Redis

This is basically Igor's email from Mon 10/19/2015 5:58 PM

I have built RPMs & deployed Python client version 2.10.3 (the latest one as of today) for Redis in the external package area. This client-server software allows for an effective inter-process communication and data caching in the distributed environment. There are many interesting applications for this. Apart from using it for inter-process communication, the service is also used to temporarily cache data which are expensive to calculate or fetch from persistent sources (such as databases, etc.). Redis stores its data in memory. My specific goal is to use it as as a storage for the Monitoring data (plots, histograms, etc.). See a cluster of JIRA tickets here:
https://jira.slac.stanford.edu/browse/PSDH-10
You may find more info on Redis from:
http://redis.io/
The Python API is explained here:
https://pypi.python.org/pypi/redis
I would like to include the following tag to the next release builds (including 'ana' and 'dm'):
  redis V00-00-01
Note that Redis requires a server. I set up the one for testing at psdb3. Here is the simplest test for it (from a test release which has the above mentioned front-end package):

```
 In [1]: import redis
 In [2]: r = redis.StrictRedis(host='psdb3')
 In [3]: r.set('NameX', 'Igor Gaponenko')
 Out[3]: True
 In [4]: r.get('NameX')
 Out[4]: 'Igor Gaponenko'
```

and this is an example of how to store NumPy arrays in that database:

```
 In [26]: import numpy
 In [27]: d = numpy.empty([4096,4096])

 In [28]: r.set('numpy_1MB',d.tostring())
 Out[29]: True
 In [30]: d_copy = r.get('numpy_1MB')
 In [31]: len(d_copy)
 Out[32]: 134217728
 In [33]: d1 = numpy.fromstring(d_copy)
 In [34]: d1.dtype
 Out[35]: dtype('float64')
 In [36]: d1.size
 Out[37]: 16777216
```

Note that the performance of the service which is presently limited by the network setup of the server node (1 GbpE). So teh best one should expect would be 100 MB/s.. For the local operations (when the client runs on the same node where Redis is set up) I observed up to 1 GB/s. I presume a very similar limit would apply of the Redis server and its clients would be on the 10 GbpE network.
The maximum size of objects to be stored in this database is limited by 0.5 GB. So, it's good for images, histograms, etc.

And since this is the key/value store in which there is a single set (space) of keys (strings) then one should invent unique keys to prevent collisions. More information is found in the official documentation portal:
http://redis.io/
Regards
Igor