Implementing the Future of PostgreSQL Clustering with Tungsten

Robert Hodges

CTO, Continuent, Inc.

continuent

Agenda

- Introductions
- Framing the Problem: Clustering for the Masses
- Introducing Tungsten
- Adapting Tungsten to PostgreSQL
- Questions and Comments

About Continuent

- **Our Business:** Continuous Data Availability
- **Our Solution**
  - Continuent Tungsten (Master/Slave Database Replication)
- **Our Value:**
  - Ensure data are available when and where you need them
  - TCO less than 20% of comparable solutions
- **Our Technical Expertise**
  - Database replication
  - Database cluster management
  - Application connectivity
- **Our Partner**
  - 2ndQuadrant and Simon Riggs (thanks, Simon)

Framing the Problem: Clustering for the Masses

Terminology

Cluster: A group of hosts connected by a network that work together to perform some useful task

2005 - 2015: Rapid Technological Change

- **>95% of apps need only one DBMS host**
  - Multi-core processors
  - Cheap main memory
  - Solid state devices (SSDs)
- **Shared infrastructure dominates operations**
  - Virtualization/clouds for small DBMS
  - Shared database instances for ISP/SaaS
- **Massive growth in non-OLTP uses**
  - Cheap, simple data stores
  - Read-intensive, web-facing applications
  - Webscale processing

2005 - 2015: Changing User Needs

- Availability
- Data Protection
- Resource utilization
- Performance
- Open source/commercial integration
- Geographically distributed data

2005-2015: What's Cool and What's Not

- **Tight coupling is OUT**
  - Master/master (Postgres-R, Sequoia)
  - Shared disk (Oracle RAC)
- **Loose coupling is IN**
  - Master/slave (MySQL)
  - Eventual consistency (SimpleDB, BigTable, Bucardo)
- **Simple management is IN**
- **Efficient utilization is IN**
  - Partitioning/multi-tenant models
  - Migration to more/less capable resources
  - Virtualized operation
- **Data protection is IN**

Introducing Tungsten

continuent

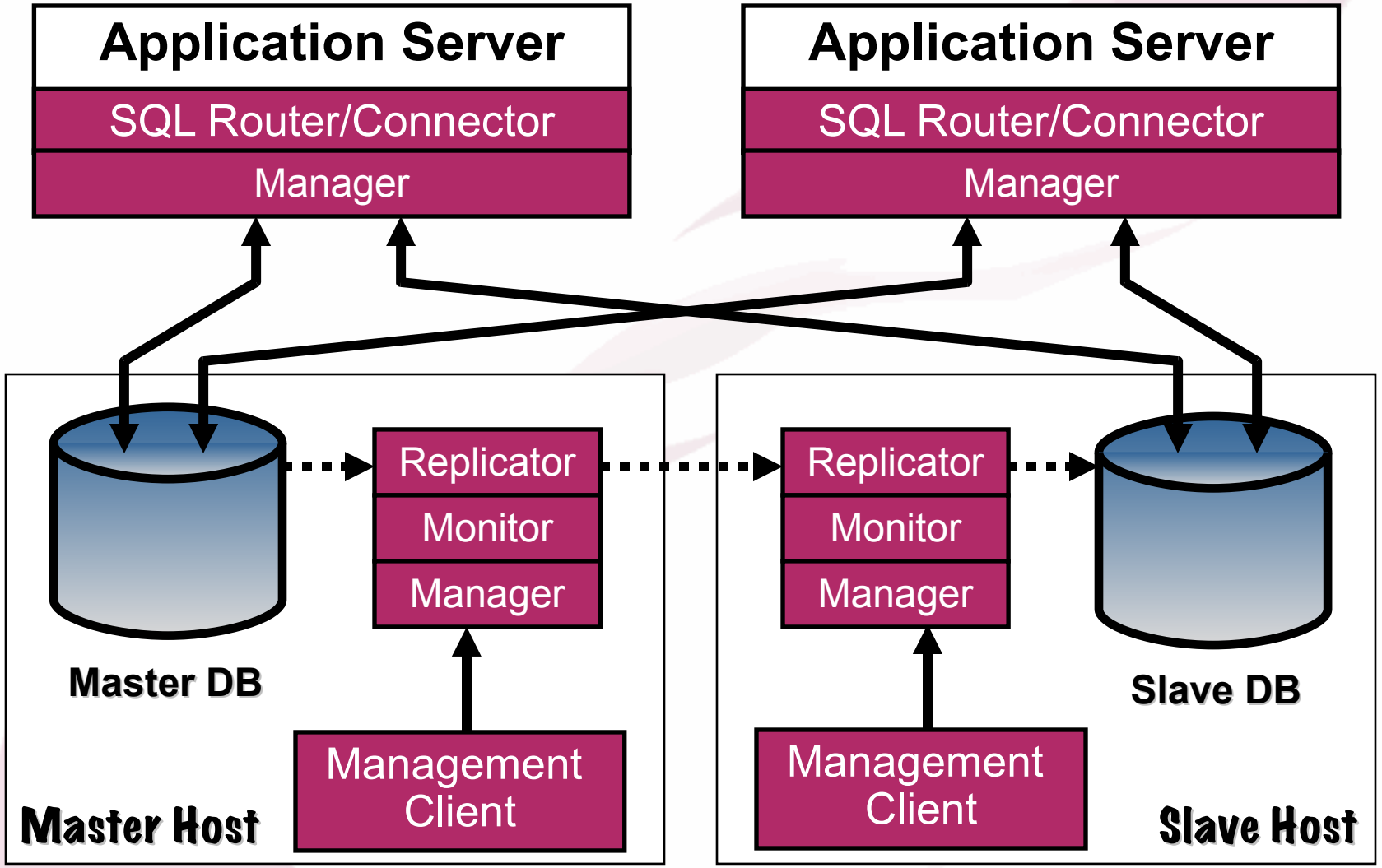
What Is Tungsten?

- **Tungsten implements master/slave clusters to:**
  - Protect data
  - Maintain high availability
  - Improve resource utilization
  - Raise performance
- **Install and set up in a few minutes**
- **Integrated backup/restore and data integrity checks**
- **Efficient failover operations**
- **Distributed, rule-driven management**
- **No/minimal application changes**
- **Highly pluggable**
- **No specialized hardware requirements**

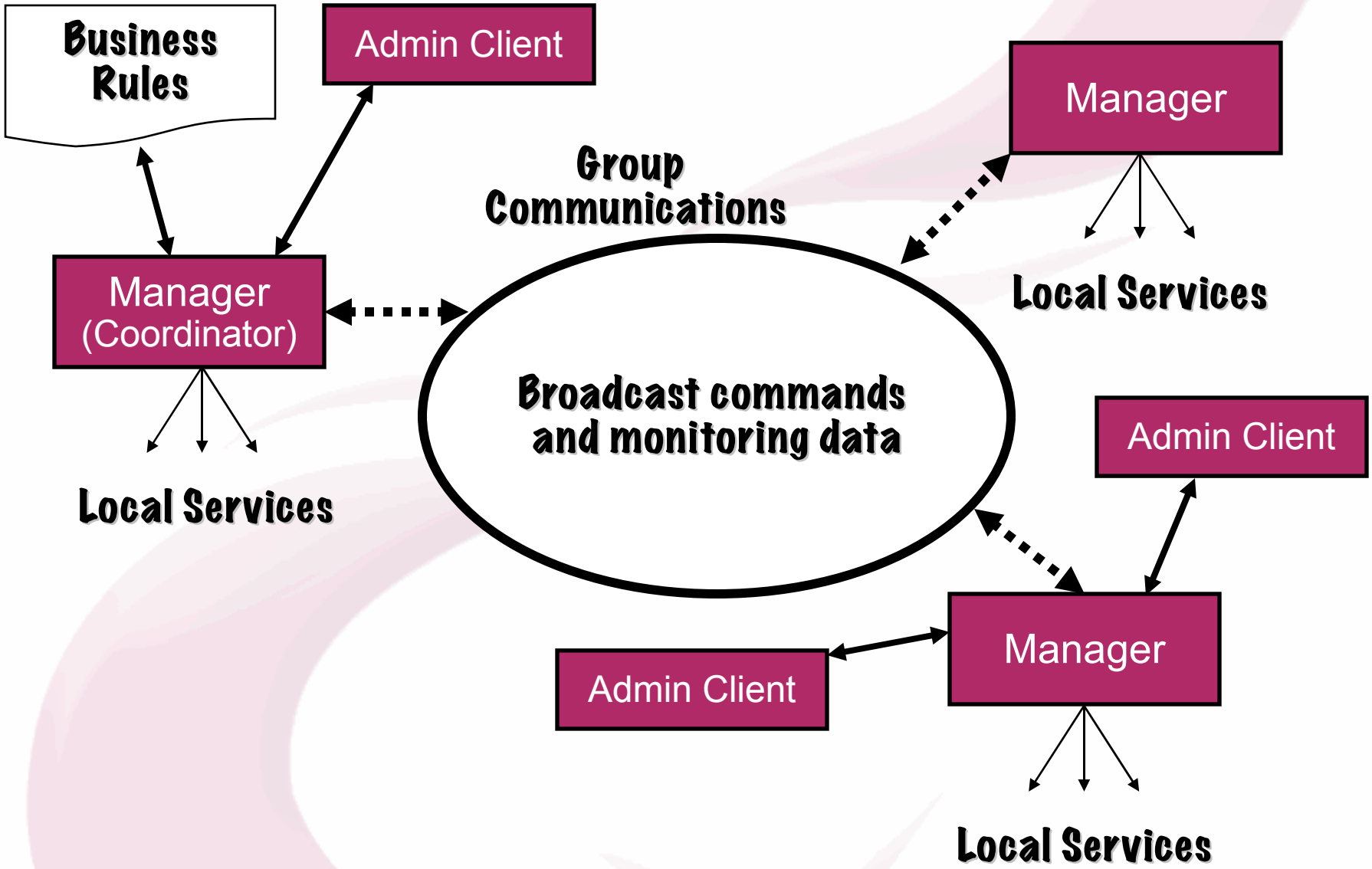
Tungsten Open Source Foundation

- **Tungsten Replicator**
  - Database-neutral, platform independent master/slave replication
  - Extensible to manage other types of replication
- **Tungsten Connector**
  - Fast MySQL/PostgreSQL client to JDBC proxying
- **Tungsten SQL Router**
  - JDBC wrapper for high-performance and transparent failover, load-balancing, and partitioning (no proxy required)
- **Tungsten Manager**
  - Distributed administration with autonomic, rule-based configuration and no single point of failure
- **Tungsten Monitor**
  - Measure latency and detect whether resources are up/down

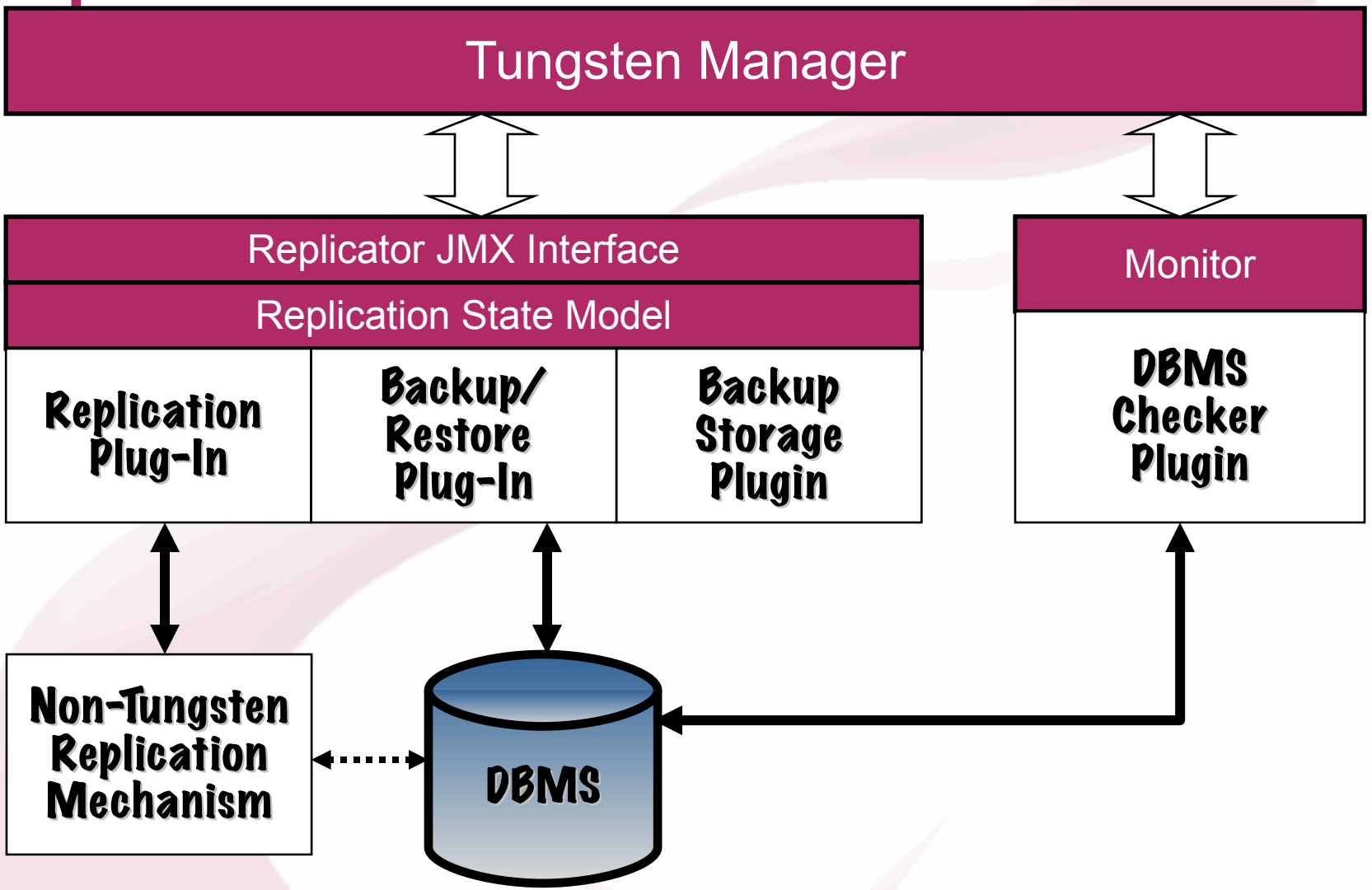
Tungsten Clustering In Action



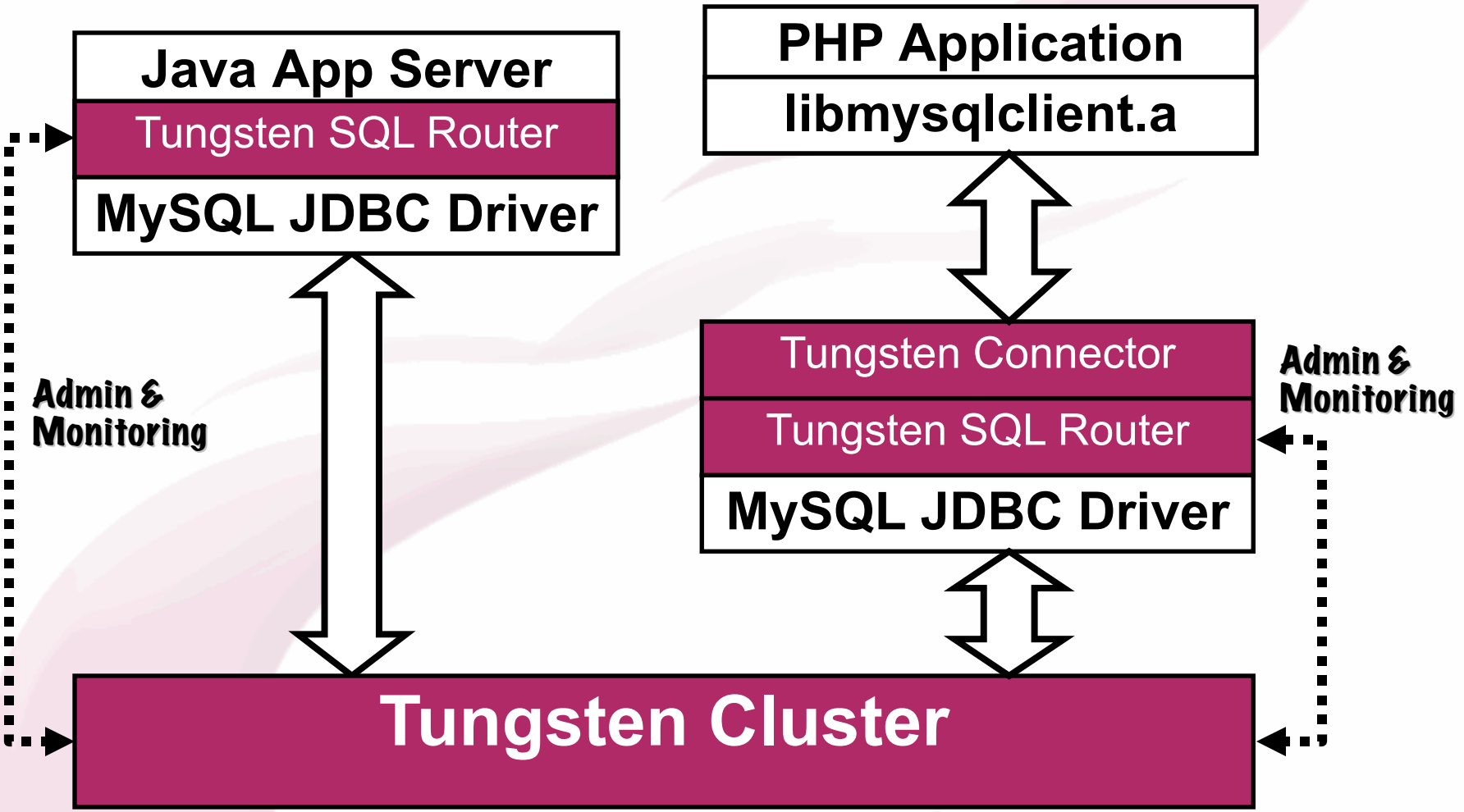
Distributed Rule-Based Management



Open Replicator To Manage Non-Tungsten Replication



SQL Routing



What Does This Get Us?

- **15 minute installation**
- **Single commands to:**
  - View cluster status
  - Backup a server
  - Restore a server
  - Verify data across copies
  - Confirm liveness of replication
  - Switch servers safely for maintenance
  - Failover a dead server to most current replica
- **Automatic discovery of new database replicas**
- **Automatic failover when databases fail**
- **Simple procedures for provisioning**

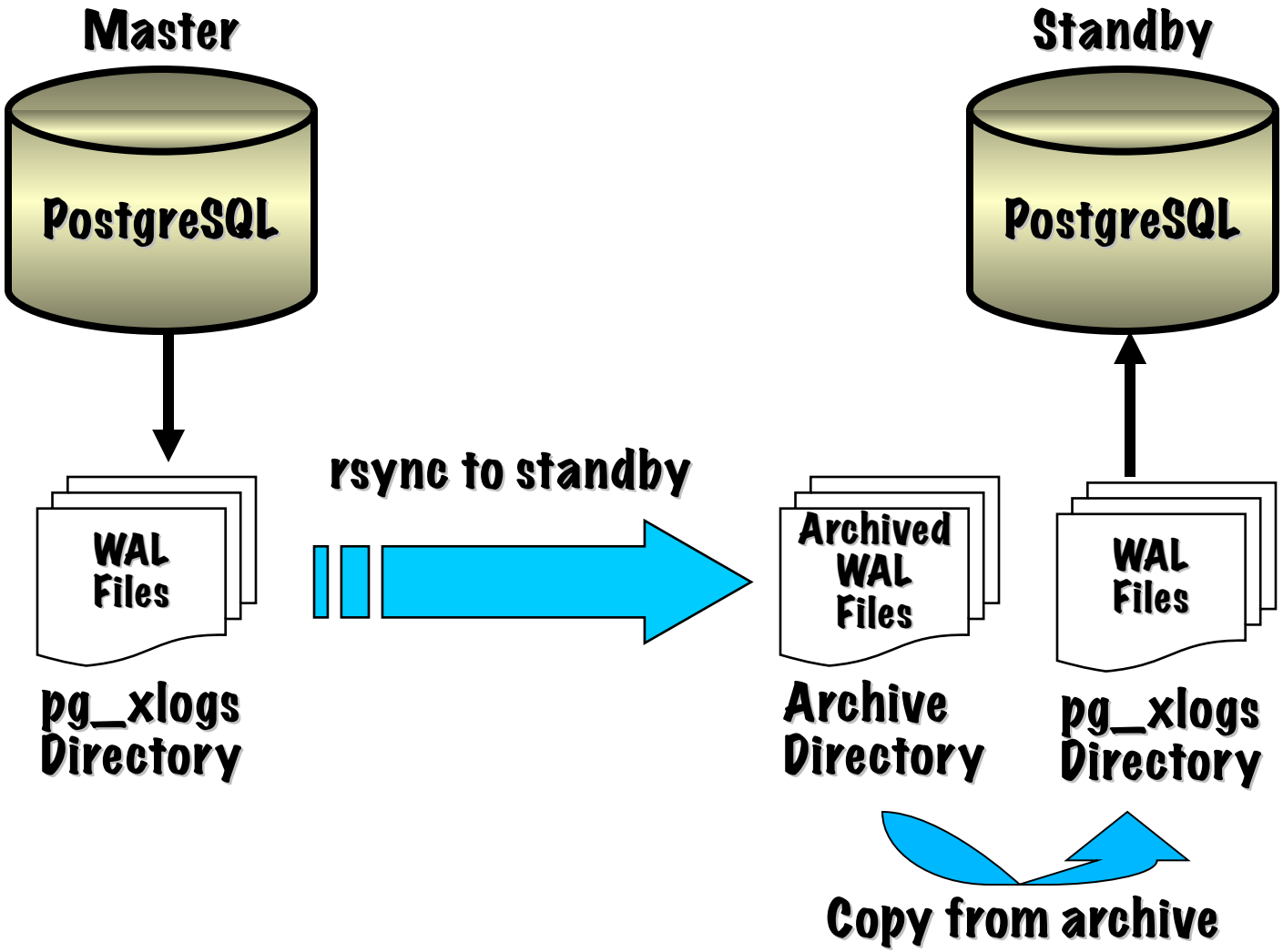
Adapting Tungsten to PostgreSQL

continuent

Moving Tungsten to PostgreSQL

- **Problem: We can't read PostgreSQL logs (yet)**
- **Solution: Manage Warm Standby/PITR to replicate data to standby DBMS**
  - Good basic availability/fast failover
  - Once hot standby works this looks pretty good!
  - Does not cover maintenance especially well
- **Solution: Manage Londiste to replicate to active replicas**
  - Covers maintenance and read scaling

Warm Standby Implementation



Setting Up Warm Standby (Old Way)

- **Configure master postgresql.conf and reboot**

```
archive_mode = on
archive_command ='rsync -cz $1 ${STANDBY}:${PGHOME}/archive/$2
  %p %f'
archive_timeout = 60
```

- **Set up standby recovery.conf**

```
restore_command ='pg_standby -c -d -k 96 -r 1 -s 30 -w 0 -t
  ${PGDATA}/trigger.dat ${PGHOME}/archive %f %p %r'
```

- **Provision standby**

```
psql# select pg_switch_xlog();
psql# select pg_xlogfile_name(pg_start_backup('base_backup'));
rsync -cva --inplace --exclude=*pg_xlog* ${PGHOME}/
  $STANDBY:$PGHOME/archive
psql# select pg_xlogfile_name(pg_stop_backup());
```

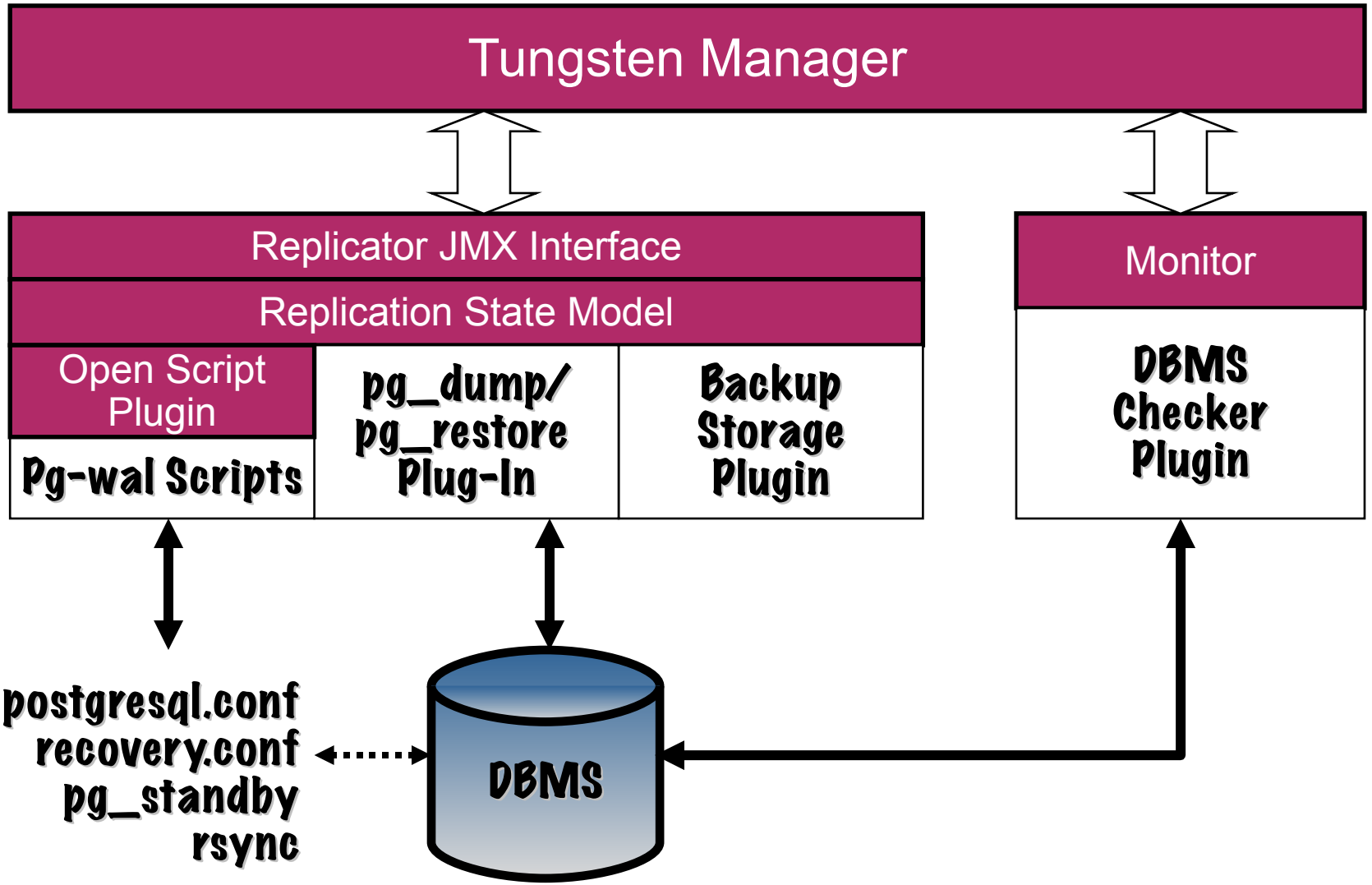
- **Start standby, recovery starts**
- **Touch ${PGDATA}/trigger.dat to fail over**

Warm Standby Caveats

- Warm standby helps with availability, not scaling
- Warm standby can lose data on unplanned failover!
- Master recovery requires re-provisioning
- Set-up/management is harder than it looks
- Monitoring is critical
- Cannot open standby before failover
- Need to ensure all logs are read before failover

Despite all the caveats it's a great feature!!

Tungsten Warm Standby Implementation



What Does This Get Us?

- **Easy setup of warm standby**
- **Single commands to:**
  - View cluster status, including replication stats
  - Backup a server
  - Restore a server
  - Provision a server
  - Verify data across copies
  - *Confirm liveness of replication*
  - *Switch servers safely for maintenance*
  - *Failover a dead server to most current replica*
- **Automatic discovery of databases**
- **Automatic failover**

Where Do We Go Next?

- **Fill in warm standby management features**
  - Detailed WAL setup features
  - Slave backup
  - Monitoring
  - Notifications on failures/thresholds
  - Ease of recovery
  - Hot Standby/Log Streaming
- **Implement Londiste support for live replicas**
- **Read PostgreSQL logs directly**

Plus a host of other useful features like floating IP support

Summary and Questions

Summary

- Changing technology and user needs are reshaping clustering
- Continuent Tungsten clusters solve new needs more effectively than other clustering approaches
- Check out what we are doing and provide feedback

Contact Information

HQ and Americas
560 S. Winchester Blvd., Suite 500
San Jose, CA 95128
Tel (866) 998-3642
Fax (408) 668-1009

EMEA and APAC
Lars Sonckin kaari 16
02600 Espoo, Finland
Tel +358 50 517 9059
Fax +358 9 863 0060

e-mail: robert dot hodges at continuent dot com

Continuent Web Site:
http://www.continuent.com

2ndQuadrant Web Site:
http://www.2ndquadrant.com