The State of Postgres
For Modern, Scalable Applications

Umur Cubukcu | Citus Data | Strata Data Conference 2018

@umurc | @citusdata | citusdata.com
About me & Citus Data

- Umur Cubukcu, Co-Founder & CEO of Citus Data
- Citus: Distributed PostgreSQL
- Founded 2011, HQ in SOMA

@umurc | @citusdata
github.com/citusdata/citus
Databases used to be simple (2008)

- **Open Source**
- **Proprietary**

- **RDBMS**
- **Analytics (OLAP)**
- **Workloads**
- **Operations (OLTP)**
Two challenges for the relational database changed the landscape

1. Data with less structure

2. Data Growth >> Silicon Growth...

- Data: 2x every 15 mo
- Moore’s Law: 2x every 24 mo
So, how do I query the database?

It's not a database. It's a key-value store!

Ok, it's not a database. How do I query it?

You write a distributed map reduce function in Erlang!

Did you just tell me to go *** myself?

I believe I did, Bob.
Meanwhile: Short history of Postgres
Not the first time seeing similar challenges

1. **SQL or not? (1995)**
   - Post-Ingres
   - Started life as object store
   - Added SQL API in 1995

2. **Scaling out to handle data growth (2005)**
   - For analytics only: MPPs
   - So many forks! AsterData, Netezza,
     ParAccel (Redshift), Greenplum
Introducing PostgreSQL Extension APIs (2011)
Amplifying vs. breaking the ecosystem

CREATE EXTENSION ...
Addressing challenges to RDBMS

1. To structure, or **not** to structure?
2. Scaling out—compute & performance
Start from file system

(+) Any data, any structure
(+) ’Infinitely’ scalable storage
(+) Write fast

(-) Pay cost at query time
(-) Batch vs. real-time
(-) Indexes (Append only FS)
Worry about only one access pattern

Semi-structured (JSON) data

(+): Simple: Put & Get
(+): Scalable

(-): No expressiveness for analytics
(-): No JOINS, data duplication
(-): Enforce structure at app layer
(-): Multiple stores to integrate
 Extend the database for JSON data

Table "public.events"

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Sample Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_id</td>
<td>bigint</td>
<td>09288</td>
</tr>
<tr>
<td>created_at</td>
<td>timestamp</td>
<td>2018-03-08 00:57:12.6936+00</td>
</tr>
</tbody>
</table>
| payload  | jsonb      | {
|   "person":{
|     "name":"Emma",
|     "emails":{"west":"emma@sf.com", "east":"emma@nyc.com"},
|     "address":{
|       "street_number":12,
|       "street_name":"15th Street",
|       "city":"San Francisco",
|       "state":"CA",
|       "zip":94107
|     },
|     "phone_numbers":["6501234567","6500123456"]
|   },
|   "country":"USA",
|   "event_type":"push_notification",
|   "device_type":"mobile"
| }
Leverage indexing (and other fundamentals)

- B-tree indexes
- GIN & GiST indexes
- Secondary indexes
- Full text search
- Index-only scans
- Fitting indexes into memory

+ Not to forget: Parallel queries, MVCC, and many more.
Push computations (and joins) down to many PostgreSQL instances

```sql
SELECT FROM events a JOIN users b
```

Data Node 1
- `Events_101`
- `Users_101`
- `Events_103`
- `Users_103`

Data Node 2
- `Events_102`
- `Users_102`
- `Events_104`
- `Users_104`

Data Node N
Extending Postgres for horizontal scale: Citus
PostgreSQL: Vibrant, global ecosystem

Sample PostgreSQL Extensions

- citus
- pgcrypto
- pg_cron
- pg_partman
- postgresql-HLL
- cstore_fdw
- cube
- madlib
- dblink
- pg_trgm
- PostGIS
...

- pg_buffercache
- pg_prewarm
- btree_gin
- btree_gist
- pgaudit
- pg_stat_statements
- postgresql-unit
- plpgsql
- plv8
- pgmetrics
- foreign data wrappers
...

Integrations

- Tableau
- Informatica
- Spark
- Microsoft Azure
- AWS
- Google Cloud Platform
- Qlik
- Nagios
- MicroStrategy

Umur Cubukcu | Strata Data Conference | March 2018 | The State of Postgres
PostgreSQL on fire

Database adoption among developers

Source: % database job postings that mention each specific technology, across 20K+ job posts on Hacker News, https://news.ycombinator.com
Growing from already vast user base

PostgreSQL popularity = Hadoop + Mongo combined

Source: Google Trends for the past 2 years
So there’s an elephant in the room

How does it all fit in with your stack?
Modern workloads are evolving

- Improvement workloads
- Application workloads
  - Transactions
  - Short-requests
  - In-app analytics

Open Source

Proprietary

Analytics (OLAP)  Workloads  Operations (OLTP)
Modern databases serve 3 types of apps

1. **Systems of record**
   - Core workloads, transactions
   - Real-time data
   - Millisecond latencies

2. **Systems of engagement**
   - Drive engagement & revenue
   - Real-time data, multiple sources
   - Sub-second latencies

3. **Systems of improvement**
   - Identify business process improvements
   - Offline data, multiple sources
   - Sub-minute / hour latencies, data analysts

---

Umur Cubukcu | Strata Data Conference | March 2018 | The State of Postgres
PostgreSQL in your infrastructure stack

**Application**

- Standalone database
  - Storage
  - Compute
- Persistence layer for Kafka
- Persistence layer for Spark
- Adjacent to NoSQL

**PostgreSQL**

- Kafka
- Spark
- HDFS / S3

**Data**

Note: Standard PostgreSQL connectors for all tools (e.g. ODBC / JDBC, PostgreSQL language bindings) available for integrations.
Today one of our PostgreSQL tables surpassed half of trillion rows [over 54TB of data]. And still most of our queries run under 600ms
Parting thoughts: PostgreSQL becoming the Linux of Databases
Thank you

@umurc  @citusdata
github.com/citusdata/citus
citusdata.com/jobs
We’re Hiring!

Umur Cubukcu | Citus Data | Strata Data Conference 2018