



# Apache Druid 0.22 release & Roadmap

November 2021

# About me

- Jihoon Son
- PMC member & committer @ Apache Druid
- Software engineer @ Imply

# Agenda

- Druid 0.22.0 release
- What will be in the next release?
- Bonus: project Shapeshift

Druid 0.22.0 release

# druid-0.22.0

Latest

Compare



 clintropolis released this Sep 22, 2021 · 143 commits to master since this release  druid-0.22.0  cc603d6

Apache Druid 0.22.0 contains over 400 new features, bug fixes, performance enhancements, documentation improvements, and additional test coverage from 73 contributors. [See the complete set of changes for additional details.](#)

Download: <https://druid.apache.org/downloads.html>

- ARRAY\_AGG and STRING\_AGG
- Bitwise math expressions
- Native expression aggregator
- Human-readable number format functions
- Protobuf JDBC
- Streaming ingestion: task auto scaling
- Native batch: use deep storage for shuffle
- Native batch: reduced memory usage
- Improved Confluent Schema Registry support
- ... and hundreds more!

# ARRAY\_AGG

```
1 SELECT ARRAY_AGG(DISTINCT cityName, 100000) FROM wikipedia
```

▶ Run ...  Auto limit Live query: Off 1 result in 0.41s ↓

EXPR\$0

[null,"Belgrade","Scarborough","Carrollton","Narni Scalo","Berkeley","Austin","Indore","Erbusco","Seven Hills","Auburn Hills","Jerez de la Front

# STRING\_AGG

```
1 SELECT STRING_AGG(DISTINCT cityName, '-', 100000) FROM wikipedia
```

▶ Run ...  Auto limit Live query: Off 1 result in 0.09s ↓

EXPR\$0

Belgrade-Scarborough-Carrollton-Narni Scalo-Berkeley-Austin-Indore-Erbusco-Seven Hills-Auburn Hills-Jerez de la Frontera-Carimate-Belo Ho

# Bitwise math expressions

```
1 SELECT
2   added,
3   deleted,
4   delta,
5   BITWISE_AND(added,delta),
6   BITWISE_OR(added,delta),
7   BITWISE_XOR(added,delta),
8   BITWISE_COMPLEMENT(added),
9   BITWISE_SHIFT_LEFT(added, 2),
10  BITWISE_SHIFT_RIGHT(delta, 2),
11  BITWISE_CONVERT_LONG_BITS_TO_DOUBLE(added),
12  BITWISE_
```

▶ Run

<b>BITWISE_AND</b>	function
<b>BITWISE_COMPLEMENT</b>	function
<b>BITWISE_CONVERT_DOUBLE_TO_LONG_BITS</b>	
<b>BITWISE_CONVERT_LONG_BITS_TO_DOUBLE</b>	
<b>BITWISE_OR</b>	function
<b>BITWISE_SHIFT_LEFT</b>	function
<b>BITWISE_SHIFT_RIGHT</b>	function
<b>BITWISE_XOR</b>	function

## BITWISE\_CONVERT\_DOUBLE\_TO\_LONG\_BITS

### Syntax:

BITWISE\_CONVERT\_DOUBLE\_TO\_LONG\_BITS(expr)

### Description:

Converts the bits of an IEEE 754 floating-point double value to a long. If the input is not a double, it is implicitly cast to a double prior to conversion



# Native expression aggregator

"count" aggregator

```
{
  "type": "expression",
  "name": "expression_count",
  "fields": [],
  "initialValue": "0",
  "fold": "__acc + 1",
  "combine": "__acc + expression_count"
}
```

"sum" aggregator

```
{
  "type": "expression",
  "name": "expression_sum",
  "fields": ["column_a"],
  "initialValue": "0",
  "fold": "__acc + column_a"
}
```

# Streaming ingestion: task auto scaling

- Auto-scale task counts per lag in streaming ingestion

```
"ioConfig": {  
  "topic": "your_topic",  
  ...  
  "autoScalerConfig": {  
    "enableTaskAutoScaler": true,  
    "taskCountMax": 6,  
    "taskCountMin": 2  
  }  
}
```

# Native batch: reduced memory usage

- Reduce memory footprint of native batch tasks by tracking less data for in-progress segments
- **`druid.indexer.task.batchProcessingMode = OPEN_SEGMENTS`**
  - Old default. Tracks all metadata and mmaps all in-progress segments.
- **`druid.indexer.task.batchProcessingMode = CLOSED_SEGMENTS`**
  - New default. Tracks all metadata but does not mmap in-progress segments.
- **`druid.indexer.task.batchProcessingMode = CLOSED_SEGMENTS_SINKS`**
  - Most aggressive mode. Neither tracks all metadata nor mmaps in-progress segments. Experimental for now.

What will be in the next release?

# Complex dimensions

druid	druid	wikipedia	__time	0	<i>null</i>	NO	TIMESTAMP	<i>null</i>	<i>null</i>	<i>null</i>
druid	druid	wikipedia	thetaUser	25	<i>null</i>	YES	COMPLEX<thetaSketch>	<i>null</i>	<i>null</i>	<i>null</i>
druid	druid	wikipedia	distinctUsers	8	<i>null</i>	YES	COMPLEX<HLLSketch>	<i>null</i>	<i>null</i>	<i>null</i>
druid	druid	wikipedia	count	4	<i>null</i>	NO	BIGINT	<i>null</i>	<i>null</i>	19
druid	druid	wikipedia	sum_added	20	<i>null</i>	NO	BIGINT	<i>null</i>	<i>null</i>	19

- Support for user-defined complex column type
- Expression support
- TODOs
  - Bitmap index support
  - Grouping key support

# More vectorization

- Vectorize logical operators and boolean functions

```
// 30: logical and operator
"SELECT CAST(long1 as BOOLEAN) AND CAST (long2 as BOOLEAN), COUNT(*) FROM foo GROUP BY 1 ORDER BY 2",
// 31: isnull, notnull
"SELECT long5 IS NULL, long3 IS NOT NULL, count(*) FROM foo GROUP BY 1,2 ORDER BY 3"
```

Benchmark	(query)	(rowsPerSegment)	(vectorize)	Mode	Cnt	Score	Error	Units
SqlExpressionBenchmark.querySql	30	5000000	false	avgt	5	761.667 ± 35.745	ms/op	
SqlExpressionBenchmark.querySql	30	5000000	force	avgt	5	152.102 ± 9.390	ms/op	
SqlExpressionBenchmark.querySql	31	5000000	false	avgt	5	431.104 ± 32.951	ms/op	
SqlExpressionBenchmark.querySql	31	5000000	force	avgt	5	100.884 ± 8.919	ms/op	

# Native batch: multi-dimension range partitioning

- Range partitioning support with multiple partition dimensions

```
{  
  "type": "range",  
  "partitionDimensions": ["city", "device"]  
}
```

# Project Shapeshift



## Druid Summit 2021 Recordings



### Keynote - Building Modern Analytics Applications with Apache Druid

Fangjin Yang - Implied - Co-Founder and Chief Executive Officer



### Keynote - Fireside Chat with the Original Creators of Apache Druid

Gian Merlino - Implied - Co-Founder and Chief Technology Officer  
Vadim Ogievetsky - Implied - Co-Founder and Chief Experience Officer  
Fangjin Yang - Implied - Co-Founder and Chief Executive Officer  
Eric Tschetter - Implied - Field CTO



### Evolution to Data Mesh

druid SUMMIT  
THE VIRTUAL CONFERENCE BY imply



### Unveiling What's Next for Apache Druid and Implied

Recordings: <https://druidsummit.org/past-event-recordings/>

# Project Shapeshift

Full video: [https://www.youtube.com/watch?v=uogaBfCT\\_oQ](https://www.youtube.com/watch?v=uogaBfCT_oQ)

- Cloud-native
  - Auto-tuning, auto-scaling, push-based ingestion
- Simple
  - Easy-to-use, ingestion with SQL
- Complete
  - More use cases for reports and alerts

# Cloud-native

- Imply SaaS: Druid as a Service
  - Serverless
  - Auto-scaling
  - Pay-as-you-go

# Simple

- What if everything was SQL?
  - Batch ingestion with SQL: <https://github.com/apache/druid/issues/11929>

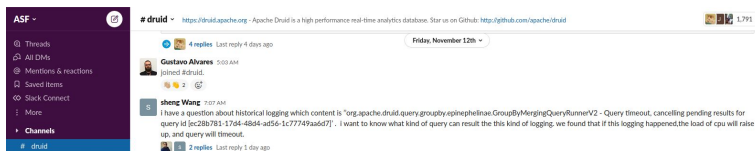
```
INSERT INTO tbl
SELECT
  TIME_PARSE("timestamp") AS __time,
  channel,
  cityName,
  countryName
FROM TABLE(
  EXTERN(
    '{"type": "s3", "uris": ["s3://bucket/file"]}',
    '{"type": "json"}',
    '[{"name": "channel", "type": "string"}, {"name": "cityName", "type": "string"}, {"name": "countryName", "type":
"string"}, {"name": "timestamp", "type": "string"}]'
  )
)
BUCKET BY FLOOR(__time TO DAY)
ORDER BY channel
```

# Complete

- Expanding Druid use cases
  - Reports and alerts
  - Better support for long-running queries
  - Asynchronous APIs
  - And more!

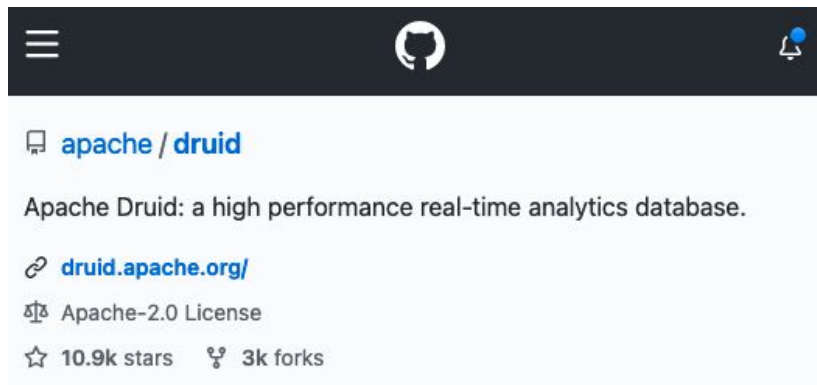
# Join in

Follow the Druid project on Twitter!



Join the community!

<https://druid.apache.org/community/>



Any questions?



# Thank you for attending

November 2021