Brytlyt is an ultra-high performance database that combines patent pending intellectual property and Graphics Processor Unit (GPU) hardware.

- **300 times** faster than existing database technology
- **Real-time analytics** replaces slow batch processing
- **Massively reducing cost** per query by a factor of 10 or more

Graphics Processor Units offer massive performance improvements over traditional computer hardware. However, working with GPUs is tough because everything has to be done in parallel. Brytlyt has patent pending IP to overcome this.
How are analysts spending their time?

64% of time is spent cleaning and organizing data

3 days per month is spent mining data for patterns or refining algorithms

37% of insight takes more than a week

SQL is the most common technology used ahead of Hadoop, Python and R
It’s possible that the reason data insights are challenging to obtain for the majority of respondents is that their analytics solutions simply do not have the speed, flexibility, or ease of use to answer the data questions people have.
More data means more opportunity
...and a greater challenge

✓ Real-time insights with real-time data
✓ No more overnight data queries
✓ Improved data governance and security

Business Leaders
Grappling with situational awareness on large amounts of high velocity data...

Analysts and Data Scientists
Struggling to interactively explore large datasets for detailed insights and answer business questions in real-time...

Because ultimately everything boils down to the performance that only a GPU database can deliver.

...requires a radical rethink of the status-quo
Usain Bolt’s reaction time: 0.155 seconds

Brytlyt can query a billion row dataset before Usain Bolt has even twitched a muscle.

But how fast is ultra-fast?

Faster than Usain Bolt’s reaction time!
Programming database operations for parallel execution is not trivial and requires a fundamentally new approach.
Brytlyt Indexes are advanced data structures, designed for GPU to improve performance

Query 1
SELECT cab_type, count(*)
FROM trips
GROUP BY cab_type;

Query 2
SELECT passenger_count, avg(total_amount)
FROM trips
GROUP BY passenger_count;

Indexed Queries

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brytlyt</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>MapD</td>
<td>10</td>
<td>50</td>
</tr>
</tbody>
</table>

time in milliseconds
Brytlyt is exceptionally fast, even without indexing

Query 3
```sql
SELECT passenger_count, 
extract(year from pickup_datetime), 
count(*)
FROM trips 
GROUP BY passenger_count, 
pickup_year;
```

Query 4
```sql
SELECT passenger_count, 
extract(year from pickup_datetime), 
cast(trip_distance as int), 
count(*)
FROM trips 
GROUP BY passenger_count, 
pickup_year, 
distance 
ORDER BY pickup_year, 
the_count desc;
```
SELECT cab_type,
       count(*)
FROM trips
GROUP BY cab_type;

SELECT cab_type,
       count(*)
FROM trips
GROUP BY cab_type,
         trip_type,
         passenger_count;
Ultra-Fast GPU JOIN
The Foundation of a Relational Database

part
- p_partkey INT (11)
- p_name VARCHAR (55)
- p_mfgr CHAR (25)
- p_brand CHAR (10)
- p_type VARCHAR (25)
- p_size INT (11)
- p_container CHAR (10)
- p_retailprice FLOAT
- p_comment VARCHAR (23)

orders
- o_orderkey INT (11)
- o_custkey INT (11)
- o_orderstatus CHAR (1)
- o_totalprice FLOAT
- o_orderdate DATE
- o_orderpriority CHAR (15)
- o_clerk CHAR (15)
- o_shippriority INT (11)
- o_comment VARCHAR (79)

customers
- c_custkey INT (11)
- c_name VARCHAR (25)
- c_address VARCHAR (40)
- c_nationkey INT (11)
- c_phone CHAR (15)
- c_acctbal FLOAT
- c_mktsegment CHAR (10)
- c_comment VARCHAR (117)

lineitem
- l_orderkey INT (11)
- l_partkey INT (11)
- l_suppkey INT (11)
- l_quantity FLOAT
- l_extendedprice FLOAT
- l_discount FLOAT
- l_tax FLOAT
- l_returnflag CHAR (1)
- l_linestatus CHAR (1)
- l_shipdate DATE
- l_commitdate DATE
- l_receiptdate DATE
- l_shipinstruct CHAR (25)
- l_shipmode CHAR (10)
- l_comment VARCHAR (199)

partsupp
- ps_partkey INT (11)
- ps_suppkey INT (11)
- ps_availqty INT (11)
- ps_supplycost FLOAT
- ps_comment VARCHAR (199)

supplier
- s_suppkey INT (11)
- s_name CHAR (25)
- s_address VARCHAR (40)
- s_nationkey INT (11)
- s_phone CHAR (15)
- s_acctbal FLOAT
- s_comment VARCHAR (101)
nation
- n_nationkey INT (11)
- n_name CHAR (25)
- n_regionkey INT (11)
- n_comment VARCHAR (152)

region
- r_regionkey INT (11)
- r_name CHAR (25)
- r_comment VARCHAR (152)
Extreme performance that is enterprise ready

- **Real-time performance** on large datasets
  - replace slow and costly batch processing

- **Easy to integrate** with existing infrastructure
  - realise value within hours of deployment

- **Easy to use** and based on established technology
  - existing investments in code work out-of-the-box

- **Extensions and APIs**
  - easy to add custom functionality

Rapid Deployment

Improved ROI

Immediate Value
Data Acquisition

- A single line of code is all that is needed to connect to a third-party data source
- Connection treated as a table within Brytlyt and can be used directly within SQL queries
- Over thirty data connectors already exist
- Data connectors are bi-directional with both read and write capability
- Using the Foreign Data Wrapper API it is quick and easy to create a new connector
- Add streaming workloads
- Very easy to incorporate into ETL processing
- Bulk import and export also available

User Interaction

- PostgreSQL client tools used to manage and work with data using ANSI SQL
- Stored procedures, database cursors, native JSON support, arrays and much more
- User Defined Functions can be developed in C/C++/CUDA where necessary
- Torch Machine Learning and AI available with zero data copy
- SpotLyt visualization tool built on Plot.ly with twenty interactive chart types
- Point map can navigate billions of geospatial data points in real time
- PostgreSQL data connector means any visualization tool can benefit from GPU acceleration

GPU Acceleration

- Brytlyt patent pending IP combined with Graphic Processor Unit (GPU) acceleration
- Query billions of rows of data in milliseconds, up to 1,000 times faster than alternatives
- Fast GPU JOIN
- Hot data caching in CPU RAM used for immediate access to large data sets
- GPUs used to filter and render billions of geospatial data points in real time
- Horizontal scale out allows multiple servers to be used in a single cluster

SpotLyt

- Visualisation tool built on Plot.ly with twenty interactive chart types
- Point map can navigate billions of geospatial data points in real time
- PostgreSQL data connector means any visualisation tool can benefit from GPU acceleration
Pairing Brytlyt with best-in-class IBM POWER hardware to deliver the next generation Database & Analytics Platform
Brytlyt is a PostgreSQL fork
Open Source

MariaDB
BrytMind

SQL + AI + GPU
With Zero Copy
Zero Copy
By Extending Torch Memory Management

- SQL DDL and DML via PostgreSQL
- AI and ML operations via Torch
SpotLyt

SQL + AI Workbench for Interactive analytics on billion row datasets
Demo
Market Leader with Superior Product and IP

**Brytlyt DB**
- Patent pending IP
- PostgreSQL - stored procedures, correlated queries, cursors, nested queries, window functions, native JSON and XML support
- The first GPU database to run TPC-H benchmark
- Uses Torch memory caching

**SpotLyt**
- Visualization tool with server side graphics rendering on GPU
- Twenty customizable and interactive charts
- Built on open source Plot.ly
- Single developer produced SpotLyt in only six weeks

**BrytMind**
- SQL + Artificial Intelligence + GPU
- Integrate Torch directly into GpuManager and BrytlytDB