if(kakao)2021

Druid@Kakao Druid 도입 사례 및 Multi-Tenant 클러스터 소개

황보동규 Evan.Hwangbo 카카오

Copyright 2021. Kakao Corp. All rights reserved. Redistribution or public display is not permitted without written permission from Kakao.

Druid에 대한 기대 Use Case – 1 Use Case – 2 Use Case – 3 Use Case – 4

Multi-Tenant 클러스터 소개

Druid에 대한 기대

Use Case – 1

Use Case – 2

Use Case – 3

Use Case – 4

Multi-Tenant 클러스터 소개

systems. Some of Druid's key features are:

- fast scans and aggregations.
- sub-second to a few seconds.
- available for querying) or in batches.
- any reason, including configuration changes and software updates.
- recovers.
- indexes that power fast filtering and searching across multiple columns.
- the query. This leads to significant performance improvements for time-based data.
- performance boosts.

1. Columnar storage format. Druid uses column-oriented storage, meaning it only needs to load the exact columns needed for a particular query. This gives a huge speed boost to queries that only hit a few columns. In addition, each column is stored optimized for its particular data type, which supports

2. Scalable distributed system. Druid is typically deployed in clusters of tens to hundreds of servers, and can offer ingest rates of millions of records/sec, retention of trillions of records, and query latencies of

3. Massively parallel processing. Druid can process a query in parallel across the entire cluster. 4. Realtime or batch ingestion. Druid can ingest data either real-time (ingested data is immediately

5. Self-healing, self-balancing, easy to operate. As an operator, to scale the cluster out or in, simply add or remove servers and the cluster will rebalance itself automatically, in the background, without any downtime. If any Druid servers fail, the system will automatically route around the damage until those servers can be replaced. Druid is designed to run 24/7 with no need for planned downtimes for

6. Cloud-native, fault-tolerant architecture that won't lose data. Once Druid has ingested your data, a copy is stored safely in deep storage (typically cloud storage, HDFS, or a shared filesystem). Your data can be recovered from deep storage even if every single Druid server fails. For more limited failures affecting just a few Druid servers, replication ensures that queries are still possible while the system

7. Indexes for quick filtering. Druid uses Roaring or CONCISE compressed bitmap indexes to create

8. Time-based partitioning. Druid first partitions data by time, and can additionally partition based on other fields. This means time-based queries will only access the partitions that match the time range of

9. Approximate algorithms. Druid includes algorithms for approximate count-distinct, approximate ranking, and computation of approximate histograms and quantiles. These algorithms offer bounded memory usage and are often substantially faster than exact computations. For situations where accuracy is more important than speed, Druid also offers exact count-distinct and exact ranking.

10. Automatic summarization at ingest time. Druid optionally supports data summarization at ingestion time. This summarization partially pre-aggregates your data, and can lead to big costs savings and

Druid's core architecture combines ideas from data warehouses, timeseries databases, and logsearch systems. Some of Druid's key features are:

- fast scans and aggregations.
- sub-second to a few seconds.
- available for querying) or in batches.

those servers can be re any reason, including of 6. Cloud-native, fault-to

copy is stored safely in a can be recovered from deep storage even if every single Druid server fails. For more limited failures affecting just a few Druid servers, replication ensures that queries are still possible while the system recovers.

- indexes that power fast filtering and searching across multiple columns.
- the query. This leads to significant performance improvements for time-based data.
- performance boosts.

1. Columnar storage format. Druid uses column-oriented storage, meaning it only needs to load the exact columns needed for a particular query. This gives a huge speed boost to queries that only hit a few columns. In addition, each column is stored optimized for its particular data type, which supports

2. Scalable distributed system. Druid is typically deployed in clusters of tens to hundreds of servers, and can offer ingest rates of millions of records/sec, retention of trillions of records, and query latencies of

3. Massively parallel processing. Druid can process a query in parallel across the entire cluster. 4. Realtime or batch ingestion. Druid can ingest data either real-time (ingested data is immediately

5. Self-healing, self-balancing, easy to operate. As an operator, to scale the cluster out or in, simply add or remove servers and the cluster will rebalance itself automatically, in the background, without any downtime. If any Druid servers fail, the system will automatically route around the damage until





as ingested your data, a red filesystem). Your data

7. Indexes for quick filtering. Druid uses Roaring or CONCISE compressed bitmap indexes to create

8. Time-based partitioning. Druid first partitions data by time, and can additionally partition based on other fields. This means time-based queries will only access the partitions that match the time range of

9. Approximate algorithms. Druid includes algorithms for approximate count-distinct, approximate ranking, and computation of approximate histograms and quantiles. These algorithms offer bounded memory usage and are often substantially faster than exact computations. For situations where accuracy is more important than speed, Druid also offers exact count-distinct and exact ranking.

10. Automatic summarization at ingest time. Druid optionally supports data summarization at ingestion time. This summarization partially pre-aggregates your data, and can lead to big costs savings and

名的名

Druid에 대한 기대

- Use Case 1
- Use Case 2
- Use Case 3
- Use Case 4
- Multi-Tenant 클러스터 소개

- 개인화 광고 타게팅





- 콘텐츠 관리자 통계 페이지 제공



순위	2		로:	2
1	m.search.daum.net	258	D	[넷플릭스]https://search.daum.net/search?w=tot&DA=YZR&t_r
2	m.search.naver.com	179	G	[넷플릭스 1년간의 기록]https://www.google.com/search?q=넷플릭.
3	search.daum.net	123	f	https://www.facebook.com/visualworks0199?fref=nf&_tn_=%2C
4	www.naver.com	75	D	[그라비티커피 판쿄]https://search.daum.net/search?nil_suggest=btn
5	www.tistory.com	58	D	[샤오미제품 후기]https://search.daum.net/search?nil_suggest=btn
6	www.daum.net	46	N	[샤오미제품]https://search.naver.com/search.naver?sm=tab_hty.top.
7	www.google.com	32	Ŧ	https://www.tistory.com/category/life
8	www.instagram.com	27	N	[샤오미제품]https://search.naver.com/search.naver?sm=tab_hty.top.
9	www.youtube.com	19	6	http://cafe.daum.net/50in50/1pRl/1134525?q=%B3%D8%C7%C3
10	cafe.daum.net	17	Ŧ	https://www.tistory.com/category/life
11	story.kakao.com	14	y	https://twitter.com/hashtag/%EC%B9%B4%EC%B9%B4%EC%9
12	twitter.com	6	Ð	https://story.kakao.com/_1XJCz2
13	cafe.naver.com	5	0	https://www.instagram.com/explore/tags/%EC%83%A4%EC%98
			D	[넷플릭스]https://search.daum.net/search?w=tot&DA=YZR&tnil
			N	[넷플릭스 기록]https://search.naver.com/search.naver?sm=tab_hty
			N	[넷플릭스]https://search.naver.com/search.naver?sm=tab_hty.top
			D	[그라비티커피 판쿄]https://search.daum.net/search?nil_suggest=btn
			D	[넷플릭스]https://search.daum.net/search?w=tot&DA=YZR&tnil
			N	[샤오미제품]https://search.naver.com/search.naver?sm=tab_hty.top

2)

~		
0:22		
0:22		
0:20		
0:20		
0:18		
0:17		
0:16		
0:16		
0:16		
0:09		
80:08 0:08		
0:08		
0:07		
0:07		
0:06		

- 사내 분석 용도 - 사용자 데이터의 실시간 다차원 분석





– 전사 서버 지표 실시간 수집 / 모니터링



Druid에 대한 기대 Use Case — 1 Use Case — 2 Use Case — 3

Use Case – 4

Multi-Tenant 클러스터 소개

CPU Servers

0.10.1/0.21.0120+Druid Version Datasources

160bil+ 15mil+

Daily Write



Daily Request



Master Servers

Overlord

Coordinator



Master Servers

Overlord

Coordinator













z ec autob digi up action — biz z ec autob digi up equest 2 —		pheus ten minute utobial gi upi equest 1 ccautobial gi upi equest 5
02 D0 04 D0 06 D0		
	Query Faled	
e	Metro	Value
1 08 24 14 58 00	khp service metrics 172 01 7 70 cnt	5 00
1 08 24 14 16 00	khp service metrics 172 01 7 70 cnt	5 00
1 00 24 12 21 00	khp service metrics 172 26 106 200 cnt	7 00
1 00 24 11 52 00	khp service metrics 172 01 7 70 cnt	14 00
1 00 24 11 52 00	khp hest metrics 172 01 7 70 cnt	7 00
1 00 24 11 50 00	khp service metrics 172 01 7 70 cnt	6 00
1 00 24 11 50 00	khp hest metrics 172 01 7 70 cnt	6 00
1 00 24 11 20 00	khp service metrics 172 01 7 70 cnt	9 00
1 00 24 10 54 00	khp service metrics 172 01 7 70 cnt	00 C

Query Count by DataSource



