University of California Irvine
Programming Language Research Group

IoTCloud Version 2.0

Authors

September 29, 2016

1 Introduction

2 Approach

2.1 Records

Each record has the following information included in it:

- Machine ID of the device creating the record
- The vector clock using the largest clock values from each device it knows and its own largest clock value incremented by 1.
- Add a random salt (or nonce) for the encryption safety
- Data payload
- HMAC of the record.

2.1.1 Types of Payloads

The different types of record payloads are:

- Delete notifications

  - Contains the HMAC of records that were deleted by devices.
  - Generated when a device deletes a key from the end of one of the device queues.

- Commit notifications
  - Contains list of transactions that are committed in order of commit and the current key-value pair for that key.
  - Generated by the arbitrator of a key and only the for that key (1 arbitrator per key).
  -

- Abort notifications
  - Contains a transaction ID of an aborted transaction and the machine ID of the device that created that transaction.

- Data structure re-size notifications
  - Contains new size of data structure (number of record allowed in the data structure).
  - Causes old data Structure re-size notification to no longer be live.

- Server sequence number for a specific record notifications
  - Contains a record HMAC and the server sequence number for that record

- Transactions
  - Contains:
    * Transaction ID
    * A guard condition that can be evaluated
    * A set of key-value pairs that are to be updated if the guard condition is met.

2.2 **Updates**

Updates take place as follows:

1. A device pulls the latest version of the data structure. If the device cannot pull the latest version because of network connectivity or some other issues then that device will just work using the local copy of the data structure it has.

2. The device makes a record as follows:
   a) Adds its machine ID.

b) Creates a vector clock using the largest clock values from each device it knows and its own largest clock value incremented by 1.

c) Add a random salt (or nonce) for the encryption safety

d) Fill the record data section with the transactions, key-value pairs, ext.

e) Fill the remainder of the data section with rescued key-value pairs, transactions, ext.

f) Pad the record to be the same size for all records.

g) Calculate the HMAC of the record and add that to the record

h) Encrypt the record

3. Send the record to the server for insertion into the device's queue.

4. Wait for response from server stating the new records (the one just sent) server sequence number. Save this server sequence number for when creating the next record.

5.

2.3 **Updates**

2.4 **Deletions**

2.5 **Checking the Graph**

Checking the data structure for consistency is done as follows:

1. Verify that each record in the data structure has an HMAC that matches the data in the record.

2. Verify that there are at least as many records in the data structure as stated in the largest data structure size record.

3. Make sure that for each device queue the difference between the vector clock value of the device queues clock is at most 1 between 2 consecutive messages.

4. Verify that no currently live data Structure re-size notification is smaller than the last known data structure size. Data structure can only grow in size.

2.6 **Live Status**

Live Status of entries:

1. Key-Value Entry is dead if either:
   a) there is a newer key-value pair

b) it is incomplete.

2. If there are n devices in the system then there are n separate queues