MySQL Multi-Site/Multi-Master
MySQL High Availability and Disaster Recovery
~~~
Heterogeneous Real-Time Data Replication
Oracle Replication
Continuent Quick Introduction

**History**

- **2004**: Continuent founded in USA
- **2009**: 3rd Generation Continuent Tungsten ships
- **2014**: 100+ customers running business-critical applications
- **Q4 2014**: Acquisition by VMware
- **Q4 2016**: Continuent spun off from VMware back to an independent entity
- **Q1 2017**: Continuent released Tungsten Replication 5.0 and Tungsten Clustering 5.0
- **Q1 2018**: Continuent will release all new Tungsten Replication 6.0 and Tungsten Clustering 6.0

**Products**

Industry-leading clustering and replication for open source DBMS

**Tungsten Clustering** – Commercial-grade HA, performance scaling, and data management for MySQL

**Tungsten Replication** – Flexible, high-performance real-time data loading
Continuent Facts

### Business-Critical Deployment Examples

<table>
<thead>
<tr>
<th>Category</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Availability for MySQL</td>
<td>Largest cluster deployment performs 1,000M+ transactions/day on 300+ TB of relational data</td>
</tr>
<tr>
<td>Business Continuity</td>
<td>Cross-site cluster topologies widely deployed including primary/DR and multi-master</td>
</tr>
<tr>
<td>High Performance Replication</td>
<td>Largest installations transfer billions of transactions daily using high speed, parallel replication</td>
</tr>
<tr>
<td>Heterogeneous Integration</td>
<td>Customers replicate from MySQL to Oracle, Hadoop, Redshift, Vertica, and others</td>
</tr>
<tr>
<td>Real-time Analytics</td>
<td>Optimized data loading for data warehouses with deployments of up to hundreds of AWS RDS and MySQL masters feeding to AWS Redshift, Hadoop and Vertica</td>
</tr>
</tbody>
</table>
Select Continuent Customers

Adobe
Bluefin Payment Systems
CARFAX
Clareity Security
Clarivate Analytics

ebay
Financial Times
F-Secure
GARMIN
KIBO

Leonardo
Marketo
Modernizing Medicine
NCR
Net-a-Porter Group

NewVoiceMedia
Samsung
Synacor
vmware
Zappos

continuent
MySQL Clustering for HA and DR
The Dream: Multiple, active DBMS servers with identical data over distance – Too good to be true?

- High Performance
- Transparent read/write to any server
- Updates propagated immediately to all servers
- High Availability
Synchronous multi-master clusters claim to deliver on the dream
Synchronous multi-master introduces new problems

- Slow writes due to synchronous messaging
- Cross-site replication is unstable
- Operations like SELECT FOR UPDATE not supported
- Large transactions lock system or cause failures
Can master/slave clusters offer the same benefits?

- High Performance?
- High Availability?
- Transparent read/write to any server?
- Updates propagated immediately?
Tungsten Clustering
MySQL Master/Slave Cluster for HA/DR
Tungsten Clustering: HA, DR and Performance Scaling

Benefits
- 24x7 data access
- Off-the-shelf MySQL
- SQL load balancing
- Simple management

Diagram:
- Application Stack
  - Tungsten Connector
    - Reads
    - Writes
    - SLAVE
    - MASTER
    - SLAVE
Tungsten clusters add HA and scaling without taking features away
Tungsten Connector operates as an intelligent proxy to the DBMS

- Any MySQL client can connect
- Connector initiates connections on behalf of client to the DBMS
Bridge mode in the Connector minimizes overhead from proxying

- Bridge mode provides full transparency and very low overhead by allowing clients to connect directly to MySQL
- We also offer various intelligent routing algorithms, including session-aware SmartScale
SmartScale provides session load balancing

• Initial write goes to master
• Reads go to replicas if it is safe to do so
SmartScale provides session load balancing

- Auto-commit reads (queries not inside an explicit transaction) are eligible to go to slave
- Reads stay on master until a slave catches up
SmartScale provides session load balancing

- Reads go to slave when it has caught up with master
- Session tags may be schema name or supplied by application

MySQL Master

```
Select Data
Tungsten Connector
(Session “X” Binlog Position)
```

MySQL Slave

MySQL Slave

MySQL Slave

Auto-commit reads from slave!

Write committed

Received but not applied

Received and applied

continuent
Connectors can be configured to support different levels of service

- Each connector may be configured differently
- Some may be read-only, some may be Bridge mode, others may run in automatic read/write splitting Proxy mode, etc.
Failover and Maintenance
Tungsten clusters automatically monitor all cluster nodes for failure

- Tungsten Manager monitors database health
- Query flow and replication traffic are managed
Cluster rules fail over master if DBMS no longer accepts network connections

- Tungsten Manager detects a database outage
- Query flow stops, and a new master is chosen

1. Detect and shun a non-responsive node
2. Halt in-coming connections
3. Find and promote the most up-to-date slave
Failed nodes can be re-provisioned from a backup with a single management command

- The replicator on the promoted node is then brought online as a Master
- Query flow is re-enabled so as to get the application online as quickly as possible
- Lastly, any remaining slave replicators will be re-pointed to the newly promoted master

4. Bring replicator online as Master
5. Resume query flow using new Master
6. Reconfigure Slave replicators to use the new Master
7. Administrator recovers the shunned node as a Slave
Tungsten clusters support Zero-downtime Maintenance operations from parameter changes to app upgrade

- **Task**: Upgrade MySQL to the latest version
- **Problem**: Requires a mysqld restart, hence can cause application downtime
- **Constraint**: Avoid application-visible restart
- **Solution**: Upgrade nodes in succession
Rolling maintenance proceeds node-by-node starting with slaves and proceeding to master

- **Slave upgrade**
  - Shun slave
  - Upgrade MySQL
  - Return node to cluster
  - Discard and re-provision on failure

- **Slave upgrade**
  - Repeat for remaining slave(s)

- **Switch**
  - Switch master to promote an upgraded slave

- **Master upgrade**
  - Upgrade old master
  - Maintenance is now done!
Tungsten Clustering Benefit Summary

- 24x7 data access
- SQL load balancing
- Off-the-shelf MySQL support
  - MySQL Community/Enterprise
  - Percona Server and MariaDB
- Simple management
Transaction Scaling with Master/Slave Topologies
Size and transaction activity on business data depend on many factors

SaaS Datasets -- Size of Top 1000 Customers

- Max = 1214 GB
- 99th percentile = 290 GB
- Median = 2.6 GB
DBMS workloads are correspondingly varied

- Complex queries
- Large batch operations
- Small online transactions
- Analytic reports

MySQL Master
MySQL Slave
MySQL Slave
Asynchronous replication decouples transaction processing on master and slave DBMS nodes

MySQL/Oracle

DBMS-specific Logging (i.e. Redo or Binary)

DBMS Logs

MySQL/Oracle

Extractor

Extractor Options

2

1

Option 1: Local Install
Extractor reads directly from the logs, even when the DBMS service is down. This is the default.

Option 2: Remote
Extractor gets log data via MySQL Replication Slave protocols (which requires the DBMS service to be online) or the Redo Reader feature. This is how we handle RDS and Oracle extraction tasks.

Master Replicator: Extractor

Slave Replicator: Applier

Download transactions via network

Apply using JDBC

THL = Events + Metadata

THL

continuent
Parallel apply maximizes DBMS I/O bandwidth when updating replicas
Distributing Data between Regions and to Other DBMS types
Tungsten Disaster Recovery creates composite clusters that span sites and are ready for immediate failover.
Tungsten Multi-Site/Multi-Master topologies operate independent, active clusters on 2 or more remote sites.
The same replication mechanism supports real-time loading of data warehouses.
Heterogeneous Real-Time Data Replication
Data Warehouse Integration is Changing

- Traditional data warehouse usage was based on dump from transactional store, loads into data warehouse
- Data warehouse and analytics were done off historical data loaded
- Data warehouses often use merged data from multiple sources, which is difficult to combine and manage
- Data warehouses are now frequently sources as well as targets for data, i.e.:
  - Export data to data warehouse
  - Analyze data
  - Feed summary data back to application to display stats to users
Modern Data Warehouse Sequences

Analysis

Raw Text → Hadoop → Map/Reduce → SQL

Data Results

SQL → SQL Query → Hadoop

SQL → Hadoop → Map/Reduce → SQL
How do we cope with that model

• Traditional Extract-Transform-Load (ETL) methods take too long
• Data needs to be replicated into a data warehouse in real-time
• Continuous stream of information
• Replicate everything
• Use data warehouse to provide join and analytics
Data Warehouse Choices

• Hadoop
  – General purpose storage platform
  – MapReduce for data processing
  – Front-end interfaces for interaction in SQL-like (Hive, HBase, Impala) and non-SQL (Pig, native, Spark)
  – JDBC/ODBC Interfaces improving
• Vertica
  – Massive cluster-based column store
  – SQL and ODBC/JDBC Interface
• Amazon Redshift
  – Highly flexible column store
  – Easy to deploy
Tungsten Replicator: Data Warehouses

Tungsten Replicator
is a fast, open source
database replication engine

Designed for speed and flexibility
Apache license
100% open source
Annual support subscription available
Tungsten Master/Slave in Action (MySQL)

Option 1: Local Install
Extractor reads directly from the logs, even when the DBMS service is down. This is the default.

Option 2: Remote
Extractor gets log data via MySQL Replication Slave protocols (which requires the DBMS service to be online). This is how we handle RDS extraction tasks.
Tungsten Master/Slave in Action (Redo Reader, All Oracle Versions)

- **Master Replicator:** Extractor
  - Extractor gets event data by reading the PLOG files generated by the Redo Reader (PLOG generation requires the DBMS service to be online).

- **DBMS Logs**
- **Redo Reader**
  - Download transactions via network
  - THL = Events + Metadata

- **Slave Replicator:** Applier
  - Apply using JDBC
  - THL

Oracle → Redo Logging → DBMS Logs → Redo Reader → Master Replicator: Extractor → Target

Extracted PLOG is used for applying changes to the target database.
Tungsten Master/Slave in Action
(Oracle CDC up through 11g)

Extractor gets change data by acting as a subscriber to the CDC subsystem in Oracle (which requires the DBMS service to be online).

**Oracle**

- Extractor
  - Master Replicator: Extractor
  - THL = Events + Metadata
- Oracle CDC
- Change Tables

**Target**

- Slave Replicator: Applier
  - THL
- Apply using JDBC
- Download transactions via network
The Data Warehouse Impedance Mismatch

Transactional Store → Dump/Provision → Batch → Data Warehouse

Transactions? ×
Transactional and Data Warehouse Metadata

• Replicating data is not just about the data
• Table structures must be replicated too
• `ddlscan` handles the translation
  – Migrates an existing MySQL or Oracle schema into the target schema
  – Template based
  – Handles underlying datatype matches
  – Needs to be executed before replication starts
The same replication mechanism supports real-time loading of data warehouses
Replicating into Vertica

Master Replicator -> Slave Replicator
Slave Replicator -> JS
JS -> Vertica

Master Replicator -> CSV
CSV -> cpimport
Vertica -> merge

JDBC
Replicating into Redshift
How Kafka Replication Works

MySQL/Oracle

DBMS-specific Logging [i.e. Redo or Binary]

DBMS Logs

Zookeeper

Kafka Applier [Native]

Download transactions via network

Master Replicator: Extractor

THL = Events + Metadata

Slave Replicator: Applier

THL
What Tungsten Replicator Does to Apply into Kafka

- Takes an incoming row and converts it to a message
- Message consists of metadata:
  - Schema name, table name
  - Sequence number
  - Commit timestamp
  - Operation Type
- Embedded Message Content
Message Structure

Schema
  Table
    Row
    Row
    Row
    Row
    Row
    Row

Topic: Schema_Table

MsgID: Schema Table PKey

Row

Row
How Elasticsearch Replication Works

- Redo Logging
- DBMS Logs
- Redo Reader
- Master Replicator: Extractor
- THL
- THL = Events + Metadata
- Slave Replicator: Applier
- Elasticsearch Applier (REST API)
- Download transactions via network
- Generated PLOG

Redo Logging

DBMS Logs

Redo Reader

Master Replicator: Extractor

THL

THL = Events + Metadata

Slave Replicator: Applier

Elasticsearch Applier (REST API)

Download transactions via network

Generated PLOG
How Cassandra Replication Works

Master Replicator → Slave Replicator → CSV

Slave Replicator → JS → Ruby Connector → Cassandra

Cassandra → merge
Replicating into Hadoop
Initial Materialization within Hadoop

Migrate staging/base DDL

load-reduce-check

HDFS

CSV

Staging Table

Hive materialization

Base Table

continuent
Sqoop and Materialization within Hadoop

Sqoop

Replicate

Hive materialization

Base Table

Staging Table

HDFS

CSV
Ongoing Materialization within Hadoop

materialize

Base Table

Hive materialization

Staging Table

HDFS

CSV

continuent
Provisioning Options (all data warehouses)

- MySQL
  - Traditional CSV export and import
  - Dump and load through Blackhole engine
  - Use tungsten_provision_thl

- Oracle
  - Traditional CSV export and import
  - Use parallel extractor
Provisioning Options (Hadoop)

- MySQL
  - Traditional CSV export and import
  - Dump and load through Blackhole engine
  - Use tungsten_provision_thl
  - Use Sqoop

- Oracle
  - Traditional CSV export and import
  - Use parallel extractor
  - Use Sqoop
## Comparing Loading Methods for Hadoop

<table>
<thead>
<tr>
<th></th>
<th>Manual via CSV</th>
<th>Sqoop</th>
<th>Tungsten Replicator</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Process</strong></td>
<td>Manual/Scripted</td>
<td>Manual/Scripted</td>
<td>Fully Automated</td>
</tr>
<tr>
<td><strong>Incremental Loading</strong></td>
<td>Possible with DDL changes</td>
<td>Requires DDL changes</td>
<td>Fully Supported</td>
</tr>
<tr>
<td><strong>Latency</strong></td>
<td>Full-load</td>
<td>Intermittent</td>
<td>Real-time</td>
</tr>
<tr>
<td><strong>Extraction Requirements</strong></td>
<td>Full table scan</td>
<td>Full and partial table scans</td>
<td>Low-impact CDC/binlog scan</td>
</tr>
</tbody>
</table>
### How the Materialization Works

<table>
<thead>
<tr>
<th>Op</th>
<th>Seqno</th>
<th>ID</th>
<th>Msg</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>1</td>
<td>Hello World!</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>2</td>
<td>Meet MC</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>3</td>
<td>1</td>
<td>Goodbye World</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Op</th>
<th>Seqno</th>
<th>ID</th>
<th>Msg</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>2</td>
<td>2</td>
<td>Meet MC</td>
</tr>
<tr>
<td>I</td>
<td>3</td>
<td>1</td>
<td>Goodbye World</td>
</tr>
</tbody>
</table>
Replication Support (Hadoop specific)

- Extract from MySQL or Oracle
- Hadoop Support
  - Cloudera (Certified), HortonWorks, MapR, Pivotal, Amazon EMR, IBM (Certified), Apache
- Provision using Sqoop or parallel extraction
- Schema generation for Hive
- Tools for generating materialized views
- Parallel CSV file loading
- Partition loaded data by commit time
- Schema Change Notification
Data Warehouse Possibilities:
Point-in-Time Tables
# Data Warehouse Possibilities: Time Series Generation

<table>
<thead>
<tr>
<th>Op</th>
<th>Seqno</th>
<th>ID</th>
<th>Date</th>
<th>Msg</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1</td>
<td>1</td>
<td>1/6/14</td>
<td>Hello World!</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
<td>2</td>
<td>2/6/14</td>
<td>Meet MC</td>
</tr>
<tr>
<td>I</td>
<td>3</td>
<td>1</td>
<td>2/6/14</td>
<td>Goodbye World</td>
</tr>
<tr>
<td>I</td>
<td>4</td>
<td>1</td>
<td>3/6/14</td>
<td>Hello Tuesday</td>
</tr>
<tr>
<td>I</td>
<td>4</td>
<td>2</td>
<td>3/6/14</td>
<td>Ruby Wednesday</td>
</tr>
<tr>
<td>I</td>
<td>5</td>
<td>1</td>
<td>4/6/14</td>
<td>Final Count</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Date</th>
<th>Msg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/6/14</td>
<td>Hello World!</td>
</tr>
<tr>
<td>1</td>
<td>2/6/14</td>
<td>Goodbye World</td>
</tr>
<tr>
<td>1</td>
<td>3/6/14</td>
<td>Hello Tuesday</td>
</tr>
<tr>
<td>1</td>
<td>4/6/14</td>
<td>Final Count</td>
</tr>
</tbody>
</table>
Oracle DR and Replication
Replication to/from MySQL

Oracle Master

Oracle Slave

MySQL Tungsten Clustering

Primary Replication Path

Secondary Replication Path
Replication to/from Oracle

- **Primary Replication Path**
  - Oracle RAC
  - Oracle Standalone

- **Secondary Replication Path**
  - Oracle RAC
  - Oracle Standalone
Replication to Data Warehouses

Oracle
Single or RAC

Replication

Amazon RedShift
Hadoop
Vertica
Wrap-Up
Master/slave clustering is a robust technology for enterprise data management!

- Very High Performance
- Transparent connectivity with full SQL semantics
- Updates propagated without cost to applications
- Very High Availability
Continuent offers...

• Highly available clusters of off-the-shelf MySQL servers
• Zero-downtime maintenance and upgrade
• High performance regardless of data volume or distance
• Replication over regions to DR sites as well as non-MySQL data warehouses
Getting Started!

• Tungsten Replicator builds are available on GitHub: http://github.com/continuent/tungsten-replicator

• Replicator documentation is available on the Continuent website: http://docs.continuent.com/tungsten-replicator-5.1

• Tungsten Hadoop tools are available on GitHub: https://github.com/continuent/continuent-tools-hadoop

Contact Continuent for Annual Support Subscriptions!
Tungsten Multi-Site/Multi-Master topologies operate independent, active clusters on 2 or more remote sites.
## Competitive Scorecard
### Clustering for HA, DR and MSMM

<table>
<thead>
<tr>
<th>Top Prioritized Use Cases</th>
<th>Tungsten Clustering</th>
<th>Amazon RDS</th>
<th>RDS w/ Aurora</th>
<th>MySQL native</th>
<th>MySQL Fabric</th>
<th>Galera Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HA (Local)</strong></td>
<td>Yes, automatic failover for primary and secondaries</td>
<td>Yes, automatic failover, can take minutes across AZs</td>
<td>Yes, automatic failover and auto replicated across three AZs</td>
<td>Not supported directly; third party tool required</td>
<td>Automatic failover and redirection</td>
<td>Not supported; can be directed to another host</td>
</tr>
<tr>
<td><strong>DR (Local/Remote)</strong></td>
<td>Yes, both local, remote and global DR is supported</td>
<td>Supported, w/ significantly increased latency</td>
<td>Supported w/ significantly increased latency</td>
<td>Not supported; replication to a remote site is possible</td>
<td>Not supported, a third party tool required</td>
<td>Not supported</td>
</tr>
<tr>
<td><strong>Load Balancing</strong></td>
<td>Yes; provided automatically and transparently</td>
<td>Not supported w/o a third party tool such as HA Proxy</td>
<td>Not supported w/o a third party tool such as HA Proxy</td>
<td>Not supported. Can be combined with MySQL Fabric</td>
<td>Supported for read and writes, through sharding and distribution</td>
<td>Automatically load balances the reads/writes to multiple primaries</td>
</tr>
<tr>
<td><strong>Zero Downtime Maintenance</strong></td>
<td>Enables both database, operating system and hardware changes</td>
<td>Not supported; changes or AWS outages can have significant impact</td>
<td>Not supported; changes or AWS outages can have significant impact</td>
<td>Not built-in. In MySQL 5.6 the use of GTIDs can make stop/restart easy</td>
<td>Not supported without a more complex configuration</td>
<td>Changes are supported, but can be costly and time consuming</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>High; built-in load balancing and read/write splitting- scale cluster up and down</td>
<td>Variable; Depends on I/O and connectivity requested; No auto distribution</td>
<td>Variable; Depends on I/O and connectivity requested; No auto distribution</td>
<td>Native performance is high, but not w/ complex topology</td>
<td>High; providing the data has been sharded and distributed effectively</td>
<td>Synchronous nature implies additional overhead during writes</td>
</tr>
<tr>
<td><strong>Transparency</strong></td>
<td>Requires no app changes or modifications to achieve all functionality</td>
<td>Requires no app changes; for transparency requires deployment w/i AWS</td>
<td>No app changes, needs add'l components for multi region or load balancing</td>
<td>Dependent on 3rd party solution used</td>
<td>Must use Oracle drivers</td>
<td>Reg. Galera cluster and app made aware of changes</td>
</tr>
</tbody>
</table>
# Competitive Scorecard

Oracle Replication and Real-Time Loading into Analytics

<table>
<thead>
<tr>
<th>Top Prioritized Use Cases</th>
<th>Tungsten Replicator</th>
<th>Oracle GoldenGate</th>
<th>Shareplex</th>
<th>Attunity</th>
<th>Talend</th>
<th>Dbvisit Replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Replication Targets</td>
<td>Oracle, MySQL, PostgreSQL, Hadoop, Vertica, MongoDB, Amazon Redshift, Kafka, Cassandra</td>
<td>Oracle, MySQL, Microsoft SQL Server, IBM DB2, Sybase, Hadoop, Kafka</td>
<td>Oracle, Hadoop, SQL Server, Kafka, Teradata</td>
<td>Oracle, Hadoop, SQL Server, Teradata, Others, Kafka</td>
<td>MongoDB, Cassandra, Hadoop, CouchDB, Couchbase, Risk</td>
<td>Oracle, MySQL, Microsoft SQL Server, Kafka, Tibero and PostgreSQL</td>
</tr>
<tr>
<td>Continuous Replication</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Not supported</td>
<td>Yes</td>
</tr>
<tr>
<td>Topology Support</td>
<td>One to One, One to Many, Many to One</td>
<td>One to One, One to Many, and Many to One</td>
<td>One to One</td>
<td>One to One, One to Many, Many to One</td>
<td>One to One, One to Many by repeating extract</td>
<td>One to One, One to Many, Many to One</td>
</tr>
<tr>
<td>Encryption/SSL</td>
<td>Between database and replicators and between replicators</td>
<td>Enhanced through secure password store</td>
<td>Not supported</td>
<td>Between replicators</td>
<td>Limited support to databases</td>
<td>Between replicators</td>
</tr>
<tr>
<td>Database Impact</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>Depends on database</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Change Data Capture Exposed</td>
<td>Yes</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Limited</td>
<td>Not supported</td>
<td></td>
</tr>
<tr>
<td>Filtering</td>
<td>Advanced selection, content and reconstruction filtering</td>
<td>Limited filtering and transformations</td>
<td>Limited table/column filtering</td>
<td>Limited table/column filtering</td>
<td>Advanced transformations and loading</td>
<td>Row filtering supported; Limited table/column filtering</td>
</tr>
<tr>
<td>Failure and monitoring</td>
<td>Simple status and start/stop recovery enable fast restart</td>
<td>Failure identified complex, bounded recovery makes restart easy</td>
<td>Replication stalls and errors difficult to identify and recovery from</td>
<td>Web-console</td>
<td>Limited; failure requires complete reload rather than restart</td>
<td>Status and console access</td>
</tr>
</tbody>
</table>

---

- **Poor**
- **Good**
For more information, contact us:

Eero Teerikorpi
Founder, CEO
eero.teerikorpi@continuent.com

MC Brown
VP Products
mc.brown@continuent.com

Eric Stone
COO
eric.stone@continuent.com

Matthew Lang
Director, Professional Services USA
matt.lang@continuent.com

Petri Virsunen
VP Marketing
petri.virsunen@continuent.com

Chris Parker
Director, Professional Services EMEA & APAC
chris.parker@continuent.com