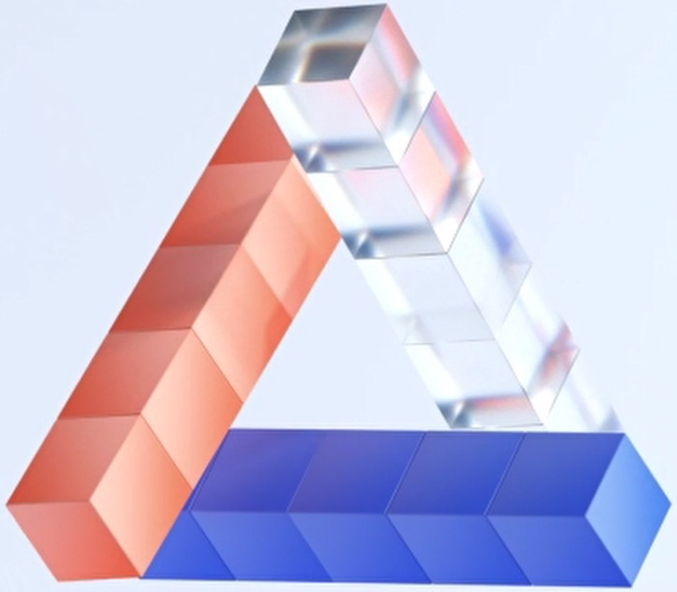
DEVOCEAN

In-Memory Data Grid 기반 Smart Factory 아키텍처링 연구 사례

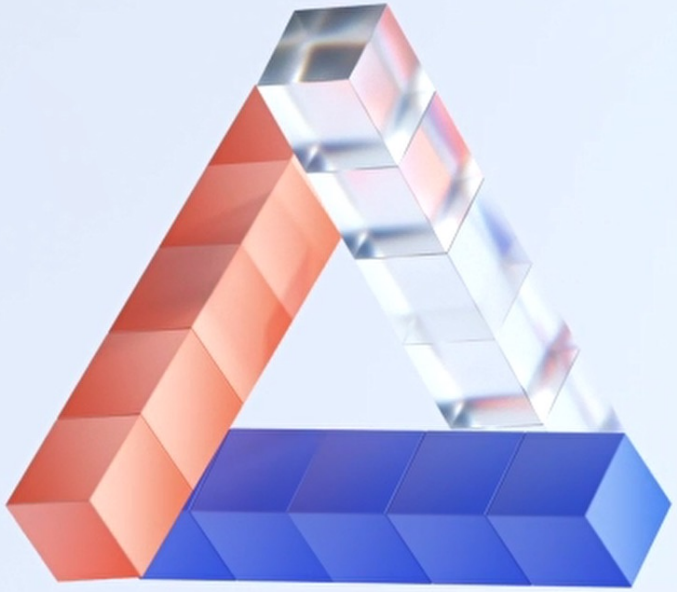
SK hynix
DT
Data Engineering/Solution팀
박재승

SK hynix
메모리시스템연구
Platform Software팀
오세진

DEVOCEAN



DEVOCEAN

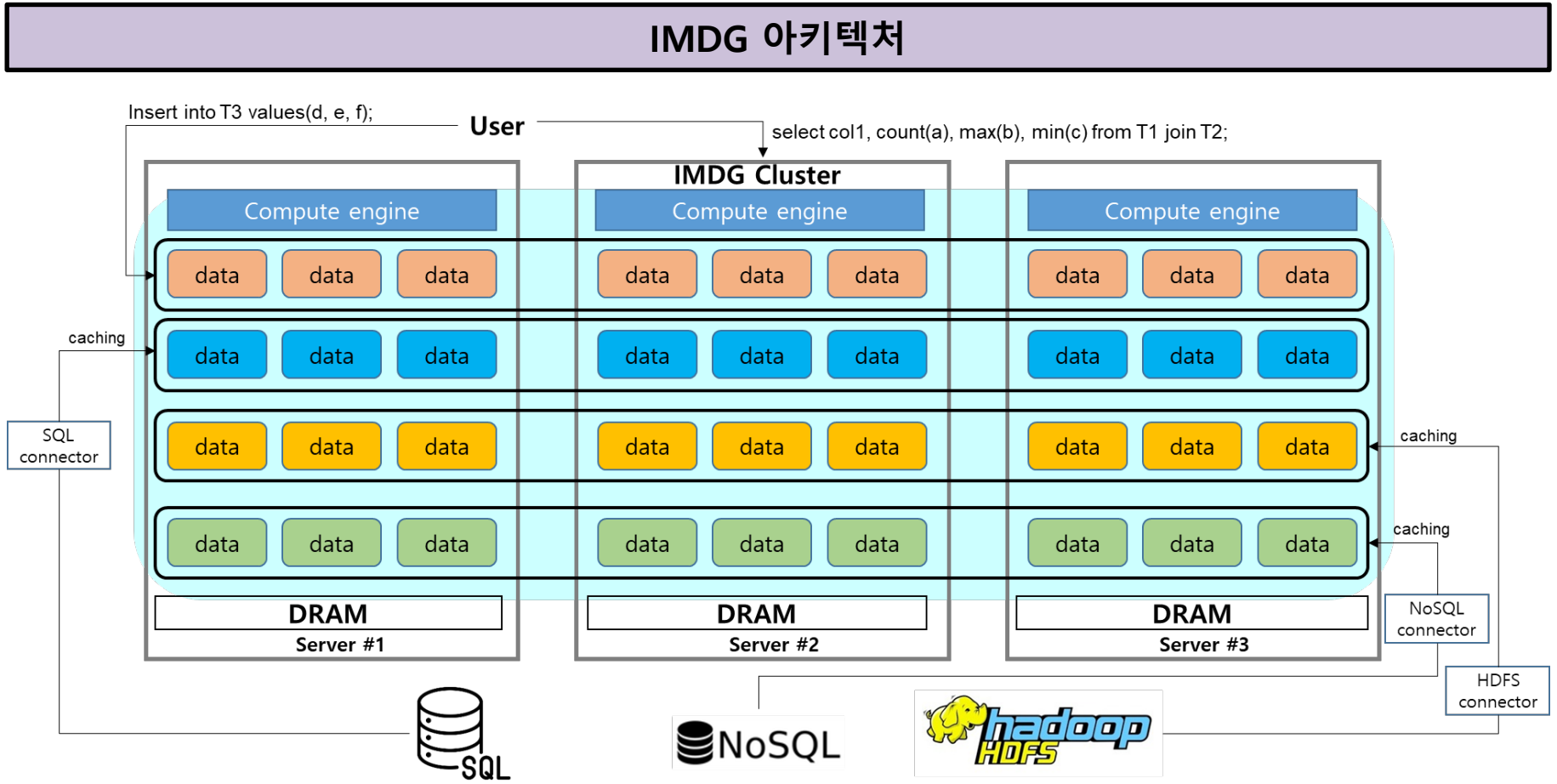


In-Memory Data Grid 소개

In-Memory Data Grid (IMDG)란

○ **데이터 통합 관리 위한 computing engine이 탑재된 In-memory distributed caching software 기술**

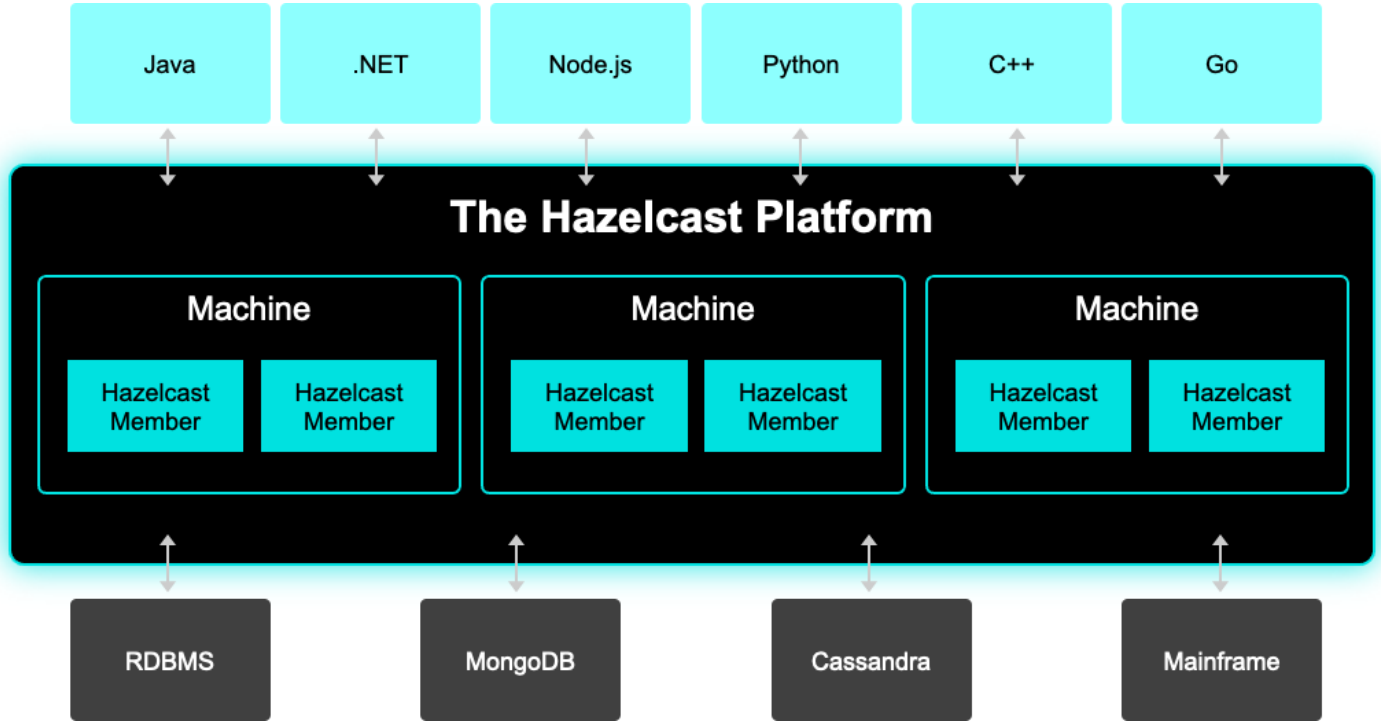
- 이종 데이터 source부터의 통합 View 제공
- Compute engine 통한 in-memory data base 기능 병행 제공
- 다수의 서버들이 서로 연결되어 클러스터 구축되면 대용량 memory pool이 형성되고, 데이터 로딩 위해 대용량 메모리 요구



Hazelcast IMDG

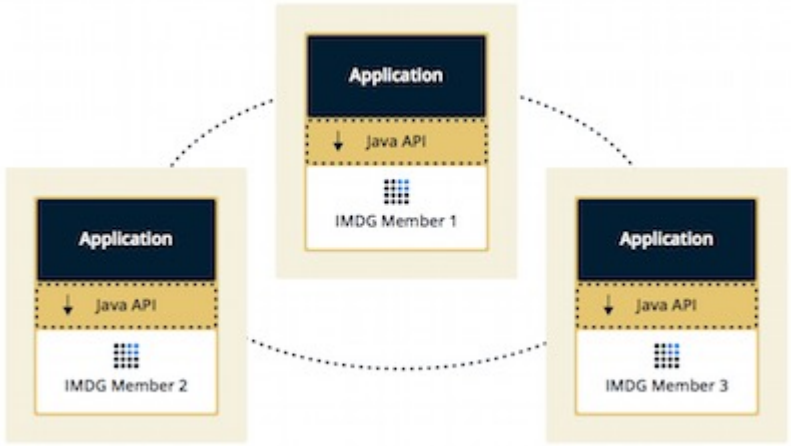
○ **대표적인 IMDG 플랫폼**

- 서버마다 Hazelcast member라는 JAVA process 생성되고, Hazelcast member들이 network으로 서로 연결되어 클러스터 생성
- Java/.NET/Node.js/Python/C++/Go와 같이 다양한 언어의 클라이언트 연결 지원
- 다양한 이기종 DB들(RDBMS, NoSQL, ...)과 연동되고, 기존 대용량 데이터들을 메모리에 로딩하여 응답 속도 개선

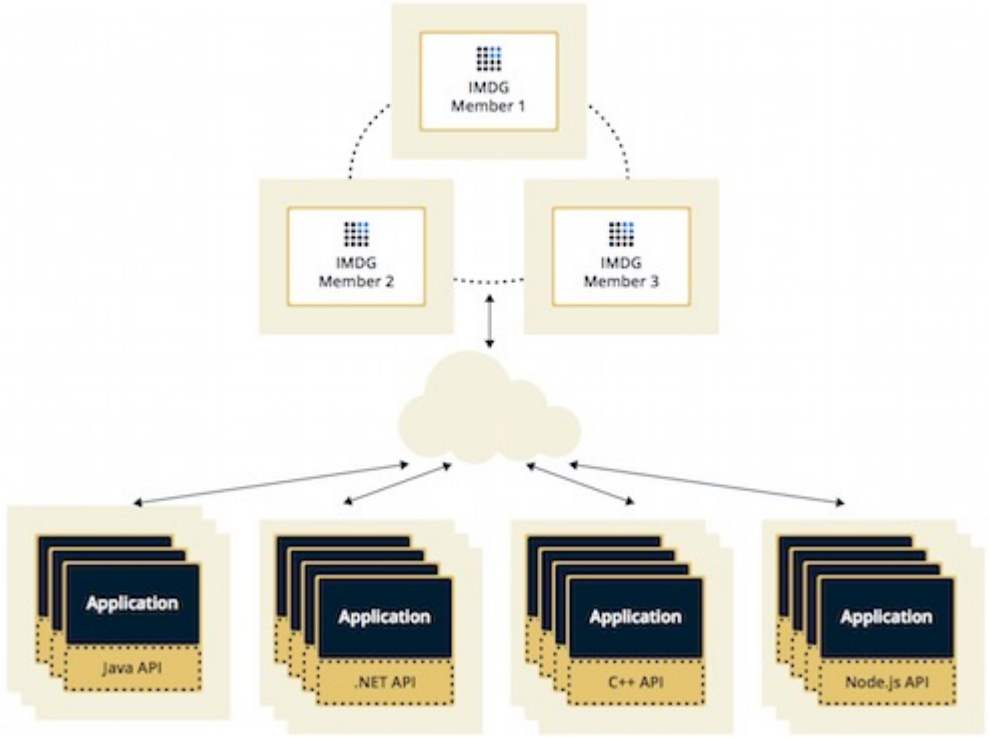


Hazelcast IMDG Cluster 형태

○ **Embedded 모드와 Client/Server 모드로 분류**



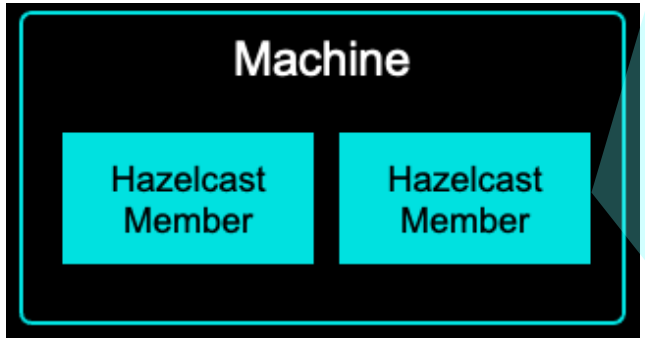
- 자바 App 내부에 hazelcast member 포함하여 실행
- Hazelcsat member 생성 즉시 app 실행

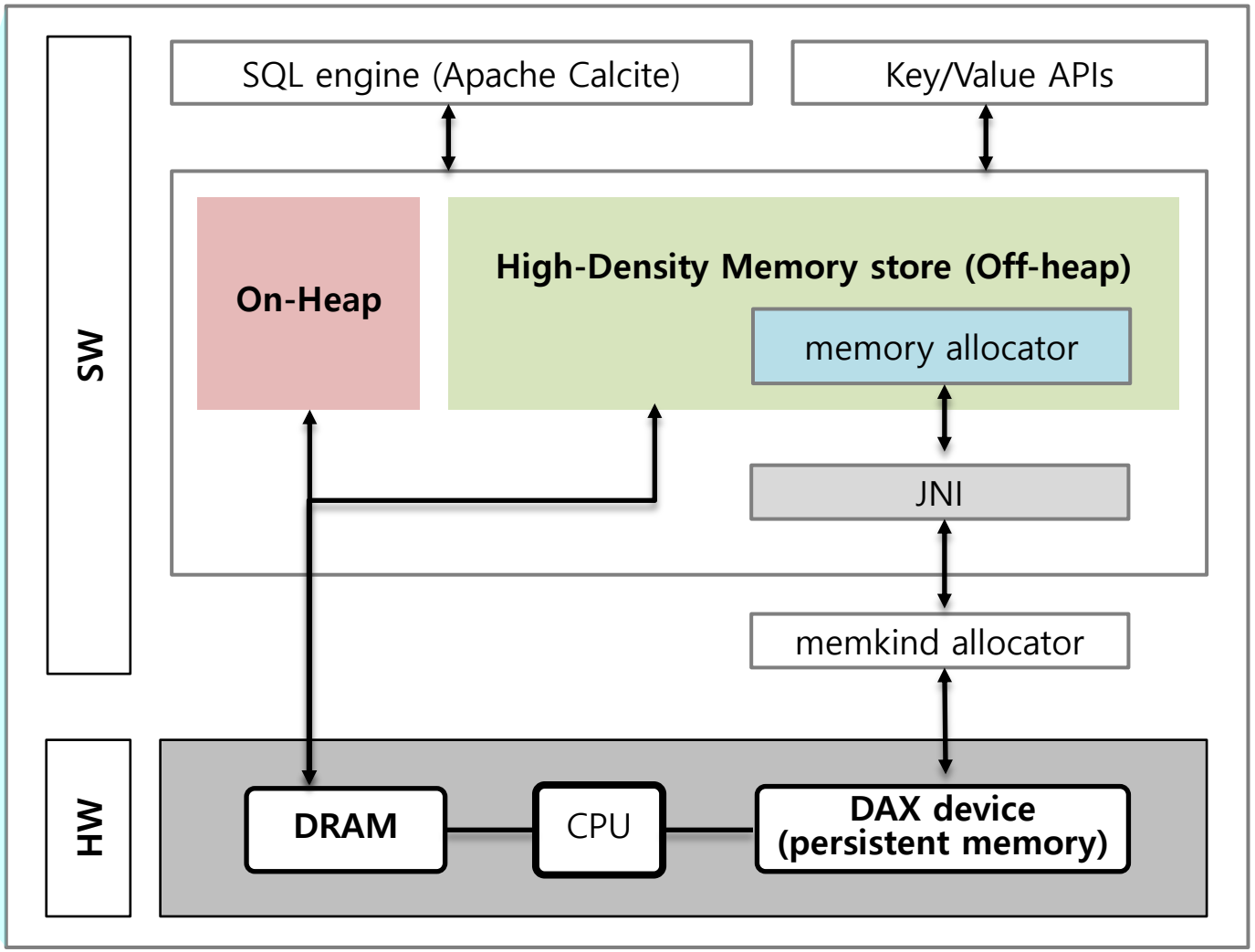


- Hazelcast member만 실행하여 Cluster 구축
- 외부 Client 요청 처리

Hazelcast IMDG software stack

- SQL engine으로 Apache Calcite 탑재, Key/Value API 지원
- Execution 메모리 영역인 On-heap, High-Density Memory Store(HDMS)라 불리는 Off-heap store 구성
- HDMS는 Java Off-heap 활용으로 JVM Garbage Collection 최소화 되기 때문에 메모리에 대용량 데이터 로딩 유리
- HDMS 지원 메모리로 DRAM과 Intel Optane™ DC Persistent Memory 지원

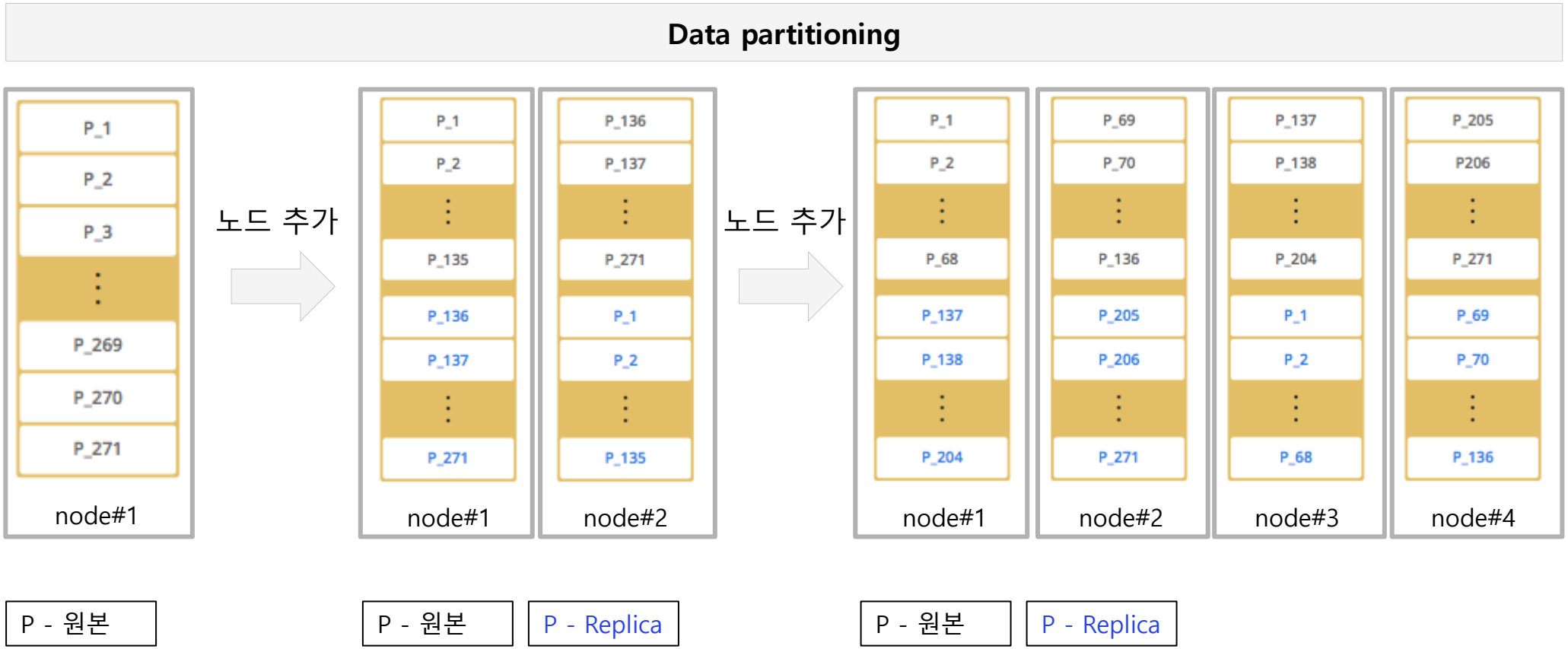




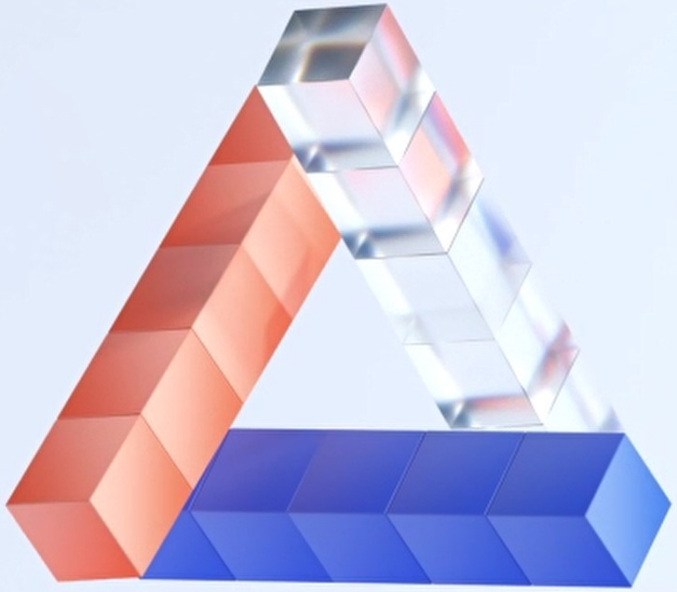
참조 : https://pmem.io/blog/2021/02/using-memkind-in-hazelcast/

Hazelcast IMDG Data partitioning

- ○ 기본 자료 구조 IMap, Java Map의 분산형으로 아래와 같은 data partition 정책을 가짐
- ○ Default로 271개 data partition 생성
- ○ 1개 node 생성 시, 원본 partition만 생성, 2개 node 이상부터 1-원본, 1-replica partition policy 적용되어 생성됨
- ○ 1개 partition = 256MB default 설정이고, memory max size 설정 증가되면 함께 증가됨
- ○ partition ID = HASH(key) mod partition count(default 271) 설정됨



DEVOCEAN



Hazelcast IMDG 사용 방법

Software 설치 방법

○ Software 다운로드

- Open source
  - 패키지 : https://hazelcast.com/open-source-projects/downloads/#hazelcast-platform
  - Github : https://github.com/hazelcast/hazelcast
- Enterprise edition 패키지 : https://hazelcast.com/get-started/download/#hazelcast-platform
  - License key 필요. 30일 free trial license key 제공함

○ 실치(Enterprise edition 기준 설명)

- 리눅스(Ubuntu 20.04.6 LTS) 기준, JDK 함께 설치 필요

```
$ unzip hazelcast-enterprise-5.2.1.zip
$ cd hazelcast-enterprise-5.2.1
$ ls -al
```

```
sejinoh80@sejinoh80:~/hazelcast-enterprise-5.2.1$ ls -al
total 60
drwxrwxr-x  8 sejinoh80 sejinoh80  4096  9월  16 14:15 .
drwxr-xr-x 41 sejinoh80 sejinoh80 12288  9월  16 14:35 ..
drwxr-xr-x  3 sejinoh80 sejinoh80  4096 11월  14  2022 bin
drwxr-xr-x  3 sejinoh80 sejinoh80  4096 11월  14  2022 config
drwxr-xr-x  2 sejinoh80 sejinoh80  4096 11월  14  2022 custom-lib
drwxr-xr-x  2 sejinoh80 sejinoh80  4096 11월  14  2022 lib
drwxrwxr-x  2 sejinoh80 sejinoh80  4096 11월  14  2022 licenses
drwxr-xr-x  3 sejinoh80 sejinoh80  4096 11월  14  2022 management-center
-rw-r--r--  1 sejinoh80 sejinoh80  3485 11월  14  2022 NOTICE
-rw-r--r--  1 sejinoh80 sejinoh80 15864 11월  14  2022 release_notes.txt
```

○ Enterprise license key 설정

- Hazelcast web site(https://hazelcast.com/trial-request) 요청하여 hazelcast configuration 파일에 설정함

Hazelcast IMDG 노드 설정

```
sejinoh80@sejinoh80:~/hazelcast-enterprise-5.2.1$ tree -L 2
.
├── bin
│   ├── common.sh
│   ├── hz
│   ├── hz-cli
│   ├── hz-cli.bat
│   ├── hz-cluster-admin
│   ├── hz-cluster-cp-admin
│   ├── hz-healthcheck
│   ├── hz-start
│   ├── hz-start.bat
│   ├── hz-stop
│   ├── hz-stop.bat
│   └── user-lib
├── config
│   ├── examples
│   ├── hazelcast-client.xml
│   ├── hazelcast-docker.xml
│   ├── hazelcast.xml
│   ├── jmx_agent_config.yaml
│   ├── jvm-client.options
│   ├── jvm.options
│   ├── log4j2-json.properties
│   └── log4j2.properties
├── custom-lib
│   ├── hazelcast-3.12.13.jar
│   ├── hazelcast-3-connector-impl-5.2.1.jar
│   └── hazelcast-client-3.12.13.jar
├── lib
│   ├── cache-api-1.1.1.jar
│   ├── hazelcast-3-connector-common-5.2.1.jar
│   ├── hazelcast-3-connector-interface-5.2.1.jar
│   ├── hazelcast-distribution-5.2.1.jar
│   ├── hazelcast-download.properties
│   ├── hazelcast-enterprise-5.2.1.jar
│   ├── hazelcast-jet-avro-5.2.1.jar
│   ├── hazelcast-jet-cdc-debezium-5.2.1.jar
│   ├── hazelcast-jet-cdc-postgres-5.2.1.jar
│   ├── hazelcast-jet-csv-5.2.1.jar
│   ├── hazelcast-jet-elasticsearch-7-5.2.1.jar
│   ├── hazelcast-jet-files-azure-5.2.1.jar
│   ├── hazelcast-jet-files-gcs-5.2.1.jar
│   ├── hazelcast-jet-files-s3-5.2.1.jar
│   ├── hazelcast-jet-grpc-5.2.1.jar
│   ├── hazelcast-jet-hadoop-all-5.2.1.jar
│   └── hazelcast-jet-kafka-5.2.1.jar
```

○ 주요 설정

– JVM on-heap 메모리 설정

$ vi bin/common.sh

```
#### you can enable following variables by uncommenting them

#### minimum heap size
MIN_HEAP_SIZE=1G

#### maximum heap size
MAX_HEAP_SIZE=1G
```

– HDMS Off-heap storage 메모리, memory format, license key 설정

$ vi config/hazelcast.xml

```
<hazelcast>
    ...
    <native-memory allocator-type="POOLED" enabled="true">
        <size unit="GIGABYTES" value="2"/>
        <min-block-size>16</min-block-size>
        <page-size>4194304</page-size>
        <metadata-space-percentage>12.5</metadata-space-percentage>
    </native-memory>

    <map name="default">
        <!--
           Possible values:
           BINARY (default): keys and values will be stored as binary data
           OBJECT : values will be stored in their object forms
           NATIVE : values will be stored in non-heap region of JVM
        -->
        <in-memory-format>NATIVE</in-memory-format>
     </map>

    <license-key>TrialLicense#i10201011000006901111109010113</license-key>
    ...
</hazelcast>
```

Hazelcast IMDG 노드 실행 및 Cluster 생성

○ **노드 실행**

– ./bin/hz-start 3회 실행 시, 3개의 hazelcast members 생성되면서 클러스터 자동 생성(TCP connection)

$ ./bin/hz-start.sh

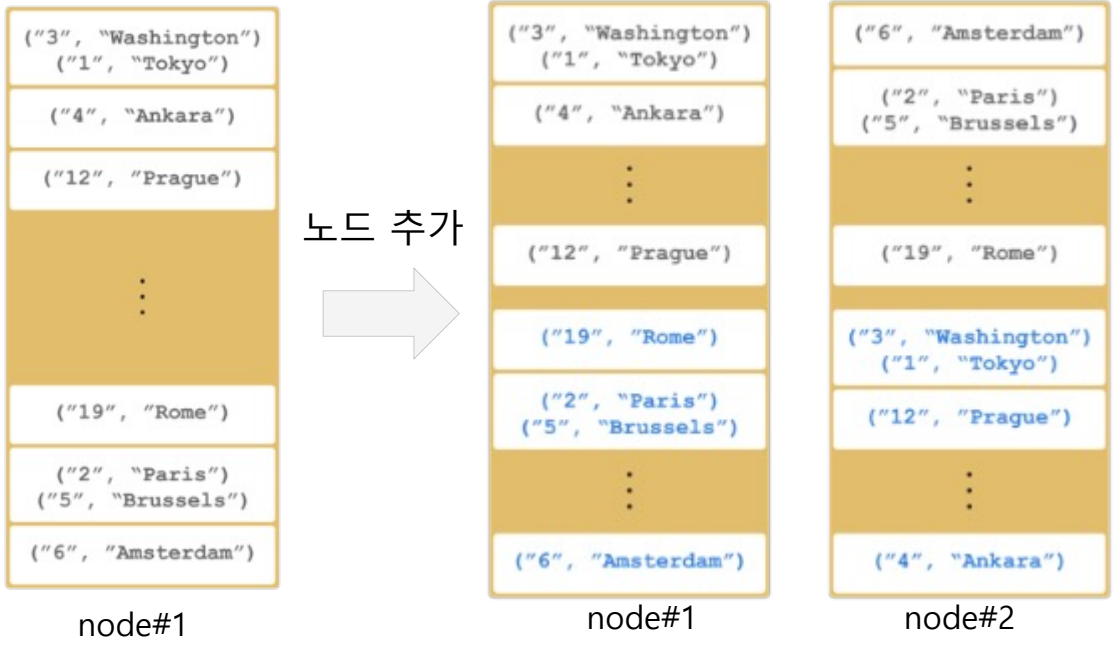
```
sejinoh80@sejinoh80:~/hazelcast-enterprise-5.2.1/bin$ ####################################
# JAVA=/usr/lib/jvm/java-8-openjdk-amd64/bin/java
# JAVA_OPTS=-Dhazelcast.logging.type=log4j2 -Dlog4j.configurationFile=file:/home/sejinoh80/hazelcast-enterprise-5.2.1/config/log4j2.properties -Dhazelcast.config=/home/sejinoh80/hazelcast-enterprise-5.2.1
/config/hazelcast.xml -Djet.custom.lib.dir=/home/sejinoh80/hazelcast-enterprise-5.2.1/custom-lib -Xms1G -Xmx1G
# CLASSPATH=:/home/sejinoh80/hazelcast-enterprise-5.2.1/lib:/home/sejinoh80/hazelcast-enterprise-5.2.1/lib/*:/home/sejinoh80/hazelcast-enterprise-5.2.1/bin/user-lib:/home/sejinoh80/hazelcast-enterprise-5.
2.1/bin/user-lib/*
####################################
2023-09-17 21:20:33,068 [ INFO] [main] [c.h.i.c.AbstractConfigLocator]: Loading configuration '/home/sejinoh80/hazelcast-enterprise-5.2.1/config/hazelcast.xml' from System property 'hazelcast.config'
2023-09-17 21:20:33,071 [ INFO] [main] [c.h.i.c.AbstractConfigLocator]: Using configuration file at /home/sejinoh80/hazelcast-enterprise-5.2.1/config/hazelcast.xml
2023-09-17 21:20:33,797 [ INFO] [main] [c.h.i.AddressPicker]: [LOCAL] [dev] [5.2.1] Interfaces is enabled, trying to pick one address matching to one of: [127.0.0.1, 127.0.0.1]
2023-09-17 21:20:33,906 [ INFO] [main] [c.h.s.logo]: [127.0.0.1]:5703 [dev] [5.2.1]
        +       +  o    o     o     o---o o----o o      o---o     o     o----o o--o--o
        + +   + +  |    |    / \       /  |      |     /         / \    |         |
        + + + + +  o----o   o   o     o   o----o |    o         o   o   o----o    |
        + +   + +  |    |  /     \   /    |      |     \       /     \       |    |
        +       +  o    o o       o o---o o----o o----o o---o o       o o----o    o
2023-09-17 21:20:33,907 [ INFO] [main] [c.h.system]: [127.0.0.1]:5703 [dev] [5.2.1] Copyright (c) 2008-2022, Hazelcast, Inc. All Rights Reserved.
2023-09-17 21:20:33,907 [ INFO] [main] [c.h.system]: [127.0.0.1]:5703 [dev] [5.2.1] Hazelcast Enterprise 5.2.1 (20221114 - 041c2a1, 531032a) starting at [127.0.0.1]:5703
2023-09-17 21:20:33,908 [ INFO] [main] [c.h.system]: [127.0.0.1]:5703 [dev] [5.2.1] Cluster name: dev
2023-09-17 21:20:33,908 [ INFO] [main] [c.h.system]: [127.0.0.1]:5703 [dev] [5.2.1] Integrity Checker is disabled. Fail-fast on corrupted executables will not be performed. For more information, see the d
ocumentation for Integrity Checker.
2023-09-17 21:20:33,908 [ INFO] [main] [c.h.system]: [127.0.0.1]:5703 [dev] [5.2.1] Jet is enabled
2023-09-17 21:20:33,910 [ INFO] [main] [c.h.i.i.NodeExtension]: [127.0.0.1]:5703 [dev] [5.2.1] Checking Hazelcast Enterprise license...
2023-09-17 21:20:33,933 [ INFO] [main] [c.h.i.i.NodeExtension]: [127.0.0.1]:5703 [dev] [5.2.1] License{allowedNumberOfNodes=10, expiryDate=Oct/16/2023 23:59:59, featureList=[ Management Center, Streaming
Management Center, Clustered JMX, Clustered REST, Security, WAN Replication, High Density Memory, Persistence, Streaming Lossless Cluster Restart, Rolling Upgrade, Streaming Job Upgrades, Cluster Client F
iltering, CP Subsystem Persistence, Dynamic Configuration, Tiered Storage ], type=null, companyName=null, ownerEmail=null, keyHash=LOLI9peVksfQLm+gcdYF1P0XomTt7BTrS2D48jy+zGI=, No Version Restriction}
2023-09-17 21:20:33,935 [ INFO] [main] [c.h.i.i.NodeExtension]: [127.0.0.1]:5703 [dev] [5.2.1] Creating POOLED native memory manager with 2048 MB size
2023-09-17 21:20:34,417 [ INFO] [main] [c.h.e.w.i.WanThrottlingAcknowledger]: [127.0.0.1]:5703 [dev] [5.2.1] Using throttling WAN acknowledgement strategy with pending invocation threshold 50000
2023-09-17 21:20:34,655 [ INFO] [main] [c.h.s.security]: [127.0.0.1]:5703 [dev] [5.2.1] Enable DEBUG/FINE log level for log category com.hazelcast.system.security  or use -Dhazelcast.security.recommendati
ons system property to see  security recommendations and the status of current config.
2023-09-17 21:20:34,748 [ INFO] [main] [c.h.i.i.Node]: [127.0.0.1]:5703 [dev] [5.2.1] Using TCP/IP discovery
2023-09-17 21:20:34,749 [ WARN] [main] [c.h.c.CPSubsystem]: [127.0.0.1]:5703 [dev] [5.2.1] CP Subsystem is not enabled. CP data structures will operate in UNSAFE mode! Please note that UNSAFE mode will no
t provide strong consistency guarantees.
2023-09-17 21:20:35,224 [ INFO] [main] [c.h.j.i.EnterpriseJetServiceBackend]: [127.0.0.1]:5703 [dev] [5.2.1] Setting number of cooperative threads and default parallelism to 4
2023-09-17 21:20:35,359 [ INFO] [main] [c.h.i.d.Diagnostics]: [127.0.0.1]:5703 [dev] [5.2.1] Diagnostics disabled. To enable add -Dhazelcast.diagnostics.enabled=true to the JVM arguments.
2023-09-17 21:20:35,375 [ INFO] [main] [c.h.c.LifecycleService]: [127.0.0.1]:5703 [dev] [5.2.1] [127.0.0.1]:5703 is STARTING
2023-09-17 21:20:35,461 [ INFO] [hz.vigorous_khorana.IO.thread-in-0] [c.h.i.s.t.TcpServerConnection]: [127.0.0.1]:5702 [dev] [5.2.1] Initialized new cluster connection between /127.0.0.1:5702 and /127.0.0
.1:45799
2023-09-17 21:20:35,462 [ INFO] [hz.frosty_mahavira.IO.thread-in-2] [c.h.i.s.t.TcpServerConnection]: [127.0.0.1]:5701 [dev] [5.2.1] Initialized new cluster connection between /127.0.0.1:5701 and /127.0.0.
1:38105
2023-09-17 21:20:35,472 [ INFO] [hz.modest_ardinghelli.IO.thread-in-1] [c.h.i.s.t.TcpServerConnection]: [127.0.0.1]:5703 [dev] [5.2.1] Initialized new cluster connection between /127.0.0.1:38105 and /127.
0.0.1:5701
2023-09-17 21:20:35,472 [ INFO] [hz.modest_ardinghelli.IO.thread-in-0] [c.h.i.s.t.TcpServerConnection]: [127.0.0.1]:5703 [dev] [5.2.1] Initialized new cluster connection between /127.0.0.1:45799 and /127.
0.0.1:5702
2023-09-17 21:20:40,557 [ INFO] [hz.frosty_mahavira.generic-operation.thread-0] [c.h.i.c.ClusterService]: [127.0.0.1]:5701 [dev] [5.2.1]

Members {size:3, ver:3} [
        Member [127.0.0.1]:5701 - 4ffd47aa-8043-495f-9b1b-3d82cccb20a6 this
        Member [127.0.0.1]:5702 - 82fda1bc-c89d-4cb2-ac68-6f655c943f0a
        Member [127.0.0.1]:5703 - 656f62fa-b28f-447f-a658-7f31dd836893
]
```

3개 hazelcast member 생성됨

SQL 활용한 IMap 생성

○ **IMap 형식**

```
CREATE MAPPING my_map
TYPE IMap
OPTIONS (
    'keyFormat'='int',
    'valueFormat'='varchar'
)
```

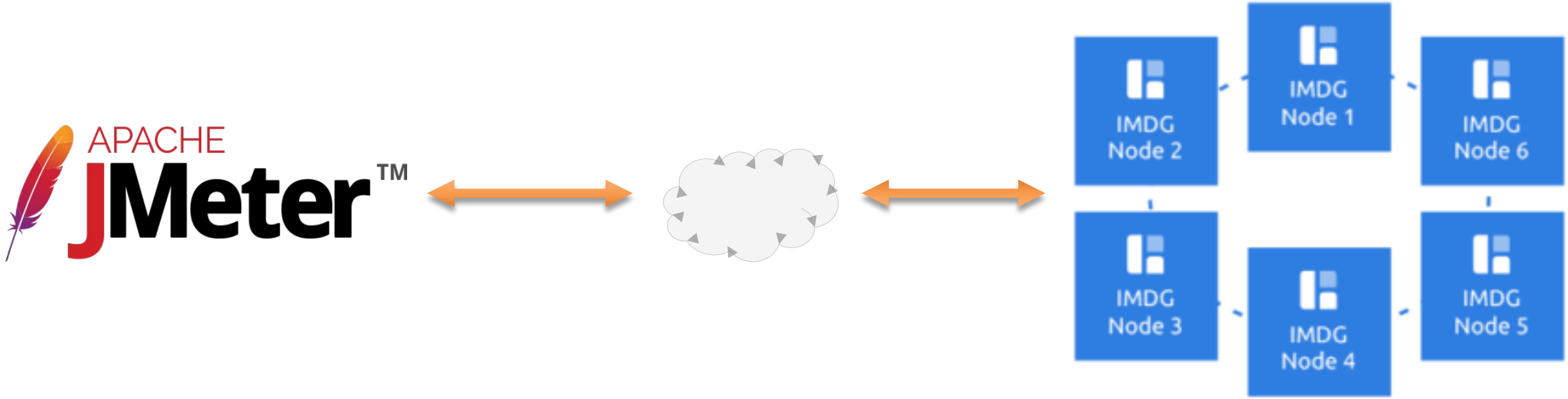


Hazelcast CLI에서 SQL로 생성 방법

```
$ ./bin/hz-cli --config config/hazelcast-client.xml sql
sql> CREATE MAPPING capitals
   >  TYPE IMap
   >  OPTIONS
   > ('keyFormat'='varchar','valueFormat'='varchar');
OK
sql> SINK INTO capitals VALUES
   > ( "1", "Tokyo" ),
   > ( "2", "Paris" ),
   > ( "3", "Washington" ),
   > ( "4", "Ankara" ),
   > ( "5", "Brussels" ),
   > ( "6", "Amsterdam" ),
   > ( "7", "New Delhi" ),
   > ( "8", "London" ),
   > ( "9", "Berlin" ),
   > ( "10", "Oslo" ),
   > ( "11", "Moscow" ),
     ...
     ( "120", "Stockholm" );
sql> SELECT * FROM capitals;
+--------------------+--------------------+
|__key               |this                |
+--------------------+--------------------+
|9                   |Berlin              |
|11                  |Moscow              |
|3                   |Washington          |
|8                   |London              |
......
n row(s) selected
```

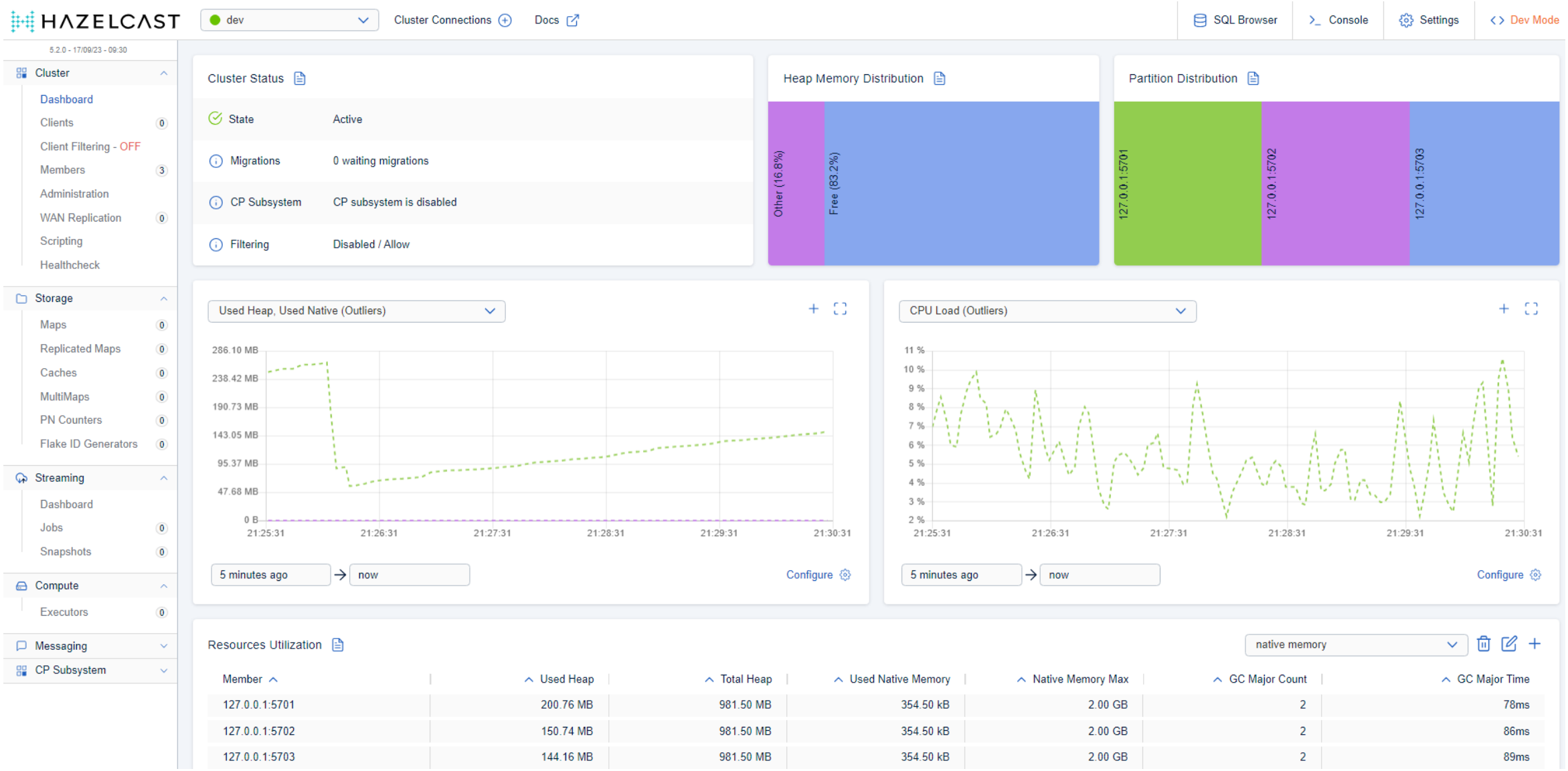
JDBC 연결 통한 SQL 질의

- Apache JMeter와 같은 JDBC 클라이언트 툴 활용하여 SQL 질의 가능
- Hazelcast JDBC 드라이버는 https://github.com/hazelcast/hazelcast-jdbc 다운로드 가능

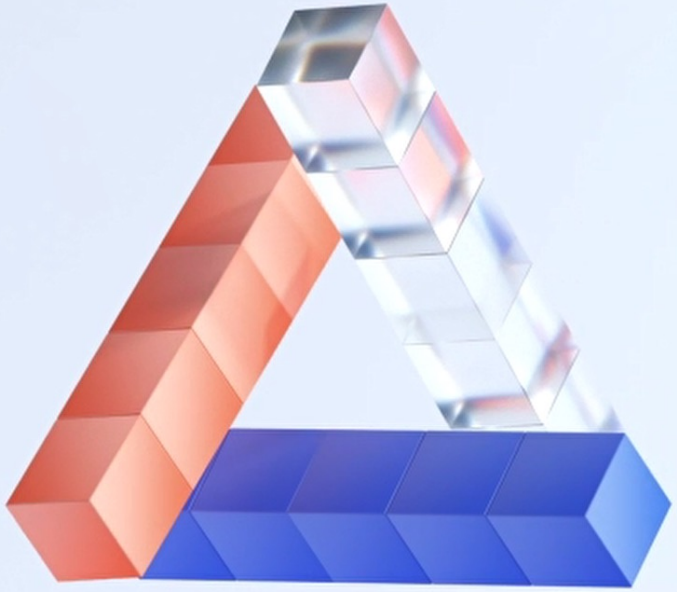


Hazelcast Management Center

○ **Web 기반 모니터링 tool, 기본 SW 패키지에 포함됨**



DEVOCEAN

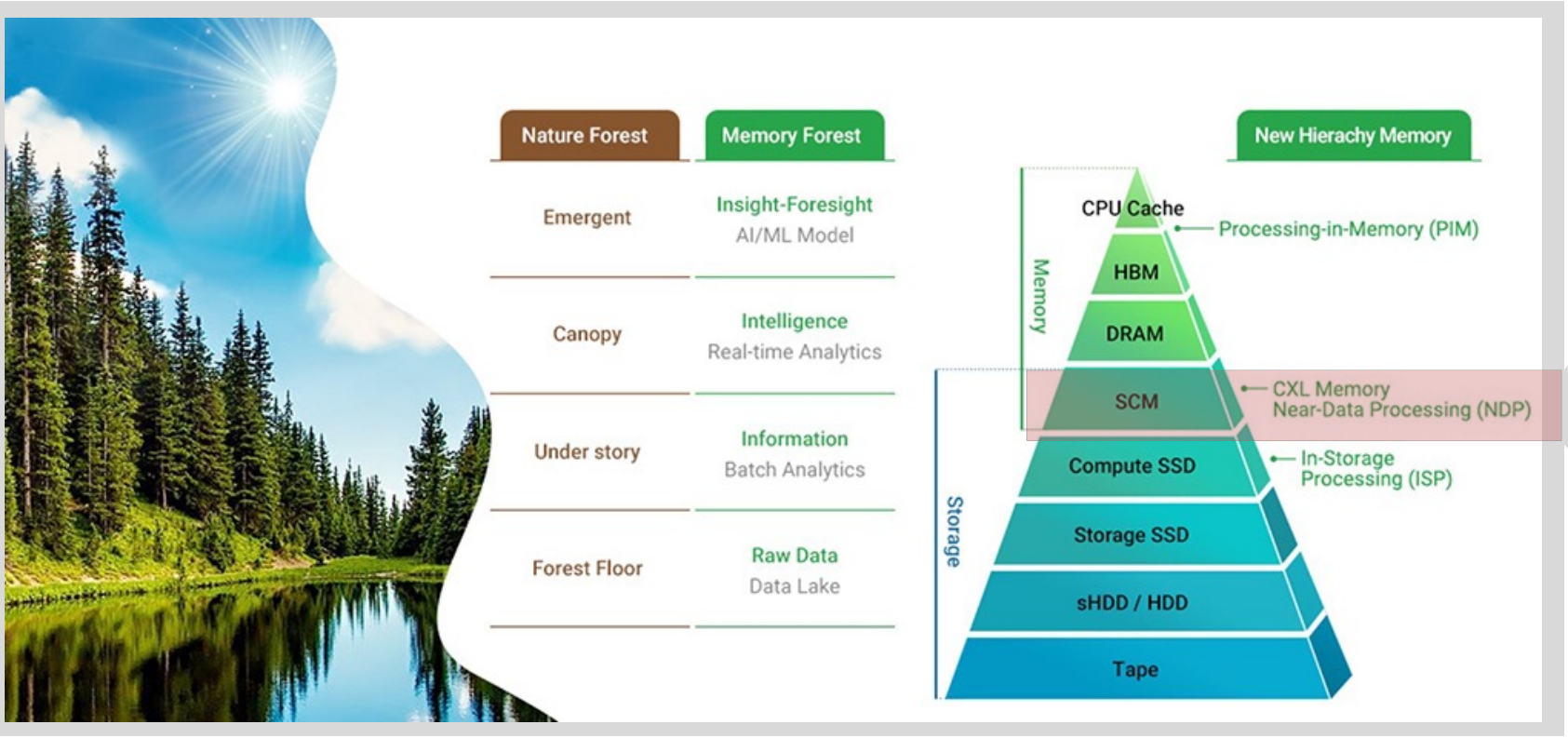


Hazelcast IMDG와 SK hynix's CXL™ Memory

SK Hynix's CXL™ memory

○ **CXL이란**
  - Compute Express Link의 약자로 PCIe slot을 이용하는 차세대 고성능 interface
  - Intel 4세대 Xeon Scalable processor(code name: Sapphire Rapids) 부터 지원

○ **SK hynix's CXL 2.0 Memory Expansion Solution**
  - 휘발성 메모리, E3.S EDSFF(Enterprise & Data Center Standard Form Factor), PCIe 5.0 x8 Lane 지원, 내부 DDR5 표준 DRAM 사용

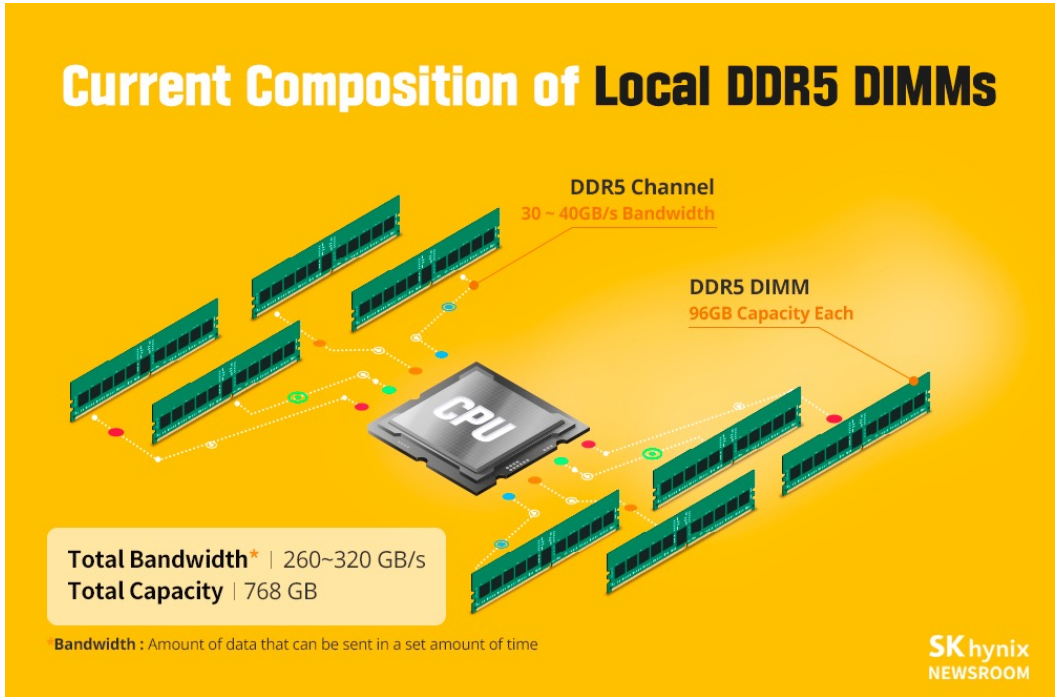


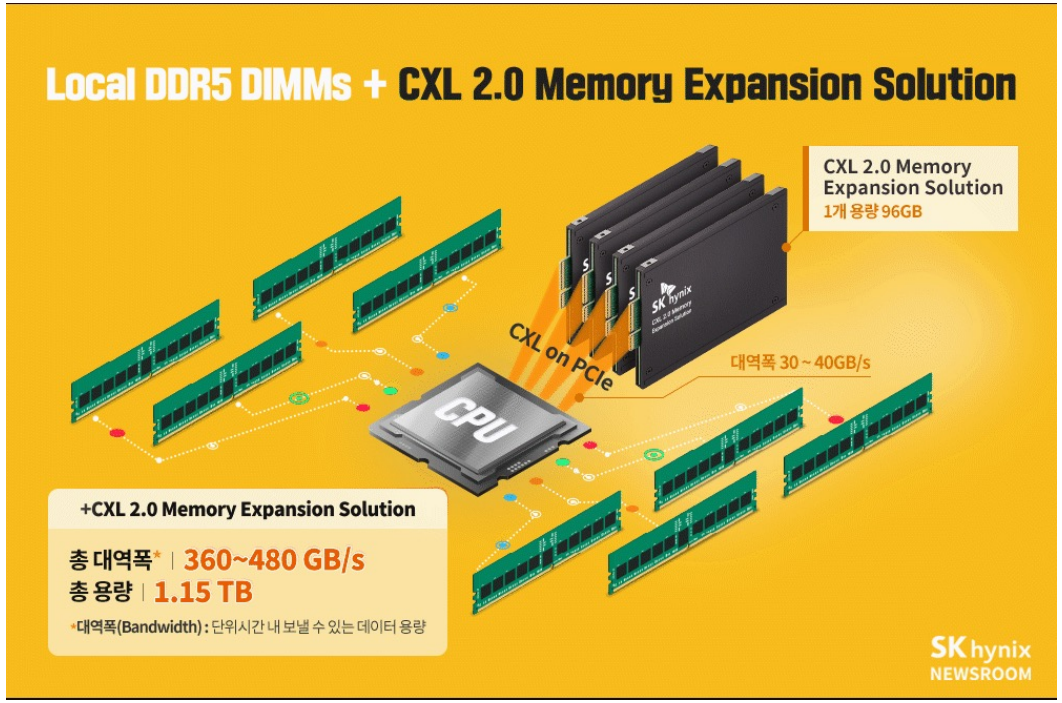
참조 : https://news.skhynix.com/sk-hynix-develops-ddr5-dram-cxltm-memory-to-expand-the-cxl-memory-ecosystem/

참조 : https://news.skhynix.com/sk-hynix-at-nvidia-gtc-2022-demonstrating-the-worlds-fastest-dram-hbm3/

SK Hynix's CXL™ memory 특징

- ○ Local DRAM에 추가적으로 Memory bandwidth 및 capacity 동시 확장
- ○ 서버 추가 없이 메모리 확장할 수 있는 장점 보유

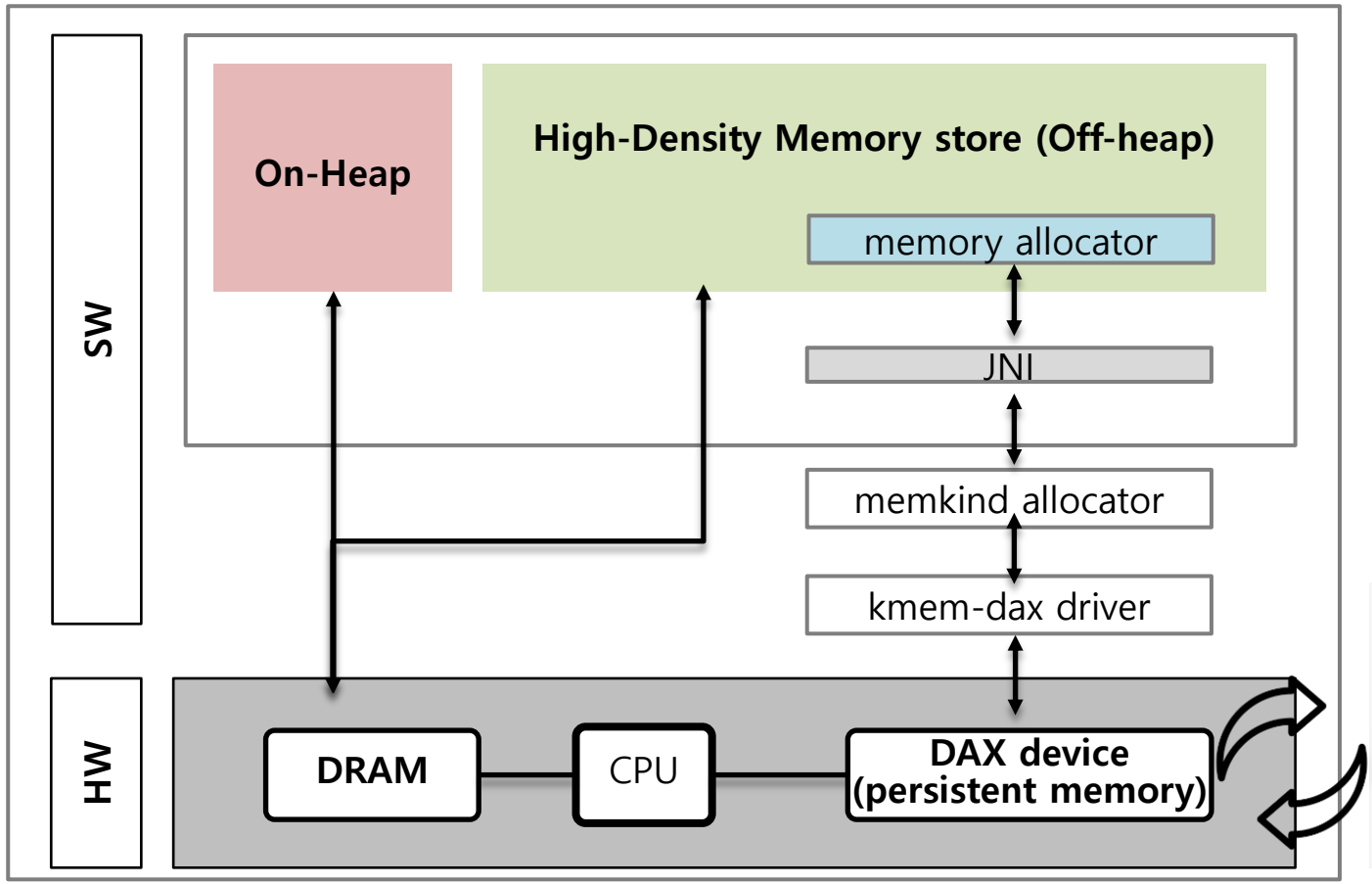




참조 : https://news.skhynix.com/sk-hynix-develops-ddr5-dram-cxltm-memory-to-expand-the-cxl-memory-ecosystem/

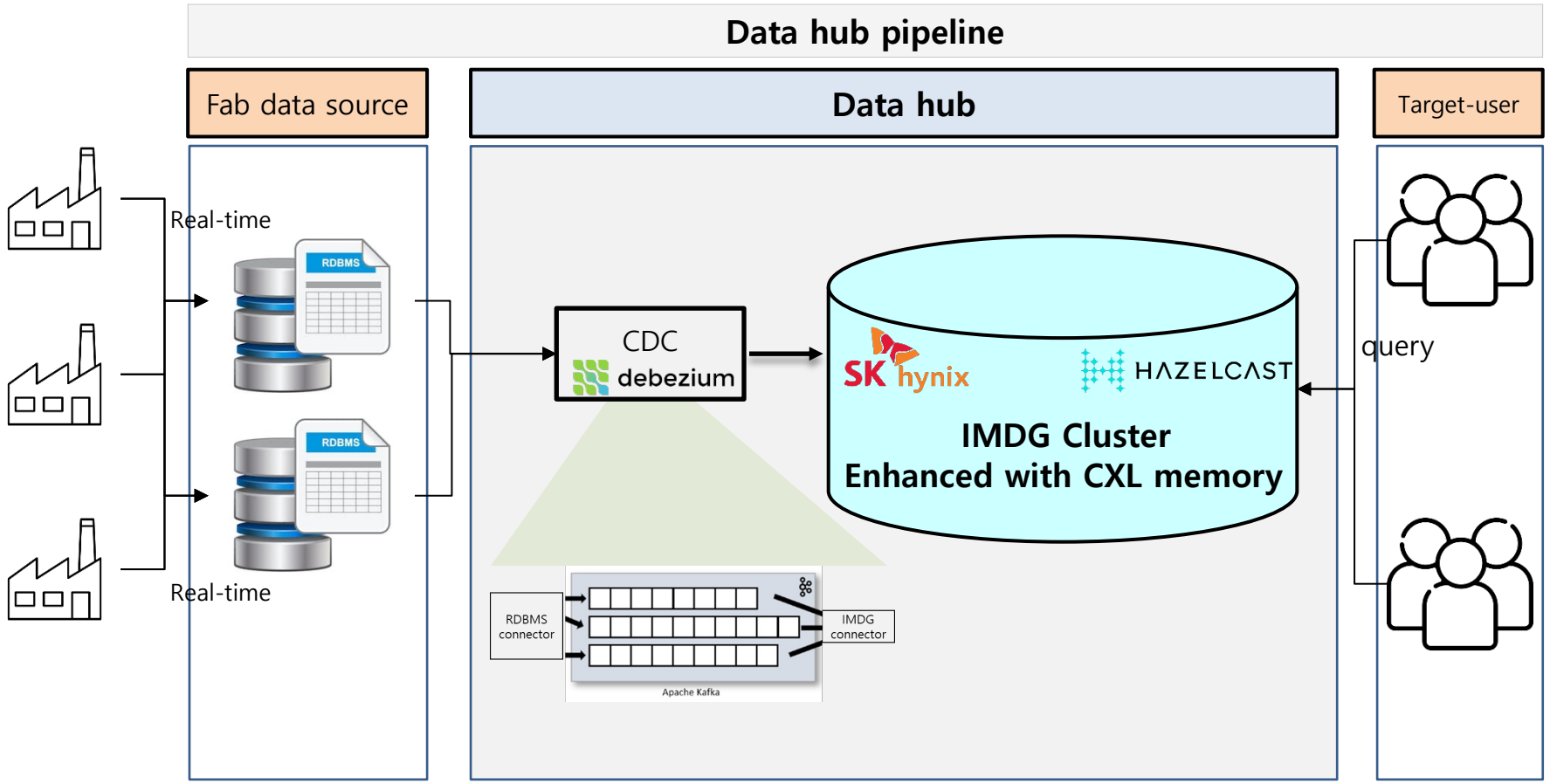
IMDG integrated w/ SK hynix's CXL™ memory

- CXL memory는 Linux에서 DAX(Direct Access) device로 인식됨
- daxctl reconfigure-device cmd 사용하여 system-ram mode로 configure하면 추가 memory numa node로 설정됨
- "LP_PRELOAD=Intel memkind library path" 선언으로 HDMS에 CXL memory 연결 가능함



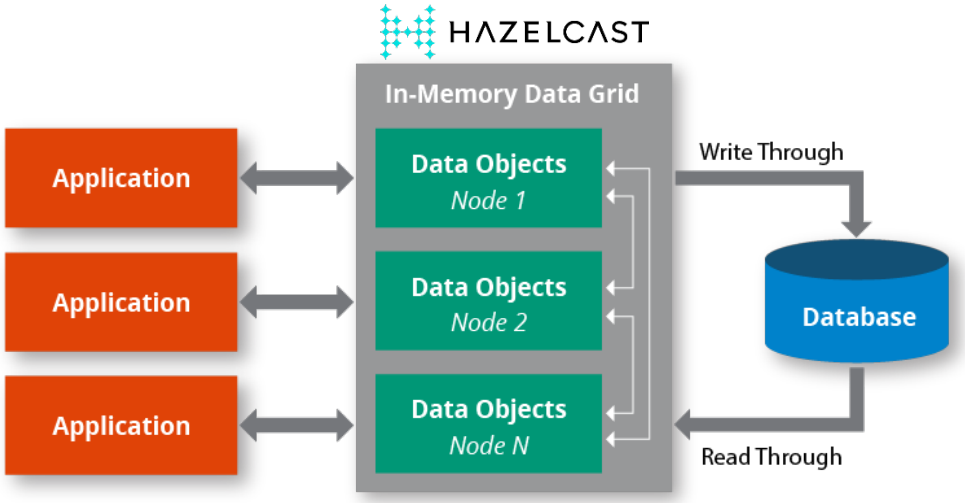
CXL-enabled IMDG와 SK hynix's Data hub

- ○ SK hynix 전사 실시간 데이터 공유 플랫폼인 Data hub 환경에 차세대 데이터 아키텍처 구축 위한 PoC 진행
- ○ Data hub 내부 모사하여 SK hynix DDR5 DRAM과 CXL memory 탑재된 서버에 소규모 Hazelcast IMDG Cluster 구축



마무리

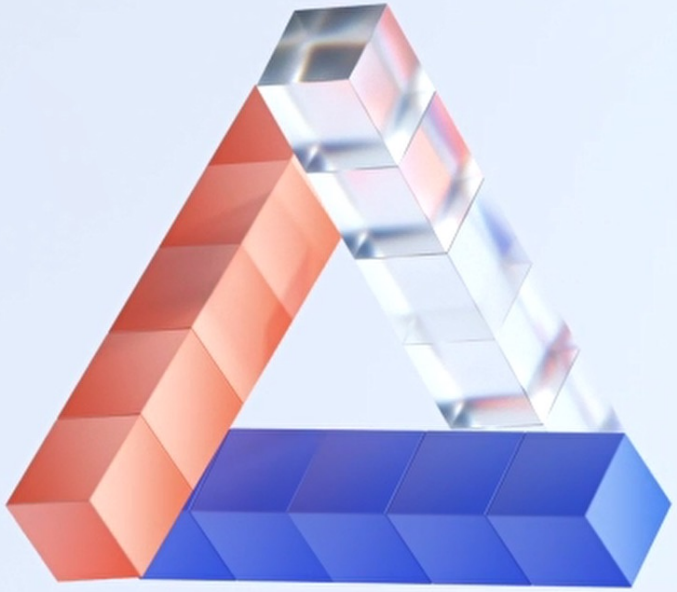
○ 데이터 통합 관리 위한 In-memory distributed caching software인 IMDG architecture 확인



○ Hazelcast IMDG에 SK hynix's CXL memory integration 통해, 메모리 value-add 가능성 탐색

Hazelcast IMDG에 SK hynix's CXL memory 적용한 상세 use-case 및 자세한 성능 효과는 SK ICT SUMMIT(11/16일)에서 확인 가능합니다

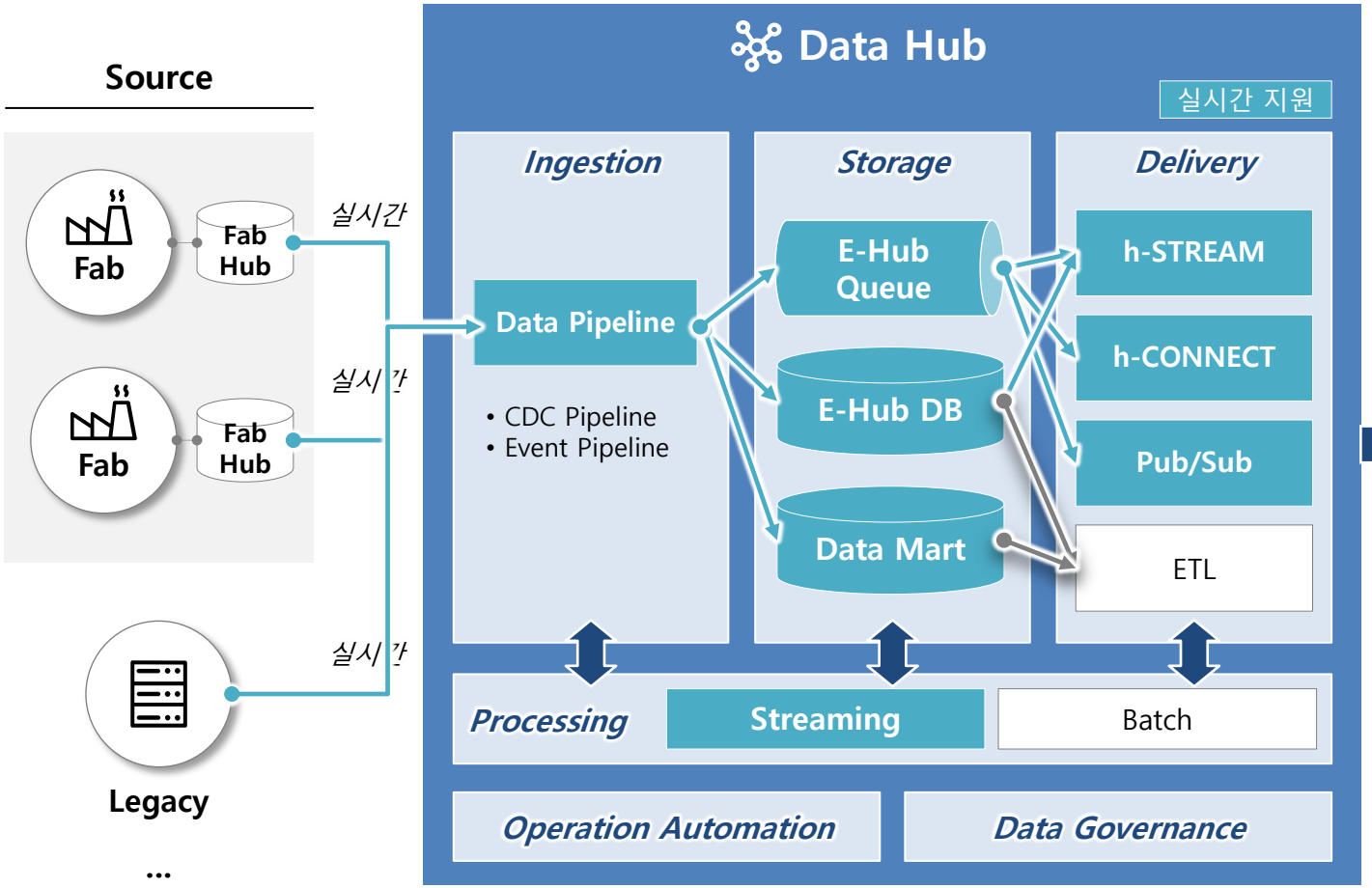
DEVOCEAN



Data Hub

Data Hub 아키텍처

전사 데이터 Interface 단일 Hub 역할을 수행하는 실시간 데이터 공유 플랫폼



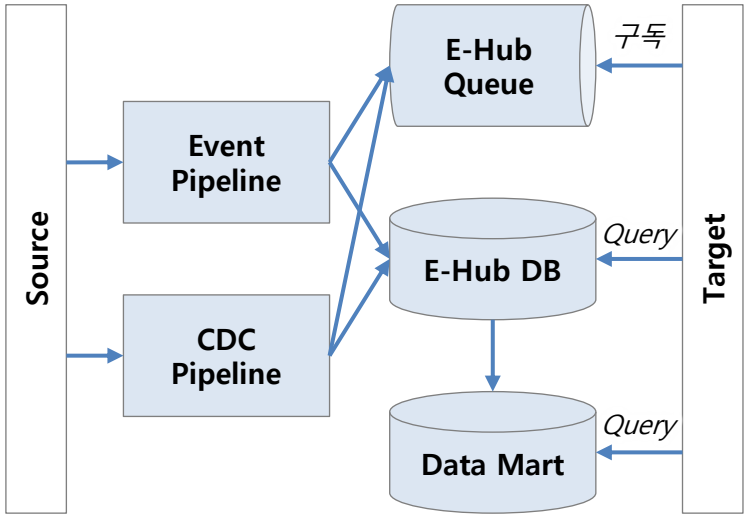
주요 특징

- **전사 Data Interface의 단일 Hub** 역할 수행
- **Source ↔ Target 間 실시간 데이터 공유** 지원
- **실시간 데이터 분석 정합성 제고**를 위한 **Data Mart** 운영
- **다양한 Data Pipeline** 제공하여 사용자 **Delivery 지원**
- Data Hub 중심 I/F를 구현하기 위한 **Governance 정책** 설계

Data Hub 특징

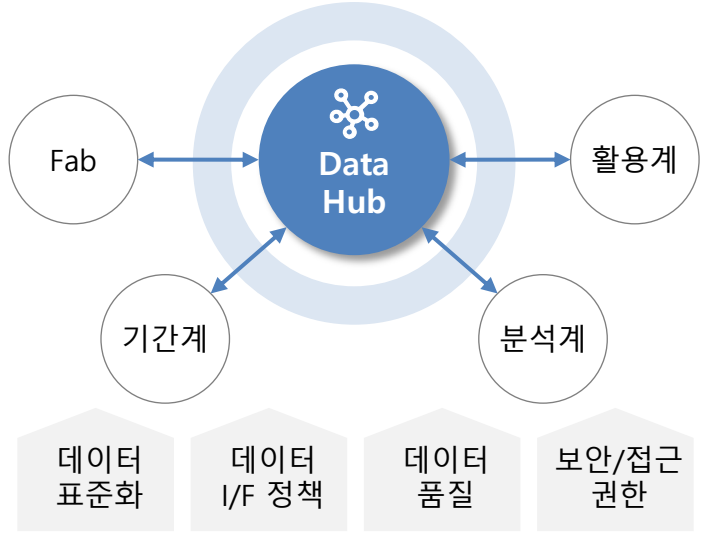
Data Hub는 전사 데이터 연계를 위한 Hub 역할을 수행하며,
Source와 Target 간 실시간 데이터 Pipeline으로서 데이터 활용의 신속/적시성을 지원함

① 실시간 데이터 연계



- **End-to-End Near Real-Time 데이터 연계** (데이터 발생 시점부터 10초 이내)
- E-Hub Queue 적재 데이터 **실시간 구독**
- 실시간성 **Mart向 Ad-hoc Query** 실행 가능

② 전사 Interface Hub 역할 수행



- 전사 데이터 연계 **Single Access Point**
- I/F 일원화를 통한 **기간계 운영 부담 경감**
- 통일된 I/F 운영 기준 기반 **데이터 품질 및 일관성 확보**

③ 다양한 Delivery Pipeline

h-STREAM (Streaming Data Service)
- Sandbox 환경에서 사용자가 Streaming 데이터를 DIY 조립/생성
- 신규 생성된 데이터를 사용자의 Notebook 환경에 전달

h-CONNECT
- 실시간 변경 데이터를 사용자가 지정한 Table과 지속적으로 자동 동기화

Pub/Sub
- E-Hub Queue 적재 데이터를 Target 시스템에서 구독

- 사용자가 **실시간 데이터를 목적에 맞게 즉시 활용**할 수 있도록 Delivery Pipeline 제공
- VM/Notebook으로 안정적인 적재 지원

Data Hub IMDG 적용 PoC > Summary

1. 검증 목적

- **향후 Data Hub가 사용자 Ad-hoc 쿼리 서비스를 실시간으로 제공하기 위하여** 아래 항목을 검증함
  - 데이터 수집 Pipeline(Event/CDC)을 고려한 기능 및 성능 검증
  - 동시 다중 사용자의 다중 쿼리 요청에 대해 실시간 대응 가능 여부 검증

2. 검증 제약 및 고려 사항

- 인력 부족 - Hazelcast Korea의 전문 검증 인력 부재와 Data Hub측의 검증 인력 리소스 부족
- CDC 역할의 오픈소스 Debezium은 Hazelcast 서비스 Package에 포함되지 않음 - 사용자측에서 개발/운영 필요

3. 검증 결과

- 기능 검증 : 8/11 (적합건수/전체건수)
- 성능 검증에서 단순 insert 성능은 Hazelcast가 높게 나왔으나, 현 데이터량 기준으로는 기존 Oracle 기반의 성능이 더 좋아 추가 확인 필요

→ **1차 검증 결과와 CDC 솔루션 미포함 조건으로는 IMDG 활용 목적의 기대효과를 충족하지 못할것으로 판단함**

4. 추가 필요 검증 항목

- 동시 사용자 성능 검증
- Hazelcast 기반의 다양한 CDC 검증 (트랜잭션보장, xSTREAM 프로토콜 적용, OGG 기능 비교등)

Data Hub IMDG 적용 PoC > 기능 검증

기능 검증

- Hazelcast가 기존의 Data Hub(Oracle)을 대체하여 일반 User들에게도 직접 접근이 가능한지 확인하기 위하여 기능 검증 진행

구분	항목	Oracle(E-Hub)	HazelCast
DB CRUD	PK 지원	○	○
	Select	○	○
	Insert	○	○
	Update	○	○
	Delete	○	○
	Merge into	○	X
기타 SQL문	Sub Query	○	X
	Join	○	○
	Group by	○	○
	Order by	○	○
	Like	○	○

- HazelCast의 경우 Oracle의 Merge Into 구문은 지원하지 않고 대신 Sink into 를 지원하여 대체하여 사용함.
  - Merge into는 ETL 툴에서 데이터 적재 시 많이 사용
- **Sub Query/Scalar Query를 지원하지 않음. Sub Query의 경우 사용자들이 많이 사용하는 유형으로 불편함이 있을 것으로 보임.**
- Oracle의 **sysdate 함수 지원하지 않음**. 아래와 같이 사용해야 함
  - TO_TIMESTAMP_TZ(TO_EPOCH_MILLIS(TIMESTAMP'2023-06-05 00:00:00'))

Data Hub IMDG 적용 PoC > 성능 검증

성능 검증

- Data Hub(Oracle)를 대체 성능이 되는지 확인하기 위하여 Event 적재를 하는 검증 진행
- HazelCast의 최대 성능이 어느 정도 수준인지 확인하기 위한 검증 진행
- Data Hub 에 Event를 적재하는 Process는 Data를 전처리하는 Basedata 와 Data를 DB에 적재하는 Inline Process로 나누어져 있음.
- 검증 환경

[Basedata Processing]

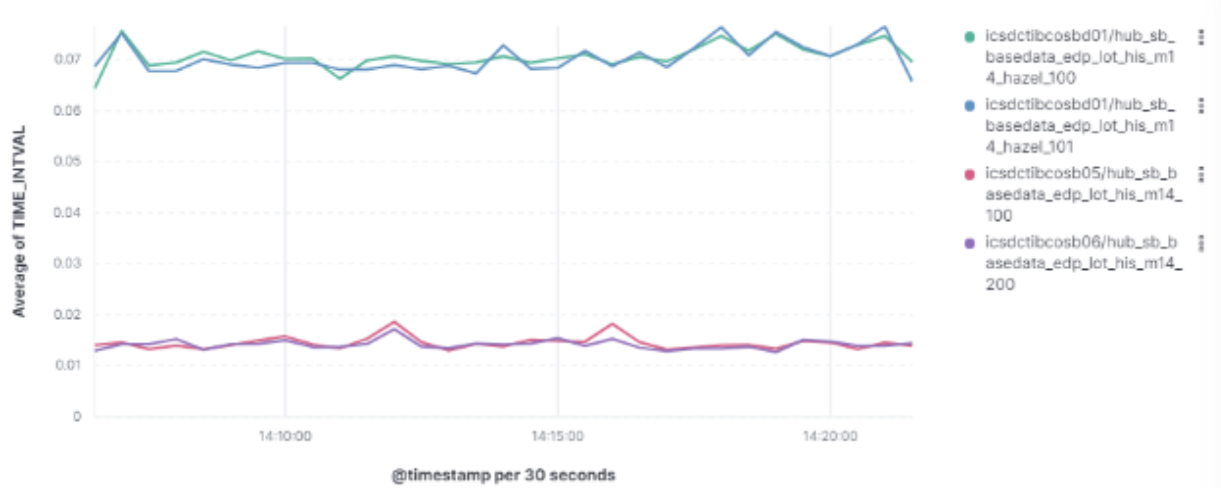
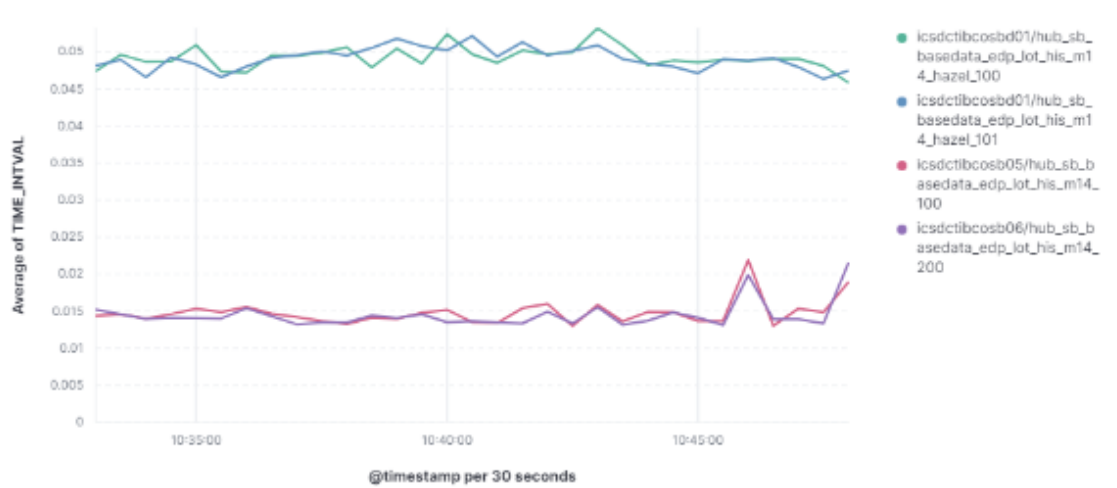
Process	Process Count	thread	JVM	비고
Oracle Prod Process	2	40 (각 20)	-Xms1g -Xms4g	
Hazel QA Process	2	40 (각 20)	-Xms1g -Xms4g	SCALAR QUERY를 adapter 2개로 나누어 하도록 Adatpee 분리

[Inline Processing]

LotHis inline	Process Count	thread	JVM	비고
Oracle Prod Process	2	2 (각 1)	-Xms1g -Xms4g	
Hazel QA Process	2	2 (각 1)	-Xms1g -Xms4g	

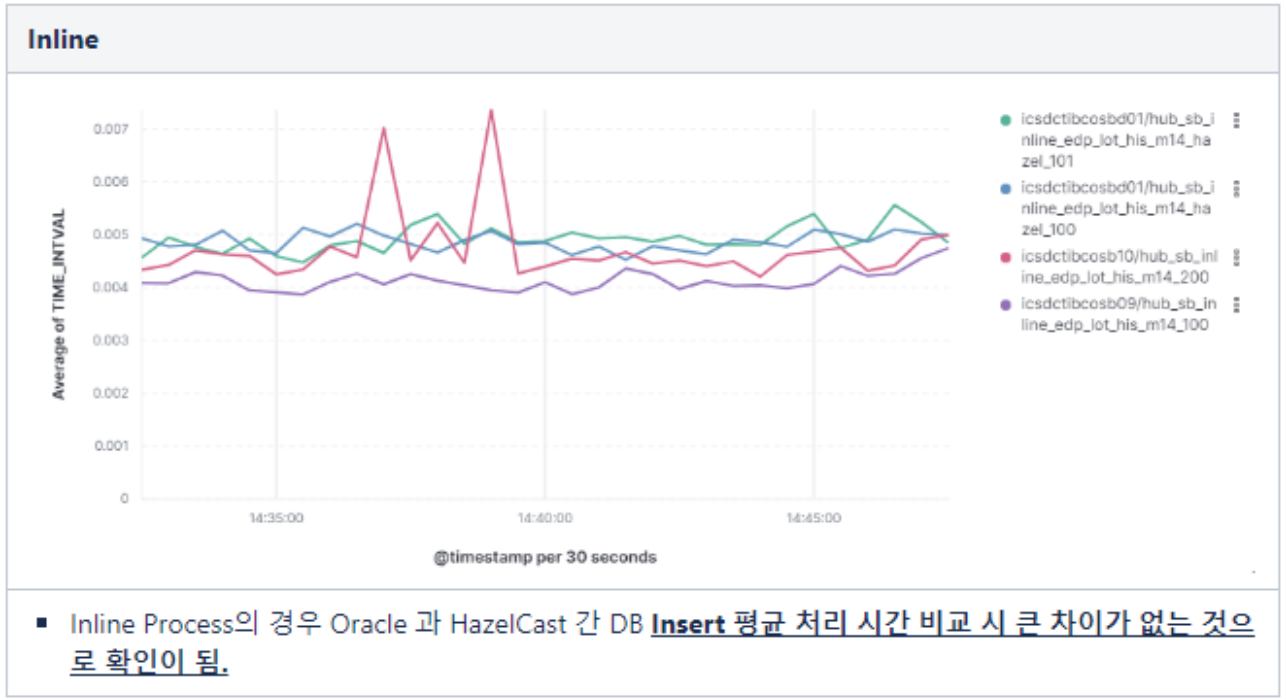
Data Hub IMDG 적용 PoC > 성능 검증

[Basedata Processing]

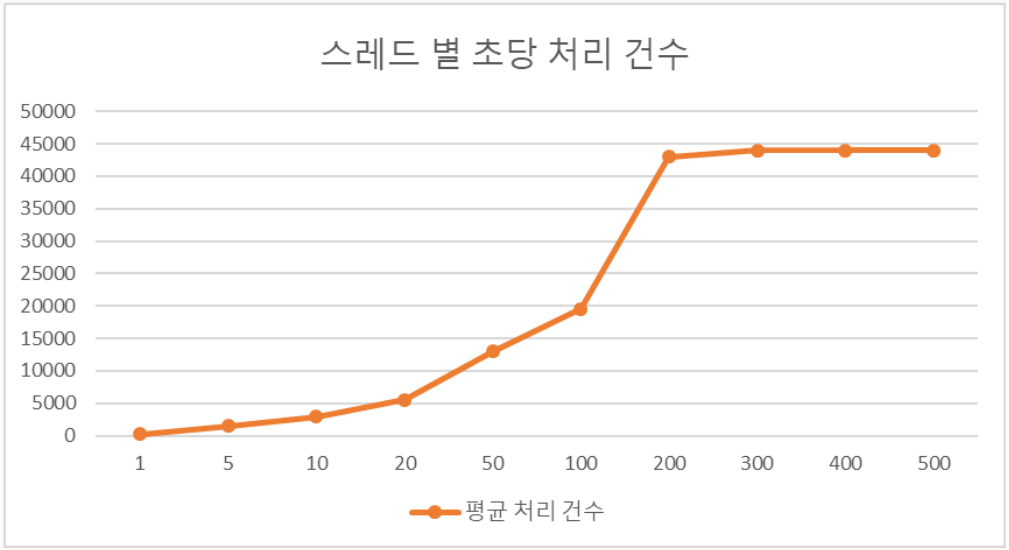
Basedata(Compact)	Basedata(추가 Json-flat)
	
▪ Basedata Process에서 Oper_desc을 조회하는 간단한 쿼리의 경우 **Oracle은 평균 0.015초 Hazelcast는 평균 0.07초로 처리 시간이 5배 정도 더 걸리는 것**을 확인. ▪ 속도 개선을 위해서 Index 추가를 진행해 보았지만 효과 없었음. ▪ Map 생성 시 Compact Type으로 생성을 했는데 __key값을 가진 json-flat 구조로 하면 더 빨라지지 않을까 하여 추가 테스트를 진행함.	▪ **json-flat**으로 기준 정보 테이블을 변경 후 테스트 해본 결과 Hazelcast의 평균 0.05초로 기존 보다 약 28% 정도 개선이 되었지만 **Oracle과 비교했을 때 여전히 3배 이상 느린 것**을 확인할 수 있었음. ▪ 이 부분에 대해서는 추가 개선이 가능한지 Hazelcast Korea 지원이 필요할 것으로 보임.

Data Hub IMDG 적용 PoC > 성능 검증

[Inline Processing]

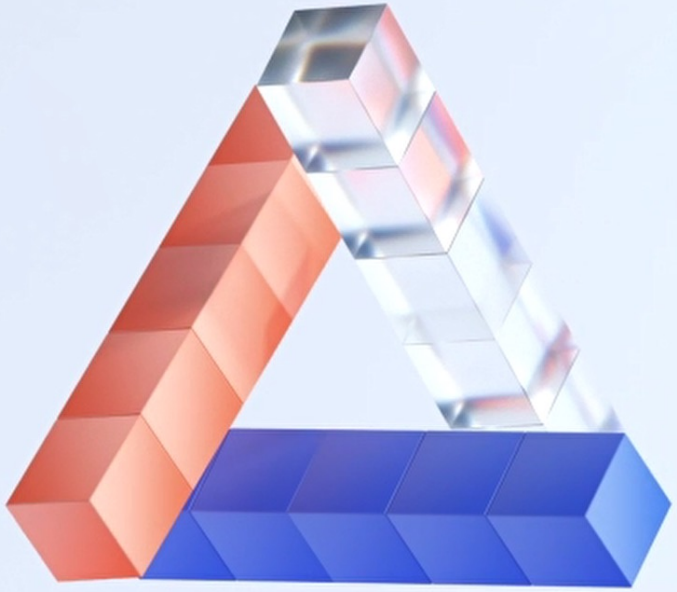


- Inline Process의 경우 Oracle 과 HazelCast 간 DB **Insert** 평균 처리 시간 비교 시 큰 차이가 없는 것으로 확인이 됨.



Thread 200개 부터는 **초당 44000건 정도의 처리 성능을 보여주고 그 이상 성능이 더 증가하진 않음.**

DEVOCEAN



Q&A