Kafka + Druid = Immediate Intelligence

Apache Kafka® and Apache Druid® as the platform for modern analytics
Kafka + Druid = Immediate Intelligence

Apache Druid
  Use Cases
  Architecture
  What Makes it Fast
Kafka Integration
  Streaming at any Scale
  Data Enhancement in the Pipeline
Learn more...
Kafka + Druid = Immediate Intelligence

Apache Druid
  Use Cases
  Architecture
  What Makes it Fast
Kafka Integration
  Streaming at any Scale
  Data Enhancement in the Pipeline
Learn more...
High performance.
Low-latency, distributed query execution and high throughput ingestion

Real-time.
Event data (clickstream, network flows, user behavior, programmatic advertising, server metrics, IoT…)

Analytics.
Counting, ranking, statistics…

Database.
Highly-available, time-sharded, partitioned, columnar, indexed, compressed, versioned materialized view
The adoption and maturity of our technology

- 10,000+ Community Members
- 400+ Active Contributors
- 450+ Code Releases
- 1,000+ Companies Using Druid
Modern Analytics Applications

- 0.1–3s query
- fresh data
- high concurrency
- highly interactive
Where Immediate Intelligence matters...

- **Risk/fraud**
- **Digital ads**
- **Data-driven applications**
- **Data warehousing**
- **Network performance**
- **Metrics**
- **Clickstreams**
- **IoT**

Exploring the data in lots of ways

- Lots of users using it concurrently
- High volumes of fresh data to act quickly
- Time comparisons, and operations on high cardinality data
Architecture - Druid uses a microservice architecture

Master server
- Coordinator
- Apache ZooKeeper

Data server
- Historical
- Indexer
- Historical
- Indexer

Deep storage

Query server

Broker

Scalable & Resilient
- Ingestion
- Data Management
- Query

Streaming data
Batch data
What Makes it Fast

- Compression
- Secondary indexes
- Operate on compressed data
- Late materialization

DATA

INDEX
[0,1,2](11100000)
[3,4](00011000)
[5,6,7](0000111)

DICT
Melbourne = 0
Perth = 1
Sydney = 2
Engine and data format are tightly integrated

<table>
<thead>
<tr>
<th>__time</th>
<th>artist (STRING)</th>
<th>city (STRING)</th>
<th>price (LONG)</th>
<th>count (LONG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1293840000000</td>
<td>0</td>
<td>AC/DC = 0</td>
<td>1800</td>
<td>25</td>
</tr>
<tr>
<td>1293840000000</td>
<td>0</td>
<td>Kylie = 1</td>
<td>2912</td>
<td>42</td>
</tr>
<tr>
<td>1293840000000</td>
<td>1</td>
<td>Silverchair = 2</td>
<td>1953</td>
<td>17</td>
</tr>
<tr>
<td>1293840000000</td>
<td>2</td>
<td>Melbourne = 0</td>
<td>3194</td>
<td>170</td>
</tr>
<tr>
<td>1293840000000</td>
<td>2</td>
<td>Perth = 1</td>
<td>5690</td>
<td>112</td>
</tr>
<tr>
<td>1293840000000</td>
<td>2</td>
<td>Sydney = 2</td>
<td>1100</td>
<td>67</td>
</tr>
<tr>
<td>1293840000000</td>
<td>2</td>
<td></td>
<td>8423</td>
<td>53</td>
</tr>
<tr>
<td>1293840000000</td>
<td>2</td>
<td></td>
<td>9080</td>
<td>94</td>
</tr>
</tbody>
</table>

Dictionary encoded (sorted)

Bitmap index (stored compressed)
Engine and data format are tightly integrated

<table>
<thead>
<tr>
<th>_time (LONG)</th>
<th>artist (STRING)</th>
<th>city (STRING)</th>
<th>price (LONG)</th>
<th>count (LONG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1293840000000</td>
<td>0,1,2(111000000)</td>
<td>0,1,2(101000000)</td>
<td>1800</td>
<td>25</td>
</tr>
<tr>
<td>1293840000000</td>
<td><a href="00011000">3,4</a></td>
<td><a href="01011000">1,3,4</a></td>
<td>2912</td>
<td>42</td>
</tr>
<tr>
<td>1293840000000</td>
<td><a href="0000111">5,6,7</a></td>
<td><a href="00000111">5,6,7</a></td>
<td>1953</td>
<td>17</td>
</tr>
<tr>
<td>1293840000000</td>
<td><a href="111000000">0,2</a></td>
<td><a href="101000000">0,2</a></td>
<td>3194</td>
<td>170</td>
</tr>
<tr>
<td>1293840000000</td>
<td><a href="00011000">3,4</a></td>
<td><a href="01011000">1,3,4</a></td>
<td>5690</td>
<td>112</td>
</tr>
<tr>
<td>1293840000000</td>
<td><a href="0000111">5,6,7</a></td>
<td><a href="00000111">5,6,7</a></td>
<td>1100</td>
<td>67</td>
</tr>
<tr>
<td>1293840000000</td>
<td>0,1,2(111000000)</td>
<td>0,1,2(101000000)</td>
<td>8423</td>
<td>53</td>
</tr>
<tr>
<td>1293840000000</td>
<td><a href="00011000">3,4</a></td>
<td><a href="01011000">1,3,4</a></td>
<td>9080</td>
<td>94</td>
</tr>
</tbody>
</table>

```
SELECT city, SUM(price) FROM sales
WHERE artist = 'AC/DC'
GROUP BY city
```

Dictionary encoded (sorted)
Bitmap index (stored compressed)
Engine and data format are tightly integrated

SELECT city, SUM(price) 
FROM sales 
WHERE artist = 'AC/DC' 
GROUP BY city

<table>
<thead>
<tr>
<th>city (STRING)</th>
<th>price (LONG)</th>
<th>count (LONG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melbourne</td>
<td>1800</td>
<td>25</td>
</tr>
<tr>
<td>Perth</td>
<td>2912</td>
<td>42</td>
</tr>
<tr>
<td>Sydney</td>
<td>1953</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>3194</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>5690</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>1100</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>8423</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>9080</td>
<td>94</td>
</tr>
</tbody>
</table>
Engine and data format are tightly integrated

```
SELECT city, SUM(price)
FROM sales
WHERE artist = 'AC/DC'
GROUP BY city
```
Engine and data format are tightly integrated

```
SELECT city, SUM(price) AS total
FROM sales
WHERE artist = 'AC/DC'
GROUP BY city

<table>
<thead>
<tr>
<th>City</th>
<th>Price</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melbourne</td>
<td>1800</td>
<td>25</td>
</tr>
<tr>
<td>Perth</td>
<td>2912</td>
<td>42</td>
</tr>
<tr>
<td>Sydney</td>
<td>1953</td>
<td>17</td>
</tr>
<tr>
<td>Melbourne</td>
<td>3194</td>
<td>170</td>
</tr>
<tr>
<td>Perth</td>
<td>5690</td>
<td>67</td>
</tr>
<tr>
<td>Sydney</td>
<td>1100</td>
<td>112</td>
</tr>
<tr>
<td>Melbourne</td>
<td>8423</td>
<td>53</td>
</tr>
<tr>
<td>Perth</td>
<td>9080</td>
<td>94</td>
</tr>
</tbody>
</table>
```
Engine and data format are tightly integrated

```
SELECT city, SUM(price)
FROM sales
WHERE artist = 'AC/DC'
GROUP BY city
```

Dictionary encoded (sorted)

```
AC/DC = 0
Kylie = 1
Silverchair = 2
```

Bitmap index (stored compressed)

```
[0,1,2](11100000)
[3,4](00011000)
[5,6,7](0000111)
```

```plaintext
<table>
<thead>
<tr>
<th>city</th>
<th>price</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melbourne</td>
<td>1800</td>
<td>25</td>
</tr>
<tr>
<td>Perth</td>
<td>2912</td>
<td>42</td>
</tr>
<tr>
<td>Sydney</td>
<td>1953</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>3194</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>5690</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>1100</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>8423</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>9080</td>
<td>94</td>
</tr>
</tbody>
</table>
```
Engine and data format are tightly integrated

```
SELECT city, SUM(price) 
FROM sales 
WHERE artist = 'AC/DC' 
GROUP BY city
```
Druid uses a microservice architecture

- Coordinator
- Apache ZooKeeper
- Master server
- Data server
- Historical
- Indexer
- Deep storage
- Streaming data
- Batch data
- Broker
- Query server
Kafka + Druid = Immediate Intelligence

Apache Druid
  Use Cases
  Architecture
  What Makes it Fast

**Kafka Integration**
  Streaming at any Scale
  Data Enhancement in the Pipeline

Learn more...
Scalable Data Ingestion

Producers
- Producer 1
- Producer 2
- Producer M

Apache Kafka
- Broker 1
  - Partition 1
  - Partition 2
- Broker 2
  - Partition 3
  - Partition 4
- Broker Nk
  - Partition P-1
  - Partition P

Apache Druid Ingestion Tasks
- Indexer 1
  - Task 1
  - Task 2
- Indexer 2
  - Task 3
  - Task 4
- Indexer Nd
  - Task T-1
  - Task T

Deep Storage

P >= T
Scalable Real-time Queries

Producers
- Producer 1
- Producer 2
- Producer Np

Apache Kafka
- Broker 1
  - Partition 1
  - Partition 2
- Broker 2
  - Partition 3
  - Partition 4
- Broker Nk
  - Partition P-1
  - Partition P

Apache Druid Ingestion Tasks
- Indexer 1
  - Task 1
  - Task 2
- Indexer 2
  - Task 3
  - Task 4
- Indexer Ni
  - Task T-1
  - Task T

Apache Druid Query Nodes
- Druid Broker 1
- Druid Broker 2
- Druid Broker Nb

Application Query
Scalable Real-time + Historical Queries

Producers
- Producer 1
- Producer 2
- Producer Np

Apache Kafka
- Broker 1
  - Partition 1
  - Partition 2
- Broker 2
  - Partition 3
  - Partition 4
- Broker Nk
  - Partition P-1
  - Partition P

Apache Druid Ingestion Tasks
- Indexer 1
  - Task 1
  - Task 2
- Indexer 2
  - Task 3
  - Task 4
- Indexer Ni
  - Task T-1
  - Task T

Apache Druid Query Nodes
- Druid Broker 1
- Druid Broker 2
- Druid Broker Nb
- Historical 1
- Historical 2
- Historical M

Application Query
Best Practice: Scalable Data Ingestion + Autocompaction

Apache Kafka
- Broker 1
  - Partition 1
  - Partition 2
- Broker 2
  - Partition 3
  - Partition 4
- Broker Nk
  - Partition P-1
  - Partition P

Apache Druid Ingestion Tasks
- Indexer 1
  - Task 1
  - Task 2
- Indexer 2
  - Task 3
  - Task 4
- Indexer Nd
  - Task T-1
  - Task T

Completed Segments
- 9:15 - 9:30
- 9:00 - 9:15

Compact
- C segments
  - C << T

\( P \geq T \)
Data Enhancement in the Pipeline - Kafka SQL

Producers

A messages

B messages

Apache Kafka

Topic A

Topic B

Topic C

Pipeline Transformations

KSQL

Enhanced messages A join B => C

Apache Druid Ingestion
Data Enhancement in the Pipeline - Other Tech

Producers
- A messages
- B messages

Apache Kafka
- Topic A
- Topic B
- Topic C

Pipeline Transformations

Apache Druid Ingestion

Enhanced messages A join B => C
Kafka + Druid = Immediate Intelligence

Apache Druid
  Use Cases
  Architecture
  What Makes it Fast
Kafka Integration
  Streaming at any Scale
  Data Enhancement in the Pipeline

Learn more...
There are numerous companies of various sizes in production with Druid. Some of them are listed below.

**Adikteev**

*Adikteev* is the leading mobile app re-engagement platform for performance-driven marketers, and is consistently ranked in the top 5 of the AppsFlyer Performance Index. By using Druid instead of relying on slow and stale dashboards, we have been able to achieve internal productivity gains, make better decisions faster, provide our external clients with strategic advice to improve the performance and effectiveness of their retargeting marketing campaigns, and notify clients quickly of potentially serious problems.

- How Adikteev helps customers succeed using self-service analytics

**Airbnb**

Druid powers slice and dice analytics on both historical and real-time data. It significantly reduces latency of analytic queries and helps people to get insights more interactively.

- How Druid enables analytics at Airbnb
- How Airbnb Achieved Metric Consistency at Scale

**Airbridge**

*Airbridge* is an people-based attribution and omni-channel campaign analytics platform helping marketers measure and optimize their marketing campaigns. Thanks to Druid's data aggregation technology, marketers using Airbridge are able to receive real-time granular reports regarding their campaign's performance executed across a variety of devices, platforms, and channels.

**Alibaba**

Alibaba
Druid Forum “Tag Browser”

[Link to Druid Forum Tag Browser](https://www.druidforum.org/tags)
Overview

A fast, modern analytics database
Druid is designed for workflows where fast ad-hoc analytics, instant data visibility, or supporting high concurrency is important. As such, Druid is often used to power UIs where an interactive, consistent user experience is desired.

Easy integration with your existing data pipelines
Druid streams data from message buses such as Kafka, and Amazon Kinesis, and batch load files from data lakes such as HDFS, and Amazon S3. Druid supports most popular file formats for structured and semi-structured data.

Fast, consistent queries at high concurrency
Druid has been benchmarked to greatly outperform legacy solutions. Druid combines novel storage ideas, indexing structures, and both exact and approximate queries to return most results in under a second.

Broad applicability

Upcoming Events

Featured Content

https://druid.apache.org/
<table>
<thead>
<tr>
<th>#</th>
<th>Title</th>
<th>Description</th>
<th>Status</th>
<th>Area</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>11849</td>
<td>warn when segment cannot be loaded by Historical nodes</td>
<td>Area - Segment Balancing/Coordination</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11848</td>
<td>[WIP] Add support for multi dimension range partitioning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11847</td>
<td>Add info for compaiton config dialog</td>
<td>Area - Web Console</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11846</td>
<td>Missing &quot;Loader=yaml.FullLoader&quot; parameter in generate-binary-license and generate-binary-notice py scripts</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11843</td>
<td>Use a simple class to sanitize JDBC exceptions and also log them</td>
<td>Area - JDBC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11839</td>
<td>fixes validation error in helm chart templates license</td>
<td>Helmet Chart</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11833</td>
<td>Optimize Method VersionedIntervalTimeline.isOvershadowed from O(N) to O(logN).</td>
<td>Performance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11828</td>
<td>Refactor ResponseContext</td>
<td>Area - Querying</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11825</td>
<td>Extract HyperLogLog into an interface</td>
<td>Area - Querying</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Imply blog “Community Spotlights” and “Discovery Series”
Apache Druid® Basics

Druid is becoming the go-to cloud-native answer to scalable time-series data storage and analytics. So, if you have time-series data, you'll want to know how to use Druid.

FREE  6 hr 30 min

Apache Druid® Ingestion and Data Modeling

Data modeling is the key to leveraging your Apache Druid® database. Learn how to ingest data into Druid data models that are fast and scalable.

FREE

Beginner and Intermediate level lessons with accreditation exams

learn.imply.io
Thank you

Apache Druid training and labs
learn.imply.io

Q&A, video library, and events
www.druidforum.org
Thank you!
Backup Slides
The hot / immediate pipeline!

- **Data lake**: Scale out
- **Query engine**: 0.1–3s query, fresh data, high concurrency, highly interactive (OLAP)
- **Kafka**: Sharded
- **Spark**: Pure storage
- **OLTP RDBMS**: Sharded
- **Druid**: Optimized copy

**IoT**
Druid: A real-time analytical data store

http://static.druid.io/docs/druid.pdf