PostgreSQL as a Big Data Platform

Chris Travers

May 10, 2019

About Adjust

Adjust provides mobile advertisement attribution and analytics on mobile application events. On average we track advertisement performance for 10 applications for each smart phone in use. We focus on fairness and transparency.

About Me

- Long time software developoer
- New Contributor to PostgreSQL
- Working with PostgreSQL since 1999
- Head of PostgreSQL team at Adjust GmbH

My Team

- Research and Development
- Database environments supporting diverse products
- PB scale deployments

Assumptions in Relational Model

- Optimized for mathematical manipulations via Set Theory
- Real implementations fall short (bags vs sets etc)
- Data modelled as tuples (ideally short) where a tuple element is assumed to be atomic.
- Think accounting or order management software.

Traditional business intelligence (relational)

- Same approach mathematically as OLTP
- Typically data is historical meaning that it can be inserted periodically.
- Sales over time per country is a standard example.
- PostgreSQL struggles a bit with BI due to table structure.

Industry Trends

- Wider variety of source data for analysis (Variety)
- Real-time analytics on streams of events (Velocity)
- More data than can be managed gracefully on one machine (Volume)

Collectively these are known as "Big Data" and might include analysis of published research, facebook posts, internet-of-things events, or MMO game data.

What Big Data Is/Is Not

Big Data Is:

- About V3 Problems
- A set of techniques
- About Attention to Detail

Big Data Is Not

- A set of products
- A set of technologies
- About following recipes.

At Adjust we apply big data techniques to large, high velocity data sets using vanilla Postgres and a lot of our custom software.

Introducing our KPI Service Pipeline

- Environment Approaching 1PB
- Delivers near-realtime analytics on user behavior
- 100-300k requests a second
- Delivers to dashboard and external API users
- Different pieces have different availability considerations

Big Data Characteristics

- High Volume and Velocity
- High availability requirements for Ingestion
- Distributed data warehouse queries
- Data has has very large clusters of values, making ordinary sharding difficult.

Engineering Approach

- Pipeline of Data
- Highly redundant initial processing nodes
- Modestly redundant customer-facing shards
- Data moves through a pipeline.

Architecture

- Initial processing systems log their results
- MapReduce to customer-facing shard databases
- MapReduce again in delivering data to client
- Covered in "PostgreSQL At 20TB and Beyond"

PostgreSQL Challenges

- PostgreSQL FDW too latency sensitive to use between datacenters.
- Multiple inheritance used for some advanced features makes data schema changes difficult.
- Our shards' WAL traffic is measured in the TB/day.

Introducing Bagger

- Elastic Search Replacement
- High velocity ingestion (1M+ data points/sec)
- Very high volume (10PB)
- Free form data (JSON documents)
- Retention for Limited Time

Big Data Characteristics

- Very high velocity (up to 1M items per second ingestion)
- Very high volume (10PB of data, capped by quantity currently).
- Could include all kinds of new data at any time, so must handle variety of semi-structured data quite gracefully.

Engineering Approach

- Optimize for bulk storage and linear writes
- Use PostgreSQL JSONB and similar indexes
- Data partitioned by hour and dropped when disks are near full.
- Client-side sharding, so dedicated client

Architecture

- Data arrives by Kafka, partitioned for the dbs
- Data partitioned by query pattern and hour
- Partitions tracked on master databases
- Client written in Perl, which queries appropriate partitions and concatenates data

PostgreSQL Challenges

- Marshalling JSONB can be expensive
- System catalogs on ZFS on spinning disk are slow
- Requires significant custom C code in triggers to keep the system fast.
- Exception handling in PostgreSQL has been a source of bugs in the past.

Introducing Audience Builder

- Retargetting platform (describe use case)
- Only 12TB but expect to grow
- Non-typical query and access patterns
- Feature requests that could push this into PB range

Big Data Characteristics

- High enough velocity that saturation of NVME storage is a concern.
- Queries touch very large amounts of data
- Expect these issues to become far worse.

Engineering Approach

- Separation of storage and query
- Settled on Parquet as storage format
- Columnar data storage useful but most software does not support our access patterns well.
- Evaluated a few of alternatives to PostgreSQL here.
- Prioritized predictability and extensibility over peak performance

Architecture

- Parquet files on CephFS
- PostgreSQL as query engine only
- Wrote Parquet FDW for PostgreSQL
- With tuning and optimization, as fast as native files.
- Data arrives via Kafka and is written to Parquet files and these are registered with the database.
- Pluggable storage might be of interest here.

https://github.com/zilder/parquet_fdw

PostgreSQL Challenges

- Hundreds of thousands of tables
- Performance requires tables to be physically sorted
- Heavy reliance on streaming APIs (COPY)

Key Takeaways

- Big Data is about technique, not technology
- PostgreSQL is quite capable in this area
- Careful attention to requirements is important
- Every big data system is different.

Major Open Source Software We Use

- Apache Kafka
- PostgreSQL
- Redis
- Go
- Apache Flink
- CephFS
- Apache Spark
- Gentoo Linux

Thank You

Thanks for coming. Any questions?

chris.travers@adjust.com