Data Replication and Synchronization Tool

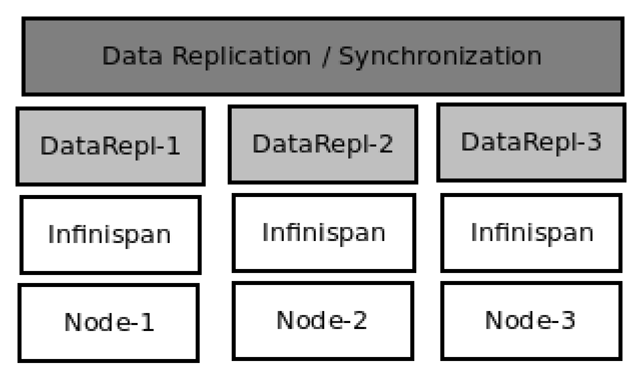
Ashish Sharma
Pradeeban Kathiravelu

Introduction

Motivation

Solution Architecture

- Users create, share, and update replica sets from a data source.
- Infinispan In-Memory Data Grid (version 6.0.2) to store the replica sets.



Data Replication / Synchronization

DataRepl-1 DataRepl-2 DataRepl-3

Infinispan Infinispan Infinispan

Node-1 Node-2 Node-3

Fig 1. Deployment Architecture

Execution Flow

- Publisher-Consumer API to consume the replica sets and Data Provider API to communicate with the data source.

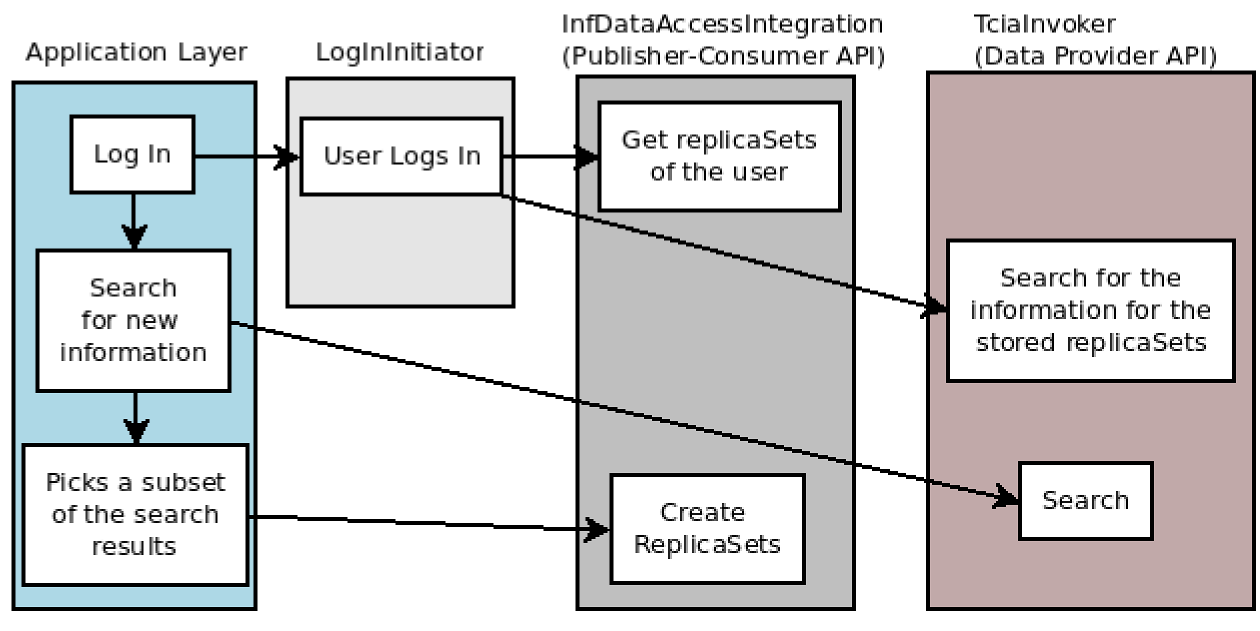

Application Layer
LogInInitiator
InfDataAccessIntegration (Publisher-Consumer API)
TciaInvoker (Data Provider API)
Log In
User Logs In
Get replicaSets of the user
Search for new information
Search for the information for the stored replicaSets
Picks a subset of the search results
Create ReplicaSets
Search

Fig 2. Execution Flow

Design

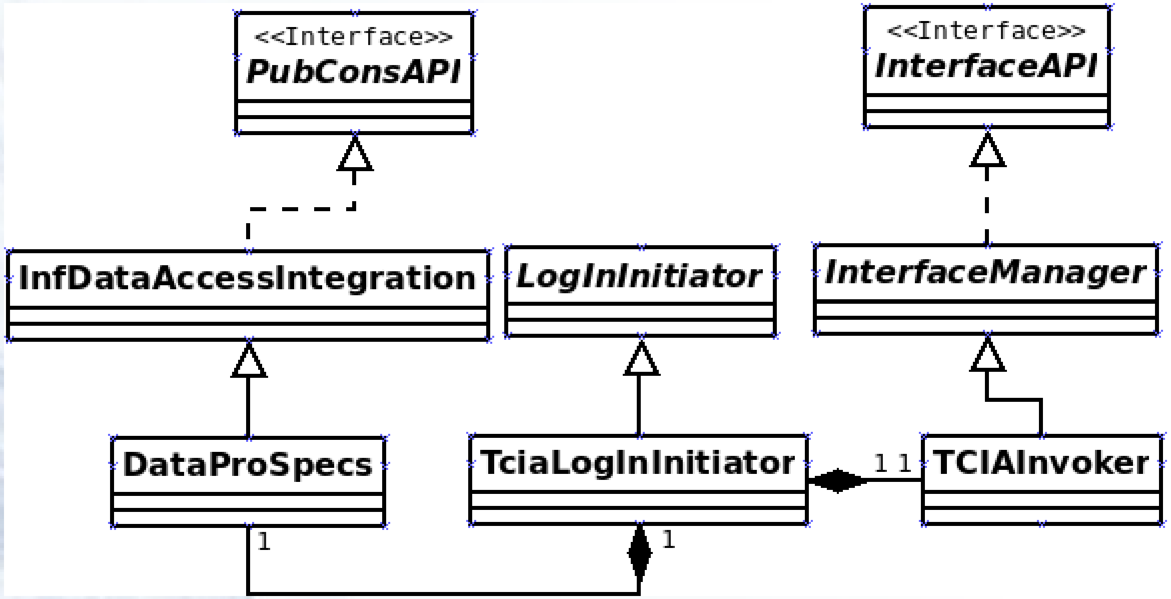


Fig 3. Back-end Class Hierarchy

- DataProSpecs API
  - createReplicaSet
  - getReplicaSet
  - updateReplicaSet
  - duplicateReplicaSet
  - deleteReplicaSet

Extensibility

- Not tightly coupled to the technology.
  - Other data-grids
    - Hazelcast, Terracotta Big Memory, Oracle Coherence
  - Persistence
    - Integration to SQL or NoSQL solutions such as Mongo DB.
  - Data sources other than TCIA.

What Infinispan offers?

- High Performance and Scalability.
- Fault-tolerance
  - Multiple nodes with TCP-IP or Multicast based JGroups clustering configurations.
- Distributed Execution.
  - Optimized for single node as a local cache as well as a multiple-node execution.
- MapReduce Framework.

What Infinispan offers?

- High Performance and Scalable.
- Fault-tolerant
  - Multiple nodes with TCP-IP or Multicast based JGroups clustering configurations.

Thank you!

- Distributed Execution.
  - Optimized for single node as a local cache as well as a multiple-node execution.
- MapReduce Framework.

What Infinispan offers?

- High Performance and Scalable.
- Fault-tolerant
  - Multiple nodes with TCP-IP or Multicast based JGroups clustering configurations.

Thank you!

- Distributed Execution.
  - Optimized for single n ocal cache as well as a mul e execution.
- MapReduce Framework.