

Fireswarm and Avalerion

Paul Reeves
IBPhoenix

mail: preeves at ibphoenix.com

Firebird Conference 2019

Berlin, 17-19 October



IBSurgeon



About the speaker

Paul

I work for IBPhoenix providing technical support.

I also maintain the windows installer for Firebird and do the Windows builds.

Part One – Avalerion

Avalerion or alerion is a term for a heraldic bird. Historically, it referred to the regular heraldic eagle. Later heralds used the term alerion to refer to "baby eagles" or "eaglets".

To differentiate them from mature Eagles, Alerions were shown as an Eagle Displayed Inverted without a beak or claws (disarmed).

(Wikipedia)

Why replicate data

- Report generation
- Semi-offline Analytical processing
- Hot backup
- Automatic Failover
- Multi-way replication can improved response time, especially if network latency is a problem.

Before Avalerion

- There was and still is IBReplicator
- Origins of IBReplicator go back to InterBase days.
- Supports more or less all and any replication configuration
 - one-way
 - n-way (multi-master)
 - heterogeneous (source and target with different schemas).
- Tested in the field
- Still supported.
- Now at v5.0
- Works with all versions of Firebird.

Limitations of IBReplicator

- Requires embedding triggers in every table to be replicated.
- Because triggers capture all data changes there is a performance penalty.
- Also there is poorer performance due to use of long running snapshot transactions to transfer data.

Enter Avalerion

- Firebird with a change data capture plugin (CDC).
- It is based on Firebird 3
- It works with ODS 12 databases (with a minor change to the ODS)
- Is just one component of the replication architecture.
- So when we talk about Avalerion we imply the whole architecture but strictly speaking Avalerion is just one element.

About the Replication architecture

- Avalerion talks to...
- The CDC plug-in which talks to...
- The replication server which talks to...
- The target databases.

Key Features

- Normal Firebird server with some improvements.
- It uses a modified engine12 library.
- It enables the change data capture module to be implemented as a plug-in.
- Target database can be any supported server. This means any version of Firebird, as well as InterBase and Oracle.
- Other dbms engines may be supported in the future.

New engine features

- Replication is configured at the database level.
- For finer grained support CREATE TABLE has also been enhanced...

```
CREATE TABLE mytable (  
... field definitions here...  
)
```

```
[ENABLE|DISABLE REPLICATION]
```

(enabled is on by default.)

- Sequences can also be similarly controlled.

Transaction level control

- At the API level there is a flag

`isc_tpb_no_replication`

which can be used in the Transaction Parameter Buffer (TPB)

If set it will disable all replication for the duration of the transaction.

- This can also be set with the following SQL:

`SET TRANSACTION READ COMMITTED NO REPLICATION`

Configuration of Avalerion

- Two files to edit:
 - Databases.conf
 - The plug-in config

Avalerion Settings in databases.conf

These parameters are available at database level. For example:

```
mydb = /srv/data/mydb.fdb
{
  ReplicationPlugin = ibpr_cdc
  ReplicationEnabled = true
}
```

Setting up replication at the database level

- Database design
- Configuration of the replication service

A word about general database design for replication

- Primary keys must be guaranteed to create unique records across all servers
 - Either by server id + sequence
 - Or UUID

This is a general rule – perhaps not necessary for one-way replication.

Configuring the CDC Plug-in

- # Directory to write log files
OutputDirectory=/var/spool/avalerion
- # License key
LicenseKey=""

Final config step – the configuration db

- There is a windows gui for this
- And it is easily scriptable

Example of a simple config script

```
CREATE CONFIGURATION localhost:/opt/firebird/config_bmsql_smoketest.fdb \  
  TYPE Firebird USER sysdba PASSWORD masterkey
```

```
MODIFY GLOBAL SET SCHEDULER on
```

```
MODIFY GLOBAL SET LOG /var/log/avalerion/bmsql_smoketest.log  
MODIFY GLOBAL SET LOG 3 3
```

```
ADD DATABASE OF TYPE File NUMBER 1 NAME bmsql_smoketest-source \  
  INBOUND /var/avalerion/data_segments \  
  AUTO REGISTRATION ENABLED
```

```
ADD DATABASE OF TYPE Firebird NUMBER 2 NAME bmsql_smoketest-target \  
  SERVER avalerion305-replica:bmsql_smoketest USER sysdba PASSWORD masterkey
```

```
ADD SCHEMA 'Schema 1' SOURCE 1 USER REPL ORDER 0
```

```
ADD TARGET DATABASE 2 USER sysdba PASSWORD masterkey ORDER 0
```

```
ADD SCHEDULE PERIODICALLY EVERY 15 SECONDS
```

Limitations

- Synchronous replication is not available - It slows down source database and cannot be used for high availability.

Part Two - FireSwarm

“ Swarm behaviour, or swarming, is a collective behaviour exhibited by entities ... arising from simple rules that are followed by individuals and does not involve any central co-ordination. “

(Wikipedia)

Why cluster servers

- Most / all the same reasons as above for replication
- Easy to add or remove nodes as demand changes
- Much greater flexibility

FireSwarm

- It is 'just a plug-in for Avalerion
- Currently plugs into current version of Avalerion based on FB3
- Will be available for FB4 in due course

FireSwarm Features

- A cluster consists of two or more servers
- Cluster architecture is Share-Nothing
- Sort of auto-configuring and self-repairing

Share-Nothing

The "share-nothing" architecture means that nodes share neither physical resources (memory, storage) nor data dependencies.

Architecture

- A cluster is made of of nodes
- A node is an Avalerion server with the FireSwarm CDC plug-in and the distribution agent which handles replication traffic to the other nodes.
- The CDC captures data changes and signals to the distribution agent.
- The distribution agent propagates changes to other nodes.
- The distribution agent also receives changes from other agents and forwards them to the CDC module.

Auto-configuring and self repairing

- There is nothing to configure at the db level. FireSwarm copies everything.
- If a db does not exist when a node comes on line the database will be copied automatically from a neighbouring node.
- If a minor problem appears the distribution agent will try to resolve it.
- If it fails it will take the node off-line.

Limitations

- Changes to system tables are not replicated
- There is not currently a mechanism to guarantee synchronized propagation of metadata changes.
- Temporary tables are not replicated.

Requirements

- On the hardware level, none, apart from the obvious – you do need multiple servers to form a cluster.
- Two nodes would be the absolute minimum. In practice at least four are required for serious use.
- At the database level tables must guarantee globally unique primary keys.
- Unique constraints wont/don't work correctly across multiple servers.

Replication Modes

- Synchronous
- Asynchronous
- Mixed

Source code is available

- Currently as a tar ball from our web-site.
- And most has already been contributed back to the Firebird project
 - (but not all used.)
- The CDC plug-ins remain closed source as do the replication servers.

Questions?