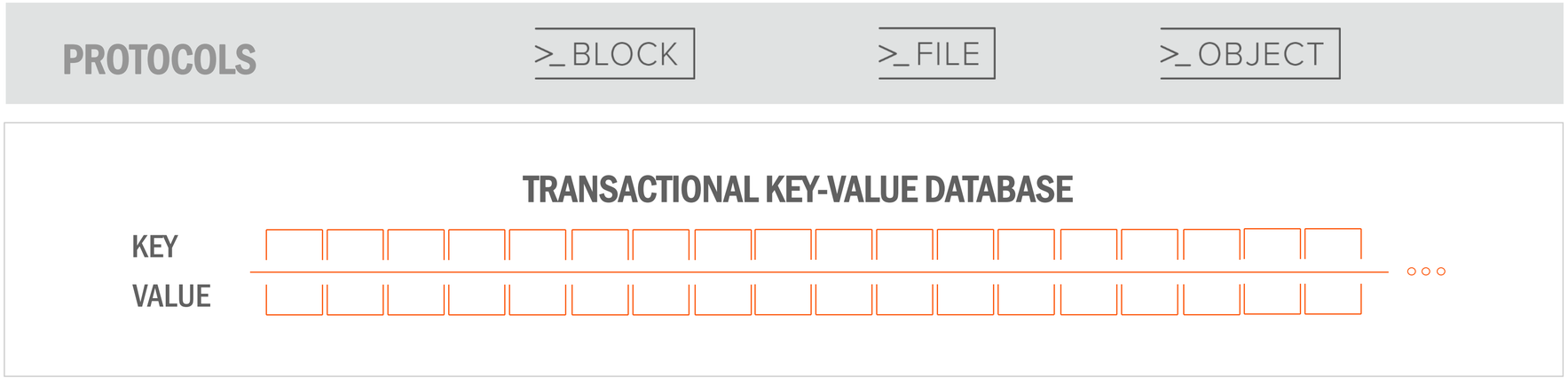
OBJECT DATA Object-id and offset within the object maps to a partition

OBJECT LINKS Containing object (e.g. dir) and name maps to a partition



✓ NFS를 구성하는 RPC Protocol 21개 중에서 약 70%의 부하를
차지하는 것이 "Procedure 1: GETATTR - Get file attributes" 입니다.
FlashBlade는 이 부분을 효율적으로 처리하여 NFS의 성능을 극대화 할 수 있도록 설계되었습니다.

FlashBlade SW > PURITY//FB



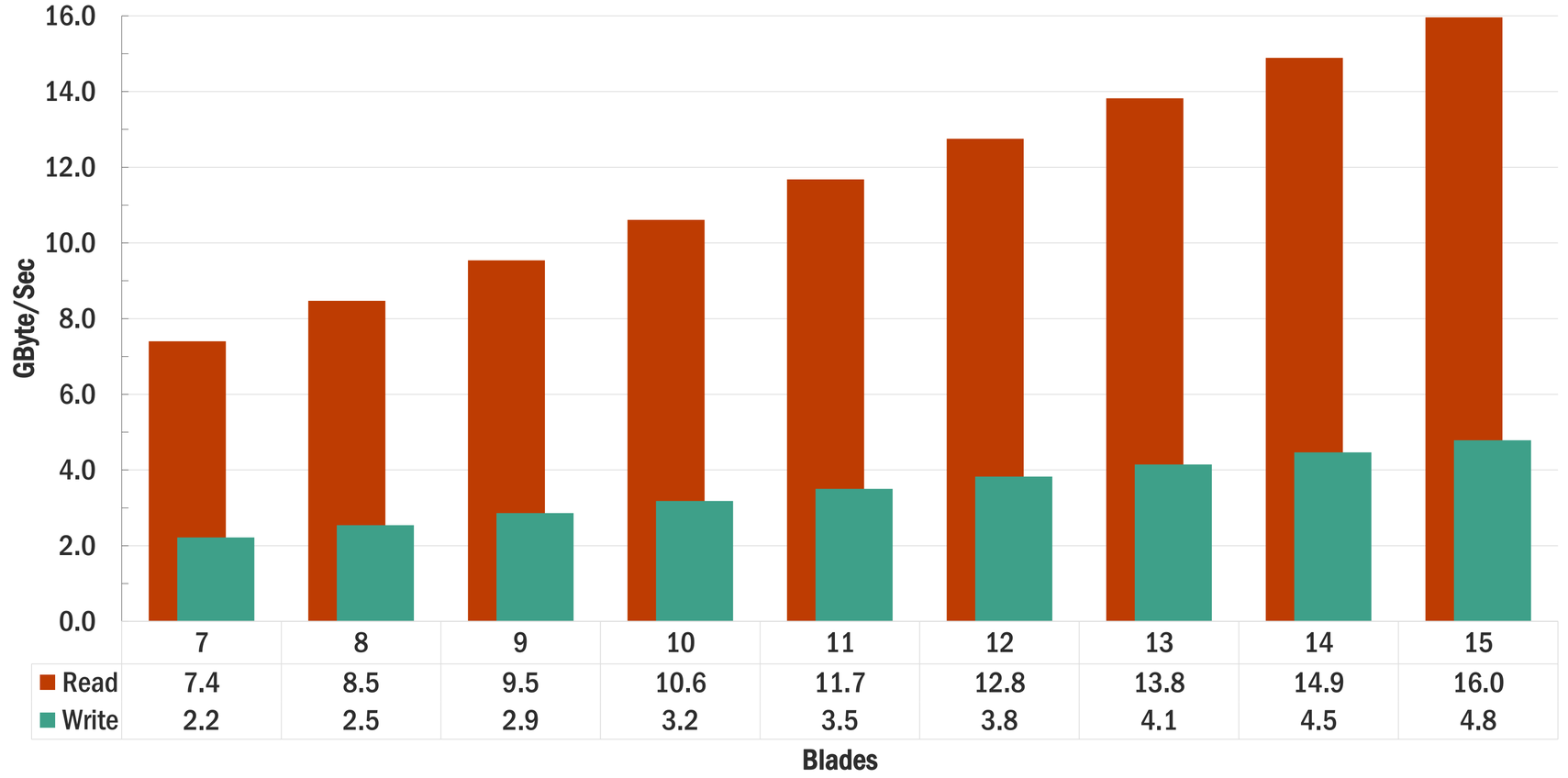
BLOCK → FILE → OBJECT →

NEXT-GENERATION OF APPS: NEW STORAGE INTERFACES

FlashBlade Performance

512K IO sizes, 16 load generators

(48 core CPU's each with 2x10GbE), 256 Containers total, NFSv3



	7	8	9	10	11	12	13	14	15
Read	7.4	8.5	9.5	10.6	11.7	12.8	13.8	14.9	16.0
Write	2.2	2.5	2.9	3.2	3.5	3.8	4.1	4.5	4.8

FlashBlade 요약

프로토콜:

- NFS v3
- SMB v2.1
- Object
- HTTPS

성능:

- 1,000,000 IOPS 이상
- 16 GB/s 이상

용량:

- 17TB, 52TB Blade 옵션
- 1개 섀시부터 최대 5개 섀시 증설 가능, 65.7TB 부터 2.6PB 주1)

데이터 보호:

- N+2, Erasure Coding
- Always-on 암호
- 내부 복제 지원

데이터 절감:

- In-line compression → 평균 압축률 3:1 (67% 데이터 절감)

주1) 압축률 고려하지 않은 실용량

도입 사례

PURESTORAGE®

운영 비용 절감 사례

- 영국 런던의 조사 기업인 IHS Markit은 항공 우주, 방위, 보안 그리고 자동차, 화학 물질, 에너지, 기술, 해상 및 무역 등의 산업 분야에 대해서 기업과 정부기관의 의사 결정 프로세스를 지원하기 위한 정보와 분석 결과를 제공하는 기업입니다.
- 기존에 사용하던 20개의 RACK을 퓨어스토리지 FlashBlade의 단 4U로 마이그레이션 하여,
  고성능의 실시간 대용량 데이터 분석 서비스를 제공하고 있습니다.

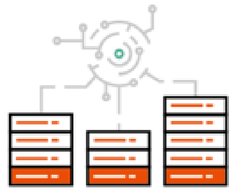
IHS Markit™

20 RACKS DISK → **4U**

PURESTORAGE®

도입 사례 요약

1 **POWERS, AI 사례**
세계에서 가장 우수한
인공지능 슈퍼컴퓨터의 구동

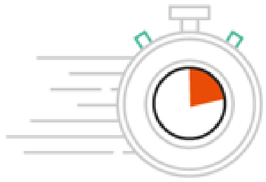


2 **HELPS, 통신사례**
대형 통신사, FlashBlade를 활용한
실시간 액티비티 분석으로 해커 제압

3 **SHORTENS, 반도체사례**
세계에서 가장 강력한
대형 반도체 칩의 설계 주기 단축

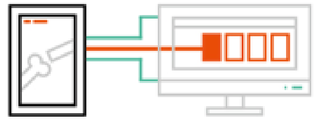
4 **ACCELERATES, 우주공학사례**
기상 및 로켓 시뮬레이션 가속화로
성공적인 발사 지원

5 **ENABLES, 대학사례**
UC Berkley Spark 팀,
쿼리 속도 12시간에서 30분으로
단축하며 24배 개선



6 **POWERS, 자동차공학사례**
Mercedes F1팀,
차량 설계 최적화를 통해
전체 경주 시간 1초 단축

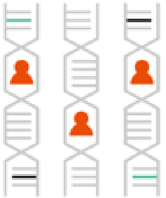
7 **DELIVERS, 의료사례**
미국 중서부 전역의 병원 네트워크를
통해 의료 영상 전송 속도 향상



8 **HELPS, 정유사례**
지구물리학자들, 멕시코 만에서 유전 발견

9 **CAPTURES, 사법기관 사례**
아이오와주 대번포트시,
법 집행 현장 바디 카메라 영상 저장

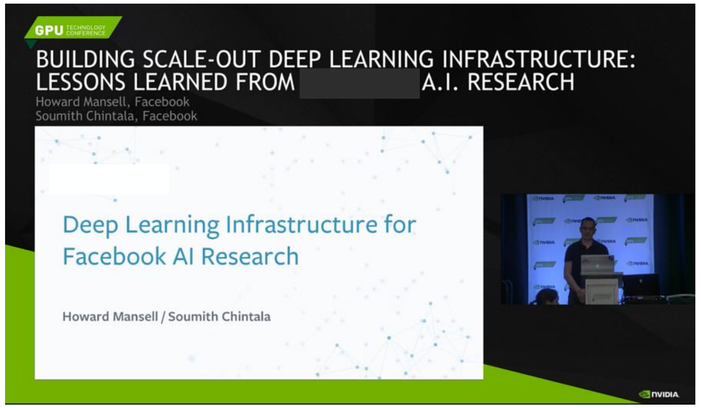
10 **ACCELERATES, 유전자 공학 사례**
보다 빠르고 개인화 된 의료 서비스 제공을 위한
유전체 분석 가속화



PURESTORAGE®

도입 사례 > 해외 대형 SNS사

- NVIDIA DGX-1 Node 128대, Linux 서버 70대
- 1분당 14만개 사진 업로드, 35만개의 자료 업데이트를 처리
- **요구 성능:**
  1. Read Size: 50KB
  2. Random Read Bandwidth: 70GB/sec 이상
  3. Random Write Bandwidth: 10GB/sec 이상
- **도입 사유:**
  1. 대량의 Write 발생시, Cache 아키텍처 기반의 스토리지 에서는 현저한 성능 저하 발생
     ➔ 퓨어스토리지는 Address Mapping 구조인 Key-Value 이므로 성능저하가 없음
  2. DGX-1 노드가 RACK 단위로 구성되어 확장이 용이한 아키텍처 제품 선정
     ➔ 퓨어스토리지는 블레이드에 각각의 Compute Module과 Storage Unit이 있어서 성능이 선형적으로 향상되며, 최대 75개의 블레이드로 확장 가능
     (향후에는 225개 까지 확장 가능)



[동영상 내용 요약]

➢ **Neural Network의 Access Pattern**

- I/O 환경: Random Access 하면서도 high throughput이 필요하여, 간단한 모델의 경우 초당 1,000개 이상의 이미지 트레이닝

- 도입사유: 클러스터로 구성되면, 트레이닝에 더 많은 Random Access가 발생함으로 **스토리지의 Cache 역할이 무색**하게 되어 급격한 성능 저하가 발생하였는데, 이러한 I/O 환경을 극복할 수 있는 아키텍처를 가진 스토리지를 채택

➢ **스토리지 구성 사항**

- Shared storage system: All flash
- Supports ~150,000 50KB files/sec per 100TB
- Shared datasets

*Source: s7815-howard-mansell-building-scale-out-deep-learning-infrastructure-lessons-lea -facebook-ai-research

도입 사례 > 국내 대형 SNS사

- GPU서버 클러스터 Node 60여대, Linux 서버 40여대
- 텍스트, 이미지, 동영상 등의 다양한 데이터 처리
- **요구 사항:**
  1. Scale Out이 용이해야 하며, Node 증가 시 성능 저하가 없어야 함
  2. 성능저하를 최소화 하는 HW, SW 아키텍처에 중점
- **도입 사유:**
  1. 성능저하 없는 Scale Out이 용이한 아키텍처
  2. 고 성능의 NVMe 기반 신기술 HW 아키텍처
  3. 기존의 계층형 디렉토리 구조의 NAS가 아닌, Object 기반의 새로운 SW 아키텍처
  4. 경쟁사 대비 탁월한 TCO(도입비용/운영비용)