Basics: The Power of the Connector

continuent

Topics

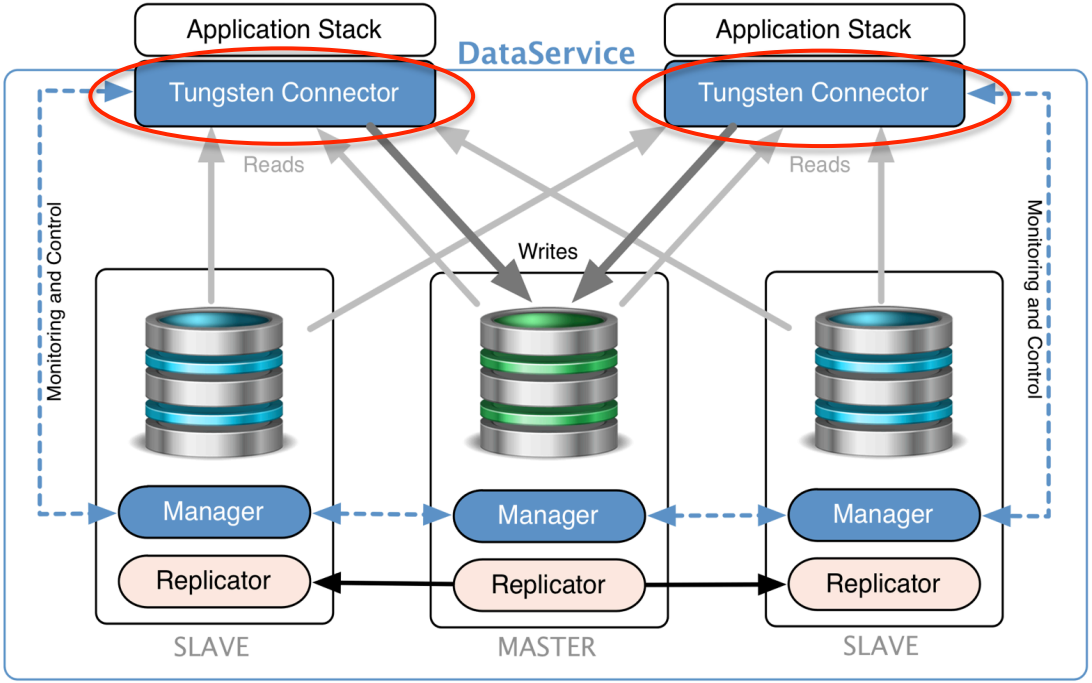
In this today's webinar we will:

- Review the Cluster Architecture
- Understand the role of the Connector
- Describe Connector deployment best practices (app, dedicated with lb, db with lb)
- Explore zero-downtime maintenance using the manual role *switch* procedure
- Learn about Connector Routing Patterns inside a Composite Cluster
- Illustrate a manual site switch
- Explain Read Affinity and the vast performance improvement of local reads
- Examine Connector multi-cluster support

continuent

Tungsten Cluster Architecture: Review

Tungsten Cluster Architecture: Review



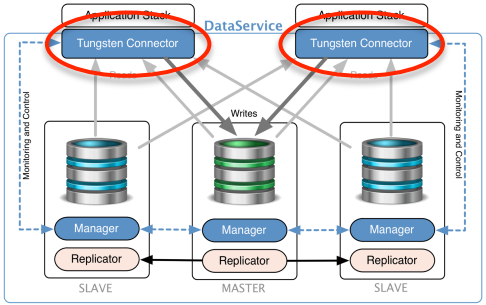
continuent

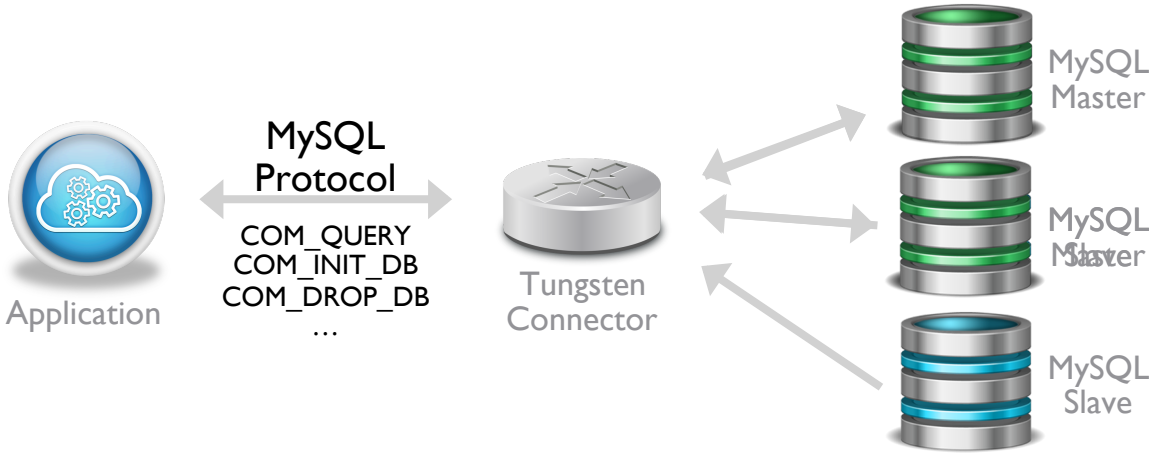
What does the Connector do?

continuent

The Connector

- Any MySQL client can connect
- Connector initiates connections on behalf of client to the DBMS





continuent

Connector Deployment Best Practices

continuent

Deployment Best Practices: General

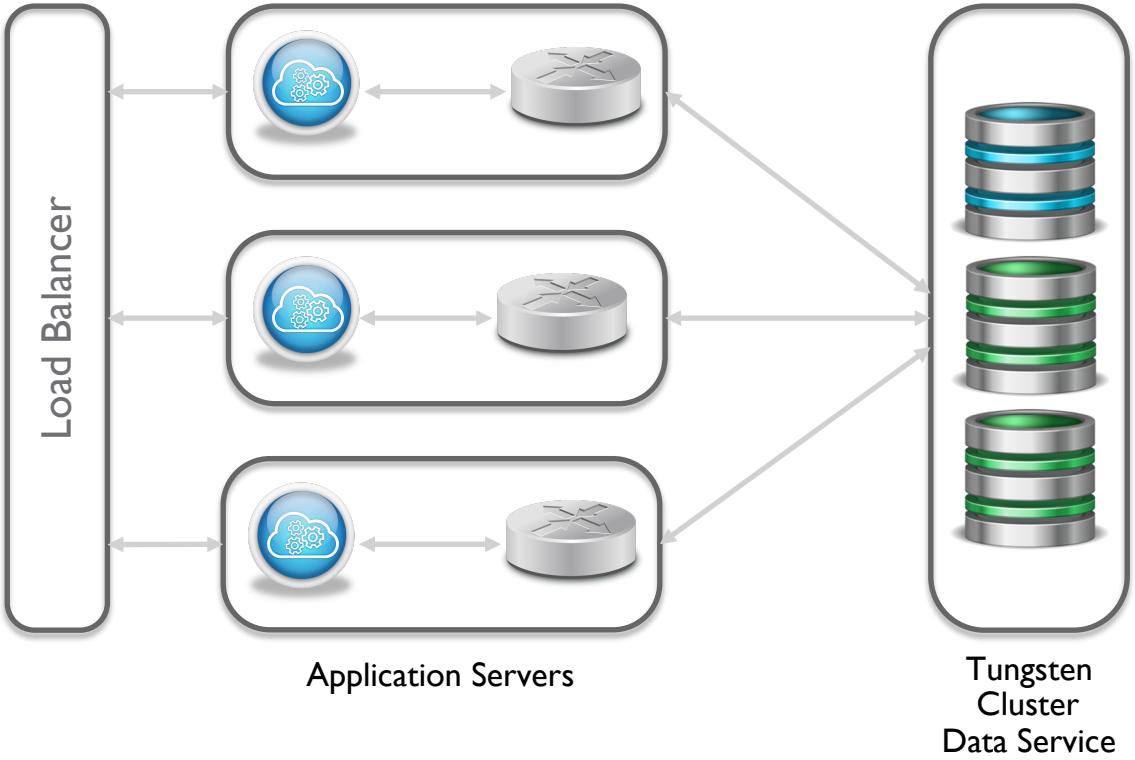
- There is a practical limit to the quantity of Connectors (approximately 20, based upon environmental speed in terms of CPU and bandwidth) due to the synchronous nature of the status updates sent to the Connectors by the Managers.
- Connectors can be configured independently from each other for even finer-grained control.
- Choosing an instance size depends upon the volume of connections. Each connection requires a certain amount of CPU and memory. Bridge mode requires the least, while Proxy mode and SmartScale use up much more because of the active inspection process.
- HA Proxy software is free, fast, stable and mature.
- Using the INI installation method is much easier when using DevOps tools such as Puppet, Chef or Ansible.

Deployment Best Practices: Topologies

- There are three main deployment topologies:
  1. Connector on the application servers, implied load balancing at the application layer
  2. Connector on dedicated Connector hosts, minimum of 2, load balancing required
  3. Connector on the Cluster Database nodes, minimum of 2, load balancing required
- For #2 and #3, either a hardware or software (i.e. HA Proxy) load balancing solution may be employed.
- All three topologies may be in use at the same time!

continuent

Deployment Topologies: Connector on the Application Servers



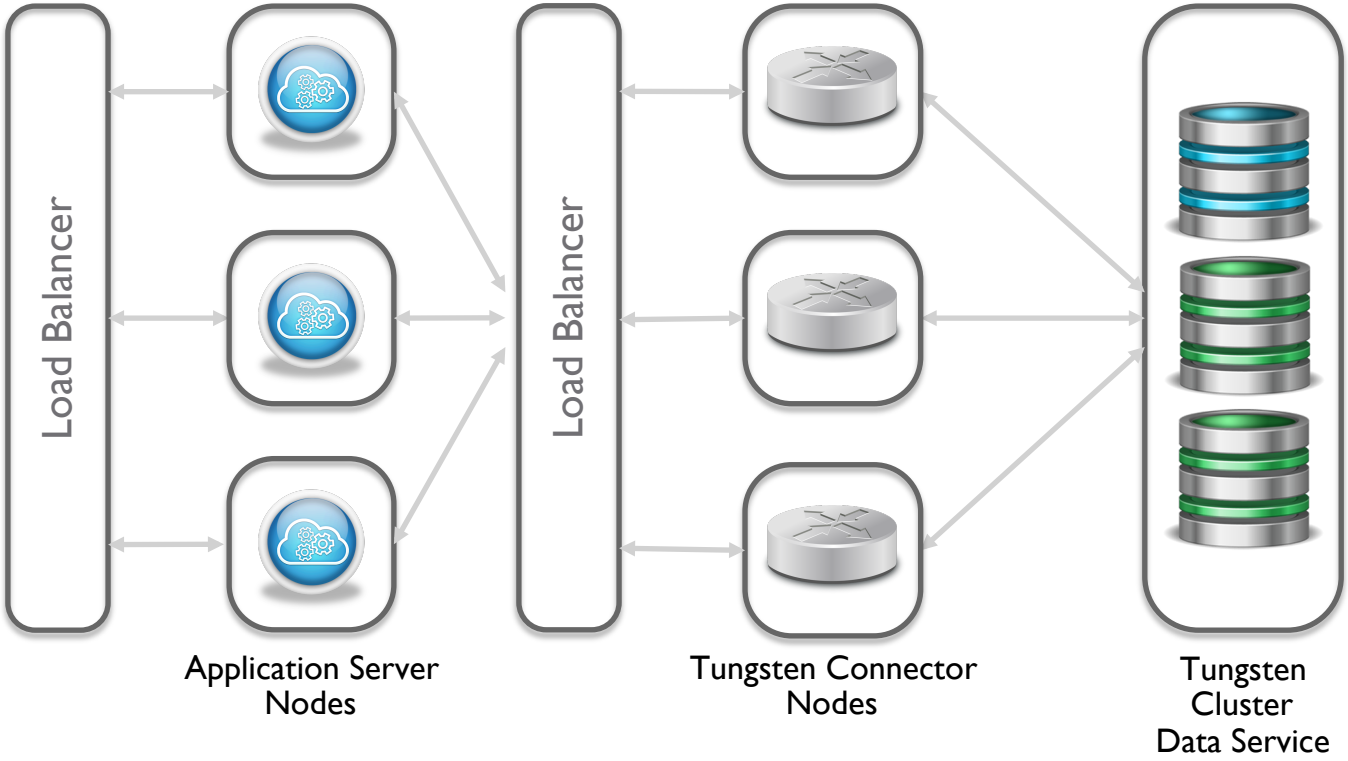
continuent

Deployment Topologies: Connector on the Application Servers

- Load balancing is assumed at the application layer, i.e. in front of the application nodes.
- If the connector somehow dies, the upper-layer load balancer should detect an application failure, and remove that node from the pool.
- No load balancing solution between the application and the connector is needed if the above assumptions are correct.
- Applications are configured to communicate to 127.0.0.1 on port 3306.
- There is a 1:1 relationship between application node and connector, so if you have 5 web servers, then you would have 5 connectors in use.
- If you will have more than twenty (20) application servers, we recommend using dedicated nodes for the connectors.

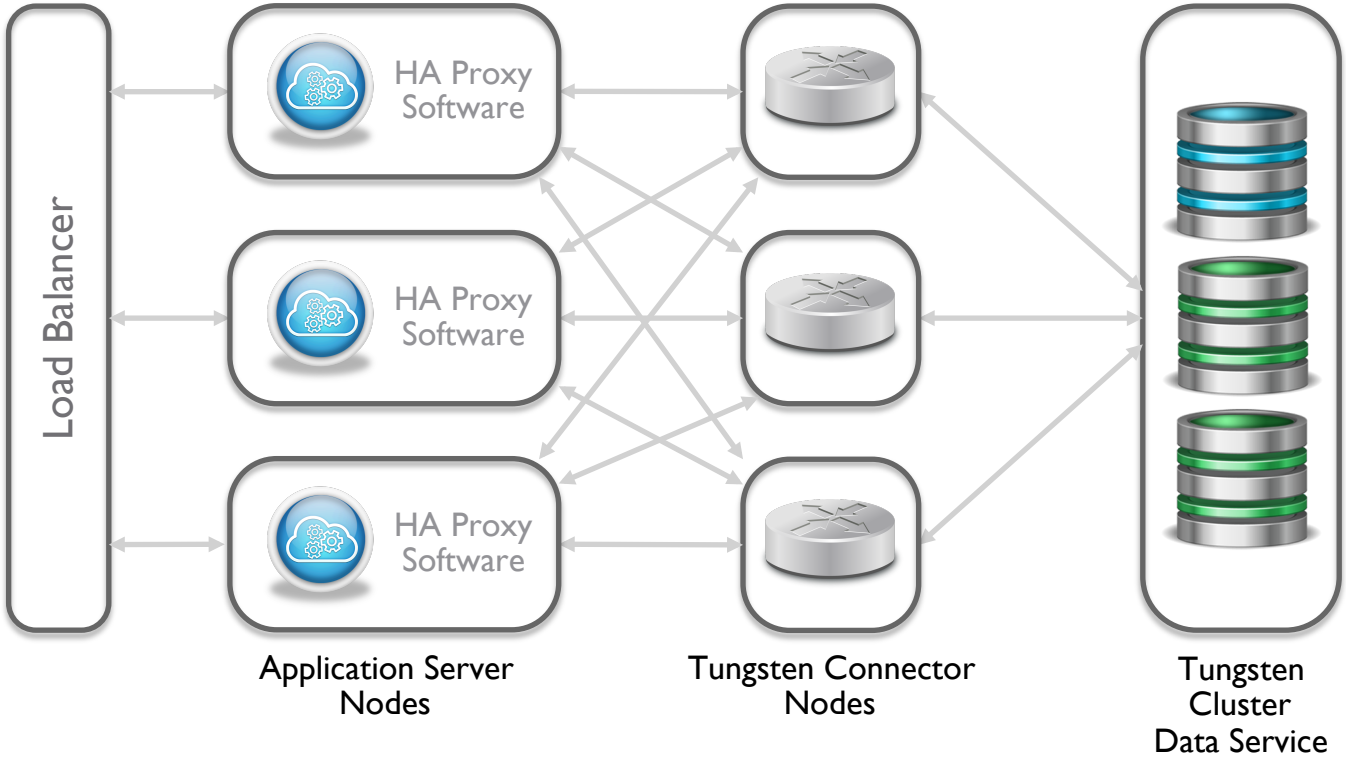
continuent

Deployment Topologies: Connector on Dedicated Nodes & Hardware Load Balancer



continuent

Deployment Topologies: Connector on Dedicated Nodes & HA Proxy Software



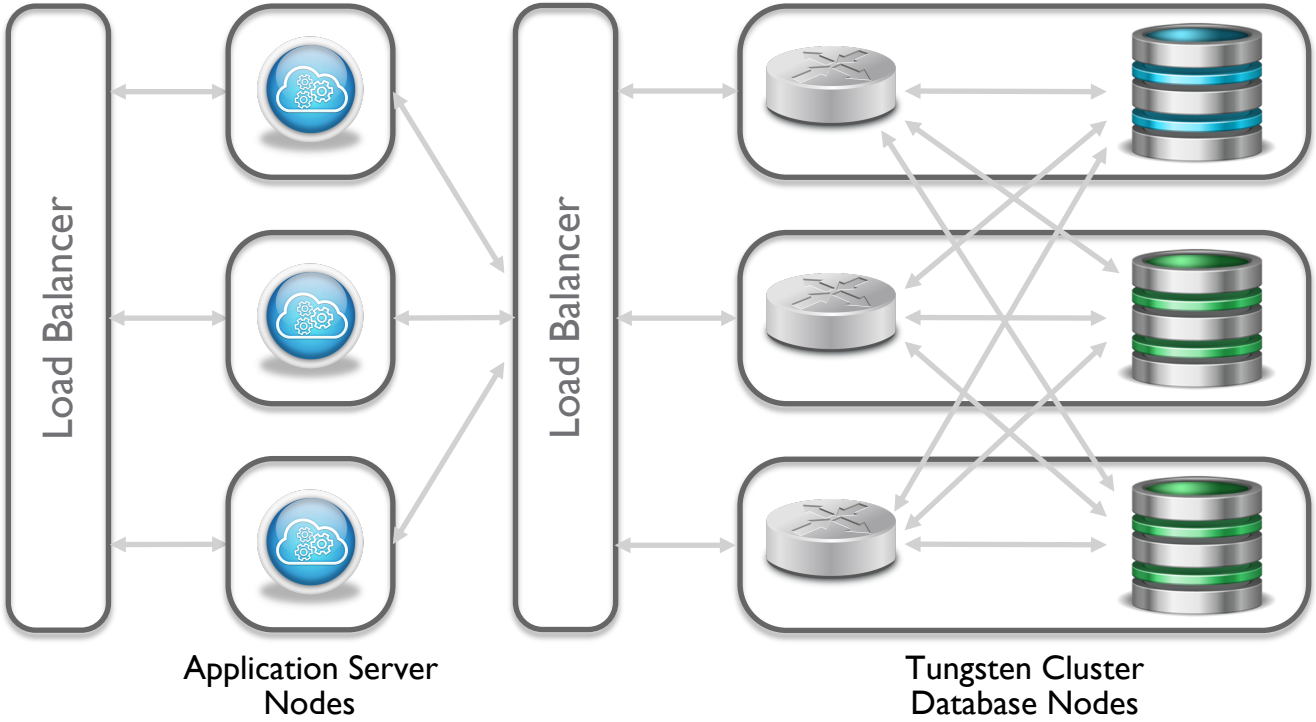
continuent

Deployment Topologies: Connector on Dedicated Nodes

- Load balancing is required because the Connectors do not have their own HA capabilities, they provide it for the database layer.
- If a Connector somehow dies, the load balancer needs to choose another, which is why a minimum of two (2) are required in this topology.
- Applications are configured to communicate to the load balancer. For HA Proxy, most likely will be 127.0.0.1 on port 3306. For a hardware load balancer, it will be whatever IP and port you select.
- This is the most desirable topology if you are unable to place the Connectors on the application nodes.

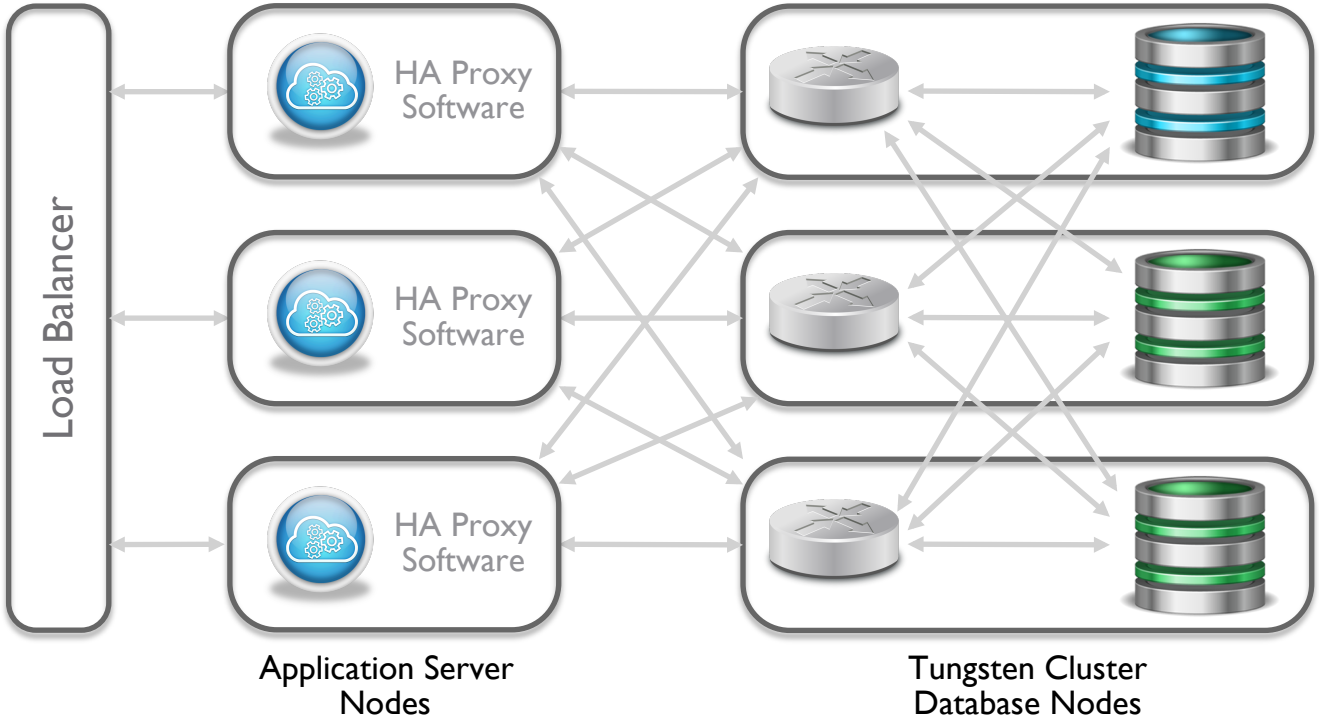
continuent

Deployment Topologies: Connector on the Cluster Database Nodes & HW LB



continuent

Deployment Options: Connector on the Cluster Database Nodes & HA Proxy



continuent

Deployment Options: Connector on the Cluster Database Nodes

- Load balancing is required because the Connectors do not have their own HA capabilities, they provide it for the database layer.
- If a Connector somehow dies, the load balancer needs to choose another, which is why a minimum of two (2) are required in this topology.
- Applications are configured to communicate with the load balancer. For HA Proxy, most likely will be 127.0.0.1 on port 3306. For a hardware load balancer, it will be whatever IP and port you select.
- This is the least-desirable topology because of the dual-use nature of this topology. The nodes now need to share resources with another process. The Connector is a traffic ROUTER, and as such, could send traffic THROUGH the node to yet another cluster node. This implies significant overhead in terms of CPU, memory and network bandwidth.
- Usually deployed in smaller environments.

Zero-Downtime Rolling Maintenance

continuent

Zero-Downtime Maintenance: The Switch

- The key to performing maintenance is the manual switch operation, where another node is selected to be the master.
- The switch command is invoked inside the cctrl command-line utility. In the below example, the Manager is allowed to select the new node:

```
cctrl> switch
```

- Instead of allowing the cluster to decide, you can switch to a specific node:

```
cctrl> switch to db3
```

- Make sure replication is 100% caught up before you switch because the cluster will let you switch to a new master that is missing transactions!

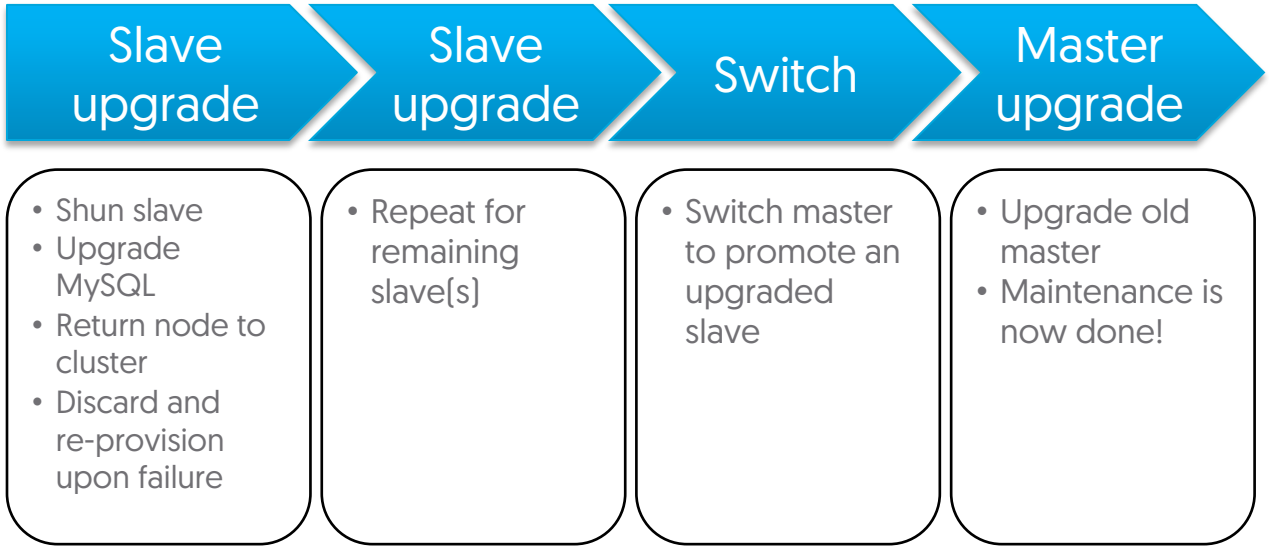
Zero-Downtime Maintenance: Switch, Manager Picks

- When the new master is not specified, the Manager uses rules to decide which node to pick:
  - Skip any slave that is either not online or that is a standby slave.
  - Skip any slave that has its status set to ARCHIVE
  - Skip any slave that does not have an online manager.
  - Skip any slave that does not have a replicator in either online or synchronizing state.
  - Now we have a target datasource prospect.
  - By comparing the last applied sequence number of the current target datasource prospect to any other previously seen prospect, we should eventually end up with a slave that has the highest applied sequence number. We also save the prospect that has the highest stored sequence number.
  - If we find that there is a tie in the highest sequence number that has been applied by any prospect with another prospect, we compare the datasource precedence and if there's a difference in this precedence, we choose the datasource with the lowest precedence number i.e. a precedence of 1 is higher than a precedence of 2. If we have a tie in precedence, we just choose the last slave that we have evaluated.
  - After we have evaluated all of the slaves, we will either have a single winner or we may have a case where we have one slave that has the highest applied sequence number but we have another slave that has the highest stored sequence number i.e. it has gotten the most number of THL records from the master prior to the switch operation. In this case, and this is particularly important in cases of failover, we choose the slave that has the highest number of stored THL records.

Zero-Downtime Maintenance: Let it Roll

- Rolling maintenance proceeds node-by-node starting with the slaves and ending with the master. http://docs.continuent.com/tungsten-clustering-5.2/operations-maintenance-dataservice.html
- A switch operation will wait for a short period of time (default: 5s) for running database queries to complete gracefully
  - This delay may be adjusted using the tpm option **--connector-disconnect-timeout**
  - http://docs.continuent.com/tungsten-clustering-5.2/connector-advanced-waitfordisconnecttimeout.html
- The application will experience a hang or a disconnect based on the mode you are using
  - Bridge mode
    - The application will be disconnected with an error returned
    - Automatic reconnection to the new master happens upon **application retry**
  - Proxy/SmartScale (non-Bridge) modes
    - If the connection is idle when the switch happens then the application connection will hang and by default will automatically reconnect to the new master when a new request is issued
    - If there was a **non-transaction read** operation in-flight, then it will be retried upon reconnect. In all other cases (i.e. long-running writes), the application will get an error and will need to retry.

Rolling maintenance proceeds node-by-node starting with the slaves and ending with the master



continuent

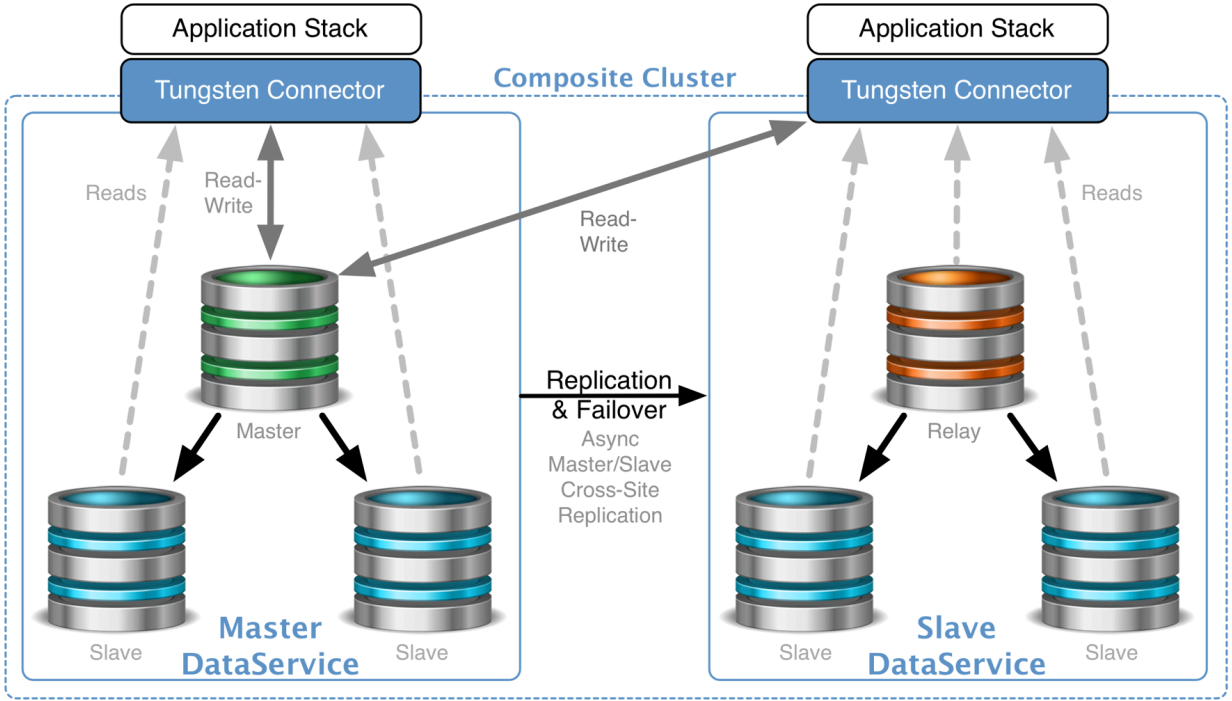
DEMO: Rolling maintenance

```
shell$ echo ls | cctrl
shell$ cctrl
Continuent Tungsten 5.2.0 build 235
east: session established, encryption=false, authentication=false
[LOGICAL] /east > datasource db3 shun
*Perform maint on db3
[LOGICAL] /east > recover (~or~ [LOGICAL] /east > datasource db3 welcome)

[LOGICAL] /east > datasource db2 shun
*Perform maint on db2
[LOGICAL] /east > recover

[LOGICAL] /east > switch to db2
SELECTED SLAVE: db2@east
SET POLICY: AUTOMATIC => MAINTENANCE
PURGE REMAINING ACTIVE SESSIONS ON CURRENT MASTER 'db1@east'
PURGED A TOTAL OF 0 ACTIVE SESSIONS ON MASTER 'db1@east'
FLUSH TRANSACTIONS ON CURRENT MASTER 'db1@east'
PUT THE NEW MASTER 'db2@east' ONLINE
PUT THE PRIOR MASTER 'db1@east' ONLINE AS A SLAVE
RECONFIGURING SLAVE 'db3@east' TO POINT TO NEW MASTER 'db2@east'
REVERT POLICY: MAINTENANCE => AUTOMATIC
SWITCH TO 'db2@east' WAS SUCCESSFUL

[LOGICAL] /east > datasource db1 shun
*Perform maint on db1
[LOGICAL] /east > recover
[LOGICAL] /east > ls
```

continuent

Composite Cluster Architecture: Connector Routing Patterns

Tungsten Composite Cluster Architecture



continuent

Tungsten Composite Cluster Architecture: Connector Routing Patterns

- A Composite cluster is a “Cluster of clusters”
- In a Composite cluster, there can be only one Primary cluster, and only one writable master node in that local cluster
- The master in all other clusters is called a relay node, responsible for pulling events from the Primary cluster and serving it to downstream slaves in the other non-Primary local clusters.
- By default all traffic goes to the master node in the Primary cluster
- Once read-write splitting is enabled, reads may go to slaves.
- The Connector uses the concept of Affinity to adjust the traffic patterns.
  More on this after the demo...

continuent

DEMO: Composite Cluster Manual Site Switch

```
shell$ cctrl -multi
Continuent Tungsten 5.2.0 build 235
east: session established, encryption=false, authentication=false
[LOGICAL] / > use global
[LOGICAL] /global > ls
…DATASOURCES:
|east(composite master:ONLINE)
|STATUS [OK] [2017/08/23 08:33:12 PM UTC]
|west(composite slave:ONLINE)
|STATUS [OK] [2017/08/23 08:33:12 PM UTC]

[LOGICAL] / > use east
[LOGICAL] /east > ls
…
|db1(master:ONLINE, progress=5, THL latency=0.130)

[LOGICAL] / > use west
LOGICAL] /west > ls
…
|db4(relay:ONLINE, progress=5, latency=0.166)
```

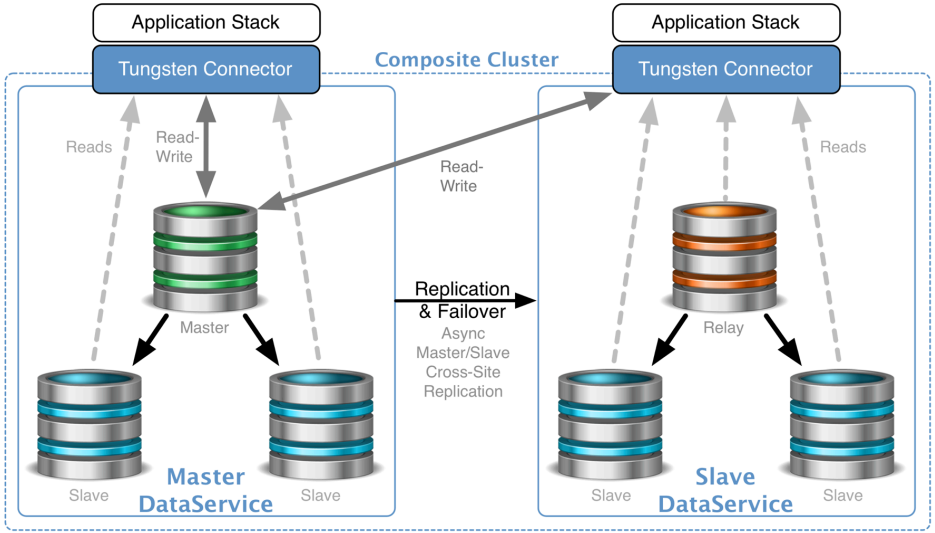
```
[LOGICAL] /global > switch
SELECTED SLAVE: 'west@global'
FLUSHING TRANSACTIONS THROUGH 'db1@east'
REPLICATOR 'db1' IS NOW USING MASTER CONNECT URI 'thl://db4:2112/'
composite data source 'west@global' is now OFFLINE
PUT THE NEW MASTER 'west@global' ONLINE
PUT THE PRIOR MASTER 'east@global' ONLINE AS A SLAVE
REVERT POLICY: MAINTENANCE => AUTOMATIC
SWITCH TO 'west@global' WAS SUCCESSFUL

[LOGICAL] /global > ls
…DATASOURCES:
|east(composite slave:ONLINE)
|STATUS [OK] [2017/08/23 10:14:12 PM UTC]
|west(composite master:ONLINE)
|STATUS [OK] [2017/08/23 10:14:12 PM UTC]

[LOGICAL] / > use east
[LOGICAL] /east > ls
|db1(relay:ONLINE, progress=8, latency=5.142)

[LOGICAL] / > use west
LOGICAL] /west > ls
|db4(master:ONLINE, progress=8, THL latency=0.706)
```

continuent

Composite Cluster Architecture: Connector Routing Affinity

Tungsten Composite Cluster: Connector Routing Affinity

- Affinity enables you to specify *at connection time* that the connector should forward the connection to a particular host or service for reads, if the host or service is available.
- Affinity is very useful with Composite clusters because it allows us to specify an entire service (site) to prefer for reads. This means that reads can come from local nodes instead of over the WAN, vastly improving read performance in geo-distributed clusters.



http://docs.continuent.com/tungsten-clustering-5.2/connector-routing-affinity.html

continuent

Tungsten Composite Cluster: Connector Routing Affinity

- Non-Bridge mode Connectors may be individually configured via the per-host `user.map` file to prefer the local site. For example, to set the affinity to prefer the west site/service in the global composite cluster:

```
#<username> <password> <service-name> <affinity>
eric secret global west
```

- Connectors will recognize arguments in the database connection string supplied by the client:
  - `shell> mysql -h127.0.0.1 –P3306 databasename@qos=RO_RELAXED\&affinity=west`
  - `jdbc://host4:3306/database?affinity=west&qos=RO_RELAXED`
- Affinity can be set globally using the tpm option `--connector-affinity`
- When specifying `--connector-affinity` in Bridge mode, any Affinity or QoS specified within the connection string are ignored. For example, in Bridge mode with `--connector-affinity=west` specified, the below is NOT possible:

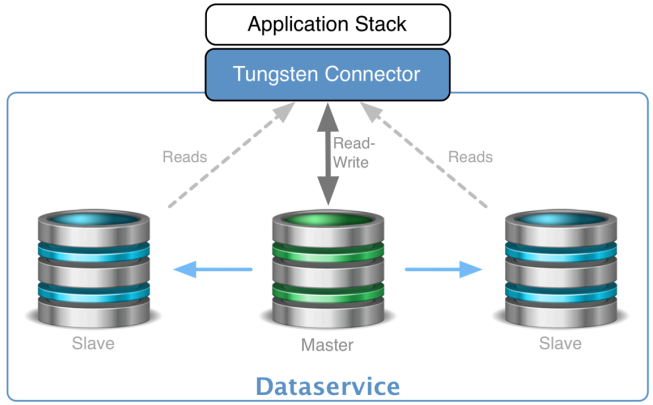
```
shell> mysql -h127.0.0.1 -P9999 databasename@qos=RO_RELAXED\&affinity=east
```

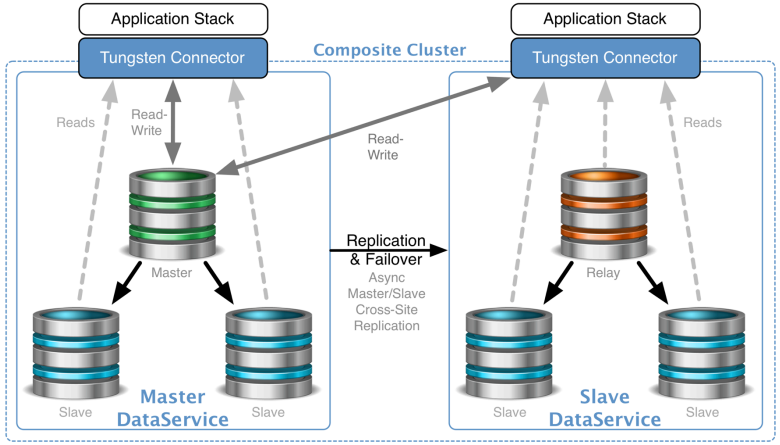
continuent

Connector Multi-Cluster Support

Connector Multi-Cluster Support: Architecture

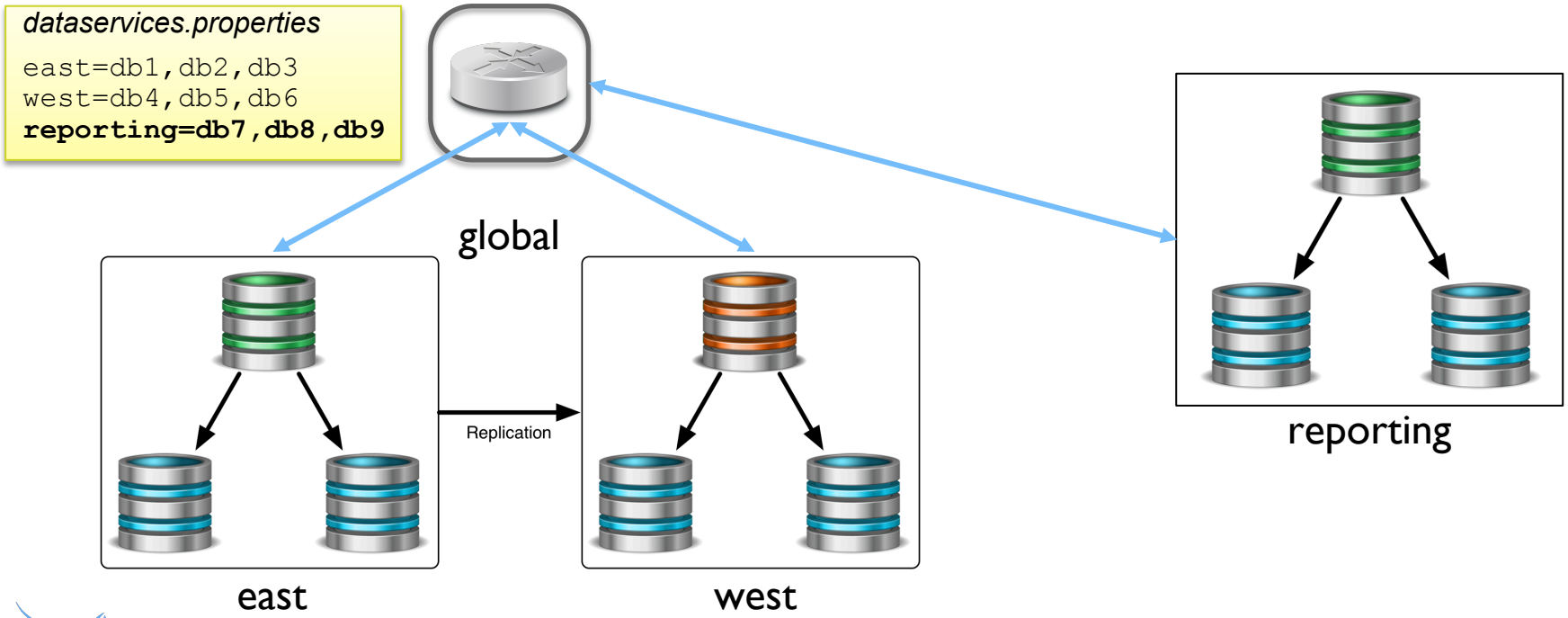
- We can think of any local or composite cluster as one single set of data (a dataservice), with a single authoritative node for writes.
- Normally a Connector is configured to speak to a single dataservice (i.e. local service *east* or composite service *global*)
- A Connector may provide write access to more than one cluster (dataservice) or composite cluster by editing a single file on each Connector node: `dataservices.properties`





continuent

Connector Multi-Cluster Support: Architecture



continuent

Connector Multi-Cluster Support: The *dataservices.properties* File

- Location: `cluster-home/conf/dataservices.properties`
- Format: one dataservice per line, with all manager hosts for each dataservice:

```
<dataservice-name1>=<manager-host>[,<manager-host>...]
<dataservice-name2>=<manager-host>[,<manager-host>...]
```

- Connectors use this file to 'discover' data services and to get the configuration of the data service. Managers use this file to find other services and to connect to a manager in that service.
- As you update dataservices.properties on each host, restart the connector and manager (if any) on that host.
- If you update dataservices.properties on both connector hosts and db hosts, you will be able to easily connect to every data service listed in dataservices.properties from cctrl.

Connector Multi-Cluster Support: The Details

- Once the Connector is up and running as part of a cluster, edit the file:
  `/opt/continuent/tungsten/cluster-home/conf/dataservices.properties`
- Add a line for each new master/slave cluster the connector will connect to, then restart the Connector.

```
east=db1,db2,db3
west=db4,db5,db6
reporting=db7,db8,db9
```

- Do not list composite dataservices in this file. The connector will automatically discover those from the managers in each cluster. Keep this file updated as you add and remove servers from each cluster.
- Update the user.map to list new users for each new dataservice.

```
<username> <password> <service-name> <affinity>
chris secret global west
eric secret global east
matt secret reporting
```

- There is a one-to-one mapping between a given user name and a specific service, so you **CANNOT have a user name appear more than once in a given user.map**

http://docs.continuent.com/tungsten-clustering-5.2/connector-advanced-multi.html
http://docs.continuent.com/tungsten-clustering-5.2/connector-authentication-config.html

continuent

Webinar Q & A/Wrap-Up

continuent

Webinar Summary

In this today's webinar we:

- Reviewed the Cluster Architecture
- Looked at the role of the Connector
- Described Connector deployment best practices (app, dedicated with lb, db with lb)
- Explored zero-downtime maintenance using the manual role *switch* procedure
- Learned about Connector Routing Patterns inside a Composite Cluster
- Illustrated a manual site switch
- Explained Read Affinity and the vast performance improvement of local reads
- Examined Connector multi-cluster support

continuent

Next Steps

- If you are interested in knowing more about the clustering software and would like to try it out for yourself, please contact our sales team who will be able to take you through the details and setup a POC – sales@continuent.com

- Read the documentation at http://docs.continuent.com/tungsten-clustering-5.2/connector.html

- Subscribe to our Tungsten University YouTube channel! http://tinyurl.com/TungstenUni

- Visit the events calendar on our website for upcoming Webinars and Training Sessions https://www.continuent.com/events/

continuent

For more information, contact us:

Eero Teerikorpi
Founder, CEO
eero.teerikorpi@continuent.com
+1 (408) 431-3305

Eric Stone
COO
eric.stone@continuent.com

MC Brown
VP Products
mc.brown@continuent.com

Matthew Lang
Director, Professional Services Americas
matthew.lang@continuent.com

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