Agenda

• Deep Learning at scale - TensorFlow integration
• Multi-language support – Thin clients
• Transparent data encryption
• Transaction SQL - Beta
• Q&A
Deep Learning at Scale
TensorFlow Integration
Continuous Learning Framework Enables Simplified ML/DL Workflow — No ETL

**Before**

- **App**
- **ML/DL Engine**
  - Model training & testing
  - Loading data for training
- **RDBMS**
  - Storing and processing working set
- **HADOOP**
  - Periodic ETL of terabytes of data

**After (With CLF)**

- **App**
- **IMC Platform**
  - Storing and processing working set
  - Instant updates of models
- **DB + ML/DL Engine**
  - Model training & testing
  - No ETL
Continuous Learning Framework

*In-Memory Machine and Deep Learning*

**GridGain Continuous Learning Framework**

- Java
- .NET
- Binary Protocol (Thin client)
- REST
- C++

- K-Means
- Regressions
- Decision Trees
- TensorFlow

**Distributed Core Algebra**

**Compute and Service Grid**

**In-Memory Data Store**

**Persistent Store**

**Features**

- Distributed Algorithms
- Large Scale Parallelization
- Multi-language Support
- Dense and Sparse Algebra
- No ETL
TensorFlow Integration

• Ignite as distributed data source
  – Perfect fit for distributed TF training

• Less ETL
  – TF nodes deployed together with Ignite nodes
    – In-machine data movement only

• TF tasks execution in-place in Ignite
  – Roadmap
TensorFlow Integration: Main Features

- Distribution of user tasks written in Python
- Automatic creation and maintenance of TF cluster
- Minimization of ETL costs
- Fault tolerance for both Ignite and TF instances
Multi-language support
Thin Clients
Thin Clients

• Socket-level binary protocol
  – Cross-platform

• Thin Client is
  – An API above the protocol and
  – Language specific implementation

• Supported languages
  – Java, .NET, C++
  – Node.JS, Python, PHP
## Regular Client vs. Thin Client

<table>
<thead>
<tr>
<th></th>
<th><strong>Regular (aka. Thick) Client</strong></th>
<th><strong>Thin Client</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cluster Communication</strong></td>
<td>• Starts a JVM process &lt;br&gt;• Part of a cluster topology &lt;br&gt;• Several ports have to be opened</td>
<td>• Connects through a proxy &lt;br&gt;• TCP/IP connection &lt;br&gt;• One port has to be opened</td>
</tr>
<tr>
<td><strong>Scalability and performance</strong></td>
<td>• Best &lt;br&gt;• Goes to primary nodes directly</td>
<td>• Worse &lt;br&gt;• Communicates through the proxy &lt;br&gt;• Partition-awareness is to be supported</td>
</tr>
<tr>
<td><strong>Supported APIs</strong></td>
<td>• All Ignite and GridGain APIs</td>
<td>• Get/put, SQL, configuration</td>
</tr>
<tr>
<td><strong>Supported Languages</strong></td>
<td>• Java, .NET, C++ &lt;br&gt;• Time consuming to support other language</td>
<td>• Easy to support any language &lt;br&gt;• Java, .NET, C++, Node.js, Python, PHP are already available</td>
</tr>
</tbody>
</table>
Transparent Data Encryption
Transparent Data Encryption

- Sensitive and personal data encryption

- Applied for data on disk
  - Data (aka. partition) files
  - Write-ahead-logs

- Per-cache/table encryption
  - Cache encryption keys
  - Stored in system cache

- Master key
  - Used for persistence and transferring of cache keys
  - Pluggable store – JDK Store is available out of the box
Transactional SQL
Beta
Transactional SQL - BETA

• Fully transactional support for SQL
  – SELECTs, INSERTs, UPDATEs, DELETEs
  – Based on MVCC

• Snapshot isolation
  – TRANSACTIONAL_SNAPSHOT mode
  – Transactions are managed by coordinator

• Current limitations
  – https://apacheignite.readme.io/docs/multiversion-concurrency-control

BEGIN;
INSERT INTO Person (id, name, city_id) VALUES (1, 'John Doe', 3);
Update City SET population = population + 1 where id = 3;
ROLLBACK;
Resources

- TensorFlow Integrations:  
  - https://apacheignite.readme.io/docs/tensor-flow

- Thin Clients:  
  - https://apacheignite.readme.io/docs/thin-clients

- Transparent Data Encryption  
  - https://apacheignite.readme.io/docs/transparent-data-encryption

- Transactional SQL  
  - https://apacheignite-sql.readme.io/docs/transactions  
  - https://apacheignite.readme.io/docs/transactions

- Full List of Changes  
  - https://ignite.apache.org/releases/2.7.0/release_notes.html
Apache Ignite Support – Faster Time to Reliable Ignite

• Get up and running faster with 2 hours initial consultation
• Ensure fast, reliable Ignite with unlimited 9x5 global support
  – Unlimited web/e-mail support
  – Identify bugs, workarounds
  – Troubleshoot performance, reliability issues

https://www.gridgain.com/products/services/support/support-apache-ignite
Any Questions?
@apacheignite
@gridgain