**Schema Modification Proposal**

This modification proposal focuses more on permanent data schema, since the string basic data type in Redis is fairly good for temporary data since temporary data can be frequent I/O without any pattern. String data type is small, low I/O cost and versatile, hence I agree with using String datatype for temporary datas. However, as for permanent data, we need to consider the integrity, memory cost and also data fragment in long term. To solve issues above, I propose few changes to permanent data schema as solution.

**User information & sensor information (permanent**):

Permanent data is separated in 3 sections: user information, sensor information and user-sensor association information. The original schema is based mainly Redis string data type and saved as pairs. There are few issues. For example, for user information, it is difficult to tell total profile validity, such as missing information. The only way to retrieve the list of all profiles is through “keys” command which involves numerous keys retrieval which is i/o costly. Hence I propose using hash-list method for user information and sensor information. Refer to excel for specific schema stable.

**User-Sensor association information (permanent):**

Since this table is also permanent, we do not want large amount of keys. Using String for association information can create quit large amount of keys. For example, if we have n users and m sensors with k different configuration, we will create n\*m\*k number of keys, which is difficult to track and it is special costly. To better resolve it, I propose using 2 lists to track mapping information from user to sensor and sensor to user. Also use hash to track specific configuration between user and sensor. With the same previous example, the special cost would be 2+n\*m at maximum.

**Measured air raw data (temporary**):

Sorted set is a great optional. However, we can take better advantage of new data type STREAM. STREAM has built in timestamp, which we do not need to worry about lock sync cross all sensor. Instead, time will be synchronized without Redis DB clock internally. The STREAM also has built in blocking operation, which is perfect for listening and data streaming. This is very trivial change since sorted set and stream shares most properties.

Here is a quick view on updated permanent data schema:

