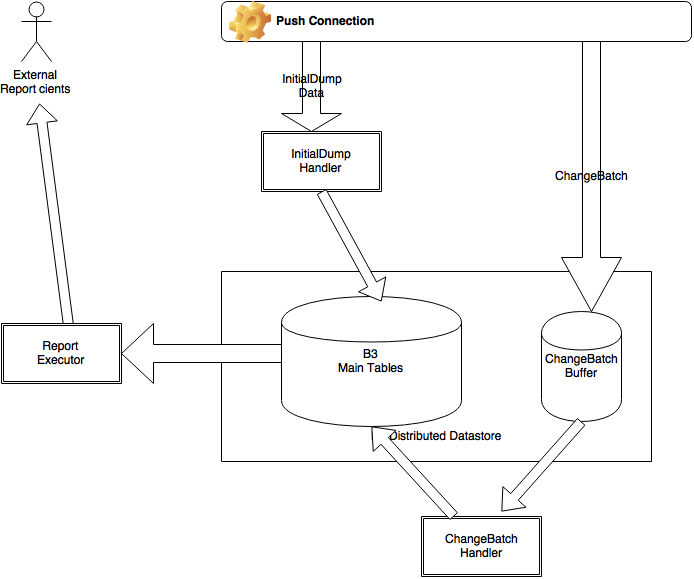
Project

Betbrain Bigdata (B3)

Technical Design

1. Overall architecture

The whole system is divided into subsystems:

1/ Distributed datastore (B3 datastore)

* The datastore contains all data from the current Sport Data Model
* Data is denormalized to support fast querying for reporting purposes
* Datastore comprises 4 main tables (currently), which are necessary for reporting queries. These tables are event, event\_info, outcome and betting\_offer.

2/ InitialDump handler

* This handler receives the initial dump data from Sport Engine (SE)’s push connection, organizes the data into 4 types of bigger data objects (object graphs with entities and linked entities). These 4 types of objects are B3Event, B3EventInfo, B3Outcome and B3BettingOffer.
* InitialDump handler is responsible to put B3 objects into B3 datastore

3/ ChangeBatch handler

The handler receives EntityChange data from SE’s push connection, propagate all the changes to B3 datastore.

4/ Report executor

* This performs appropriate data querying from B3 datastore and represents data in the report’s format.

II. Data-key design

* General design principles:

- Denormalization:

+ SE Event/EventInfo/Outcome/BettingOffer are mapped into B3 types of same names

+ Other entities are denormalized into these main entities

- Field mapping:

+ Each SE entity goes into one data column, in json format (one B3 record has multiple columns, each column is corresponding to one SE entity)

+ All IDs in one entity (entity ID, parent IDs, child IDs) are extracted, stored duplicated in its own column in a B3 record.

1/ Table Event

* Partitioned by: sportId, eventTypeId and hash code of eventId
* Keyed by: eventId

2/ Table EventInfo

* Partitioned by: sportId, eventTypeId, hash(eventId)
* Keyed by: eventId, eventInfoTypeId, eventInfoId

3/ Table Outcome

- Partitioned by: sportId, eventTypeId, hash(eventId)

* Keyed by: eventId, outcomeTypeId, outcomeId

4/ Table BettingOffer

* Partitioned by: sportId, eventTypeId, hash(eventId)
* Keyed by: eventId, outcomeTypeId, outcomeId, bettingOfferTypeId, bettingOfferId

III/ Secondary (supporting) tables

Beside the main tables, B3 uses 3 more tables for its internal purposes:

* Table entity: for fast querying of entity id and its linked entities’s IDs
* Table link: for fast querying of linked entities
* Table lookup: for locations of an entity in a main table.

IV/ Current progress

* Completed InitialDumpHandler: Initial dump data have been stored in B3 datastore
* Partial implementation of ChangeBatchHandler: EntityCreate has been handled, stored in B3 datastore
* Odds Detail Table report: able to query data in B3 with fast response time (in milliseconds)
* Data latency: not monitored yet, as ChangeBatchHandler is incomplete, but expected small, too.

V/ In next sprint

* Complete ChangeBatchHandler (Handling of entity updates and entity deletes)
* Deploy reporting service for testing