

Companies deploy Pivotal Greenplum for large-scale MPP (Massively Parallel Processing) data analytics. However, there are still performance challenges from inefficient application interaction. They include:

- Transactional overhead: Many singleton DML operations result in commit overhead
- Scaling connections: High user counts consume backend resources
- A scalable and simple way to manage data access
- Ensuring business continuity for your database

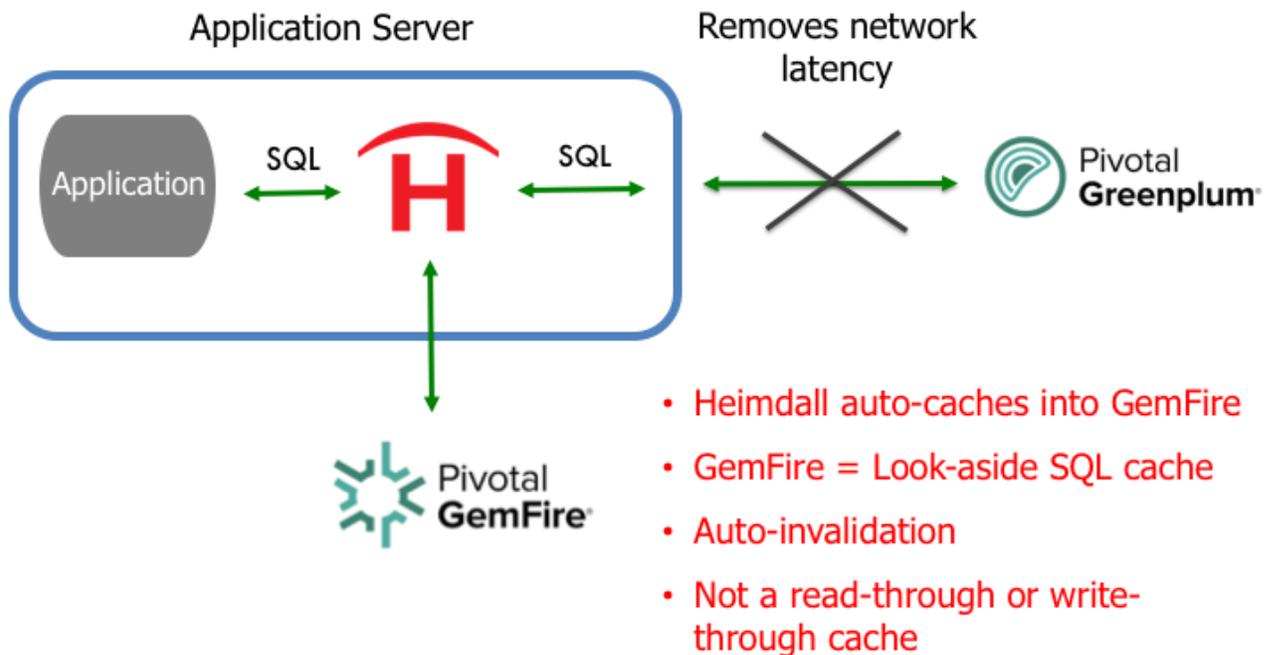
These challenges can be solved by modifying the application. However, the ideal solution requires no code changes. Heimdall Data is a database proxy providing Pivotal:

- Improved SQL Performance:
 - Automated Query Results caching
 - Batching of singleton DML operations
- Improved Database Scale: Connection pooling to support large scale end users
- Improved Security: Authentication and Authorization with Active Directory
- Improved Reliability: Automated Greenplum Master-write Failover

Automated Query Results Caching

The fastest performing query is the one that does not have to be executed. Heimdall's provides a look-aside result caching solution. You choose the cache (e.g. Redis, ElastiCache, GemFire) and the proxy provides the caching and invalidation logic. With one-

click our learning algorithms determine which query results are cached and when to invalidate. Best of all, the Heimdall proxy deployment requires zero code changes.



Connection Pooling / Multiplexing

Like Heimdall auto-caching, connection pooling and multiplexing can save you Greenplum costs and improve processing power.

It is important to maintain the correct ratio between application and database resources. Each connection to Greenplum takes valuable resources that can be better utilized for active queries. Therefore, it is desirable to reduce the total number of connections.

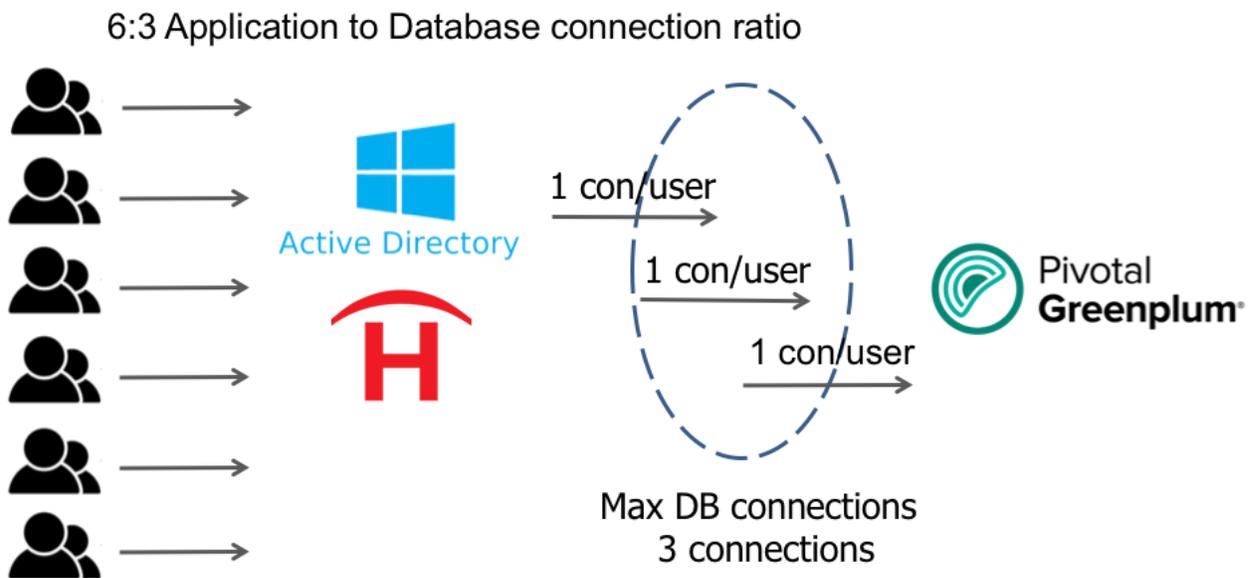
There are two Heimdall techniques that reduce connection overhead:

- Heimdall uniquely supports true multi-user connection pooling, with both per-user and global connection limits, to manage the number of active connections on Greenplum. Connection pooling allows multiple client connections to be associated to a Greenplum connection. When a connection is established from a client, an

existing connection is picked from the existing pool, instead of a new connection being established. While the client connection is open, this mapping remains. When the client-side connection is closed, the Greenplum connection is kept open to reduce CPU and latency for backend connection establishment lowering Greenplum overhead.

- Connection multiplexing is an extension of pooling. Instead of associating an entire client connection with the back-end, connection multiplexing dispatches individual queries or transactions to connections in the connection pool. As client connections are often idle, multiplexing allows for more “active” client connections to Greenplum. The net result is 1) Lower total memory overhead on Greenplum, and 2) More active queries processed at one time, and 3) Reduced Greenplum costs. Benefits will vary based on the application workload.

— Connection Pooling / Multiplexing



- Two-tier pooling configuration: 1) Per user and 2) Aggregate users
- Heimdall delivers the connections for only active concurrent queries

Authentication, Authorization, and Auditing

Most Greenplum deployments support a limited number of users to the database. Users are typically managed manually by the DBA. The Heimdall proxy supports authentication, authorization and auditing of database access:

- a) Single sign-on via Active Directory and LDAP;
- b) Group extraction and role synchronization from LDAP into Postgres for authorization;
- c) Logging of queries for auditing.

Batching Processing

The Heimdall proxy improves database write performance by batching DML operations against a table and putting them onto a separate connection under a single transaction. Batching DML operations result in:

- Improved application response times due to fewer commits
- Improved DML scale

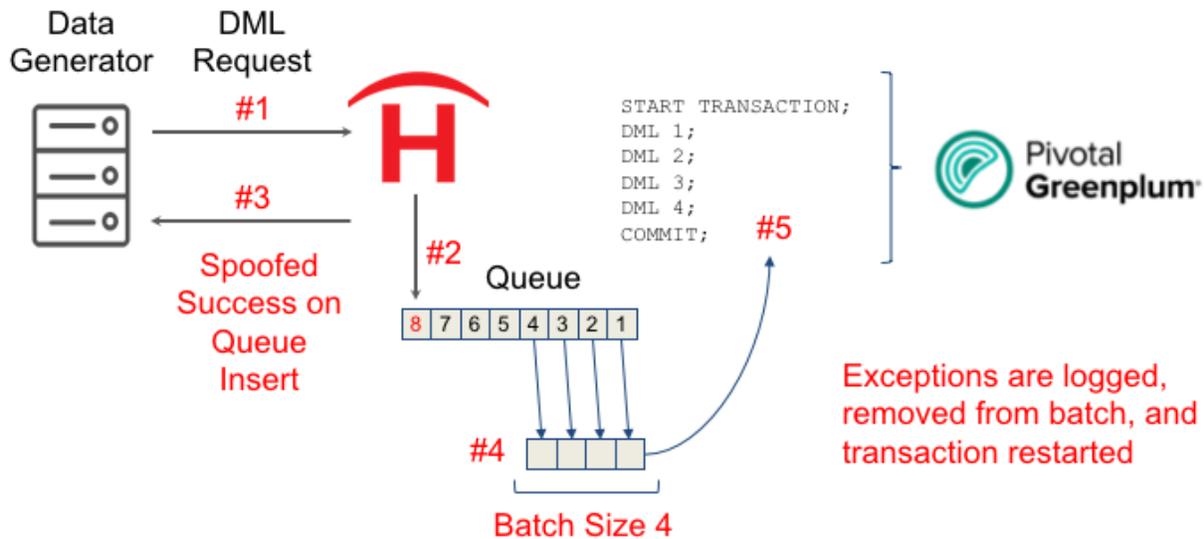
Ideal use case for batch processing:

Insert/update/delete a large amount of data at once on a thread. Heimdall can process it all at once much faster than if individual queries outside of a transaction were completed.

Not so ideal use case for batch processing:

If there are concurrent writes and reads against the same table, on the same thread, as everything will just block until the DML operation is completed anyway.

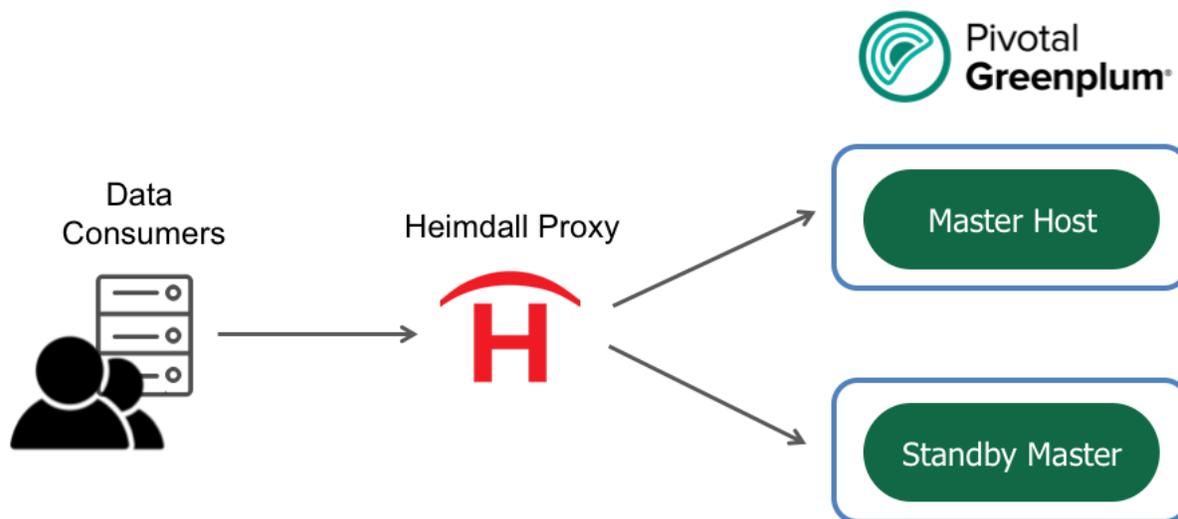
If an exception occurs in the transaction, Heimdall remove the query that caused the exception, report the exception in the logs, and reprocess the batch without the query in place. See the dataflow diagram below:



Automated Greenplum Master-write Failover

Heimdall's transparent failover is a must-have for any Enterprise desiring Greenplum to be always-on. Heimdall detects a write master failure and seamlessly failovers over to the standby. How are we different? Upon a failure, Heimdall queues up the current connection and transparently fails over to the backend master standby; our proxy persists the application connection. This helps remove application errors and exceptions upon a failover.

Heimdall Data for Pivotal Greenplum



Heimdall Data is a Pivotal Advanced Technology partner. Our database proxy provides improved SQL performance and reliability Pivotal Greenplum and Pivotal Postgres.

The Heimdall architecture was designed for ease of deployment without the need to modify the application or database. Configuration changes are updated at runtime without restarting the application. The net result is a platform that can be updated with 100% application uptime.

For more information about how Heimdall Data can help improve your SQL environment, contact us at info@heimdalldata.com.