Redis for OpenVMS I64 and Alpha (BETA)

February 2018

1.   Introduction

Thank your for your interest in this port of Redis to OpenVMS. The current release of Redis for OpenVMS is based on the Redis 4.0.0 distribution.

Redis (https://redis.io/) is an Open Source (BSD licensed), in-memory data structure store, used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, and geospatial indexes with radius queries. Redis has built-in replication facilities, Lua scripting, LRU eviction of cache entries, transactions and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster.

This OpenVMS port of Redis includes all in-memory/caching functionality, Lua scripting, and publish/subscribe functionality provided by the open source release. In addition, this OpenVMS port includes a language-agnostic API that makes it straightforward to write Redis applications using 3GL languages such as COBOL and FORTRAN. The port currently does not provide some functionality; specifically it does not currently include on-disk persistence.

Additional information about Redis can be found at https://redis.io/.

2.   Acknowledgements

VMS Software Inc. would like to acknowledge the work of the Redis development team for their ongoing efforts in developing and supporting this software.

3.   What's new in this release

For a detailed description of the features and bug fixes included in this release, please read https://raw.githubusercontent.com/antirez/redis/4.0/00-RELEASENOTES.

4.   Requirements

The kit you are receiving has been compiled and built using the operating system and compiler versions listed below. While it is highly likely that you will have no problems installing and using the kit on systems running higher versions of the operating system or products listed, we cannot say for sure that you will be so lucky if your system is running older versions.

- OpenVMS 8.4-1H1 I64; OpenVMS 8.4-2L1 Alpha

- HP TCP/IP Services V5.7

  It has not been verified whether the kit works with the MultiNet TCP/IP stack, but there is a good chance that it will.

- C compiler - HP C V7.3-018

Required for application development only. Application development in other languages available on OpenVMS such as COBOL and FORTRAN is also supported via a generic (language-agnostic) API.

In addition to the above requirements, it is recommended that the software is installed on an ODS-5-enabled file system. It is additionally assumed that the reader has a good knowledge of OpenVMS and of software development in the OpenVMS environment.

5.   Recommended reading

Before using Redis, it is recommended that users review the documentation available at https://redis.io/documentation in order to better understand how to configure and use the Redis server, and how to implement applications that use Redis.

6.   Installing the kit

The kit is provided as an OpenVMS PCSI kit (`VSI-I64VMS-REDIS-V0400-0-1.PCSI` for OpenVMS Integrity; `VSI-AXPVMS-REDIS-V0400-0-1.PCSI` for OpenVMS Alpha) that can be installed by a suitably privileged user using the following command:

```
$ PRODUCT INSTALL REDIS
```

The installation will then proceed as follows (output may differ slightly from that shown depending on operating system platform and other factors):

```
Performing product kit validation of signed kits ...

The following product has been selected:
    VSI I64VMS REDIS V4.0-0                Layered Product

Do you want to continue? [YES]

Configuration phase starting ...

You will be asked to choose options, if any, for each selected
product and for any products that may be installed to satisfy
software dependency requirements.

Configuring VSI I64VMS REDIS V4.0-0: Redis for OpenVMS is based on
Redis Version 4.0.0

    © Copyright 2017 VMS Software Inc.

    VSI Software Inc.

* This product does not have any configuration options.

Execution phase starting ...

The following product will be installed to destination:
    VSI I64VMS REDIS V4.0-0                DISK$I64SYS:[VMS$COMMON.]

Portion done: 0%...40%...50%...60%...90%...100%

The following product has been installed:
    VSI I64VMS REDIS V4.0-0                Layered Product
```

```
VSI I64VMS REDIS V4.0-0: Redis for OpenVMS is based on Redis Version
4.0.0

    Post-installation tasks are required.

    To start Redis at system boot time, add the following lines to
    SYS$MANAGER:SYSTARTUP_VMS.COM:

        $ file := SYS$STARTUP:REDIS$STARTUP.COM
        $ if f$search("''file'") .nes. "" then @'file'

    To shutdown REDIS at system shutdown, add the following lines
    to SYS$MANAGER:SYSHUTDWN.COM:

        $ file := SYS$STARTUP:REDIS$SHUTDOWN.COM
        $ if f$search("''file'") .nes. "" then @'file'
```

6.1.　　Post-installation steps

After the installation has successfully completed, include the commands displayed at the end of the installation procedure into SYSTARTUP_VMS.COM and SYSHUTDWN.COM to ensure that the Redis server process is started and stopped when OpenVMS is booted and shutdown.

Before attempting to start the Redis server, be sure to edit the supplied configuration file as appropriate for your specific requirements. The configuration file supplied with the kit (redis$root:[conf]redis.conf) may be used as a starting point, and may be adequate for many usage scenarios. Note that the configuration file is intended to be self-documenting, and includes in comments a detailed description of the various parameters that can be specified and the values they can take.

6.2.　　Privileges and quotas

The privileges TMPMBX, NETMBX, BYPASS, SYSPRV, and DETACH are required in order to run the Redis start-up and shutdown scripts, and the Redis process (run as a detached process) will inherit the default privileges for the username under which it is started.

The Redis process can require considerable resources in order to operate efficiently, depending on specific workload requirements. The following quotas should be adequate for most purposes; however resource usage should be carefully monitored, and quotas adjusted as necessary.

```
Maxjobs:         0  Fillm:       512  Bytlm:        128000
Maxacctjobs:     0  Shrfillm:      0  Pbytlm:            0
Maxdetach:       0  BIOlm:       150  JTquota:        4096
Prclm:          50  DIOlm:       150  WSdef:          4096
Prio:            4  ASTlm:       300  WSquo:          8192
Queprio:         4  TQElm:       100  WSextent:      16384
CPU:        (none)  Enqlm:      4000  Pgflquo:      256000
```

If Redis is expected to support large numbers of connections then it may also be necessary to increase FILLM and the CHANNELCNT system parameter (this parameter can usually be safely set to its maximum value of 65535). It may also be necessary to increase BYTLM and related quotas if the Redis server will be caching appreciable amounts of data (as noted elsewhere, maximum cache size is currently constrained to just under 1GB).

6.3.   Installing in an alternative location

By default the software will be installed in `SYS$SYSDEVICE:[VMS$COMMON]`. If you wish to install the software in an alternative location this can be achieved using the `/DESTINATION` qualifier with the `PRODUCT INSTALL` command to specify the desired location; however it is important to note that an additional manual step will then be required to complete the installation. Specifically, when an alternative destination is specified, the start-up and shutdown procedures (`REDIS$STARTUP.COM` and `REDIS$SHUTDOWN.COM`) will be placed into a subdirectory `[.SYS$STARTUP]` residing under the specified destination directory. If you wish to run these files from your standard `SYS$STARTUP` directory they will need to be copied from the destination subdirectory into your systems `SYS$STARTUP` directory.

7.   Example programs

The directory `REDIS$ROOT:[EXAMPLES]` contains several simple example programs in various languages (C, COBOL, and FORTRAN) that can be used to learn about the Redis client API or as a source of inspiration for the development of new applications. These examples can be compiled and linked using the provided build procedure (`EXAMPLES.COM`). Once built, the programs are simple to run, requiring the no command like arguments or input parameters. All of the programs will attempt to connect to a Redis server running on the current node; if you wish to communicate with a Redis server running on another node, the example code will need to be modified accordingly.

The kit also includes a simple benchmarking (`redis$root:[bin]redis-benchmark.exe`) that can be used to assess the performance of various Redis operations on the server in question. For a description of the various options that are supported by this utility refer to https://redis.io/topics/benchmarks.

8.   What's missing?

As noted previously, this beta release of Redis for OpenVMS includes all in-memory/caching functionality, Lua scripting, and publish/subscribe functionality. In addition, the port includes a language-agnostic API that makes it straightforward to write Redis applications using 3GL languages such as COBOL and FORTRAN. However, the port currently does not include on-disk persistence and related functions. It is hoped that this functionality will be supported in future releases.

9.   Known problems and restrictions

- This port of the Redis server and API components for OpenVMS use IEEE floating point format on both Alpha and Integrity. Application code that uses the API to store floating point data in Redis must therefore be compiled to use IEEE format or must convert between formats when sending and receiving floating point data. This restriction is most likely to affect applications running on OpenVMS Alpha, as IEEE is the default floating point format for programs on OpenVMS Integrity.

- Redis clustering and replication functionality has not been fully tested and it is possible that problems will be encountered in these areas. Please let us know about any problems that are encountered in this area (and indeed elsewhere).

- This port has been built using 32-bit pointers and according the maximum possible cache is just under 1GB of memory. It is intended that future releases will use 64-bit pointers and will therefore not have this restriction.