HPE GreenLake

Data sheet

HPE NonStop In-Memory Cache

A continuously available shared datastore on HPE NonStop

HPE NonStop In-Memory Cache is a fast data server that your mission-critical applications can leverage to share information across CPUs and nodes. It comes with the characteristic advantages of an HPE NonStop product.

Introduction

What is HPE NSIMC?
HPE NonStop In-Memory Cache (NSIMC) is an in-memory storage server that applications can utilize for storing information for faster access. HPE NSIMC can be accessed across the CPUs. Thus, it brings in the facility of a shared storage area for HPE NonStop applications.

About HPE NSIMC 7.0
HPE NSIMC is a value-added port of the open-source in-memory data structure store, Redis. Specifically, HPE NSIMC 7.0 is a port of Redis version 7.0.8. Redis is a popular NoSQL data structure store used by applications across many different domains. Its major use is in the cloud-computing area where it has proven capability to handle high transaction rates.

HPE NSIMC 7.0 offers the features of Redis 7.0 with HPE NonStop fundamentals.[1] It is inbuilt with continuous availability and data consistency that HPE NonStop users cannot do without.

HPE NSIMC works in a client-server model (Figure 1). Applications, being clients, invoke commands on the HPE NSIMC Server for storing, modifying, and retrieving data elements.

HPE NSIMC supports several data structures and is resilient to single points of failure. It is capable of automatic eviction of old or stale elements and has the ability to persist data elements in the disk. It also provides Redis master-replica data protection technique and Lua scripting facilities.

New features in HPE NSIMC 7.0
The following list is an amalgamated list of important new functionalities and enhancements released in the open-source Redis versions 5.x, 6.x, and 7.0.

New data type: Streams

RESP3

SSL protection

ACLv2

Client-side caching

New features in HPE NSIMC 4.0

- Least recently used (LRU) and least frequently used (LFU) algorithm
- Modules
- Cluster formation by client-side sharding
- Geospatial data type

[1] HPE NSIMC does not support Redis Cluster functionality. Instead, the HPE NonStop process pair (NSPP) mechanism helps ensure continuous availability of the cache server.

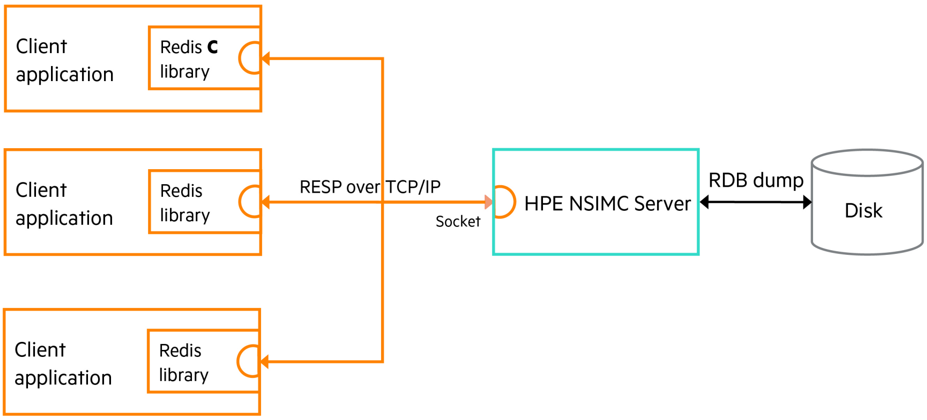

Figure 1. HPE NSIMC client-server model

Data structures[2]
A comprehensive set of data structures are supported by HPE NSIMC:

- Bitmaps
- Hashes
- HyperLogLogs
- Lists
- Sets
- Sorted sets
- Strings
- Geospatial
- Steams
- Bitfields

A variety of operations is possible with the data structures. Some examples are appending to a string; incrementing the value in a hash; pushing an element to a list; computing set intersection, union, and difference; or getting the member with the highest ranking in a sorted set. Redis also offers a concept of data type extensions using Lua scripting, Redis modules and Redis Stack. Readers of this data sheet are encouraged to study these very useful functionalities that can help implement many different use cases. Detailed information can be found in Redis documentation.

The RESP3 protocol

Clients and the HPE NSIMC Server communicate through the Redis Serialization Protocol (RESP). Clients send one or more commands to the HPE NSIMC Server using this protocol via a socket interface. It is an ASCII, human-readable, simple mechanism that utilizes TCP sockets for clients to send requests and obtain responses from the HPE NSIMC. It allows simple and bulk strings, arrays, integers, and positive and error responses to be transported. Commands that clients send to the HPE NSIMC Server and the responses from the server are packaged as arrays of bulk strings. See redis.io/clients for a comprehensive list of clients available for the Redis server.

HPE NSIMC 7.0 brings the new version of RESP, namely RESP3. This new version has improved semantics, supports new data types such as floating-point numbers and Boolean values. RESP is now binary safe while transporting error codes and strings. Get full details of the RESP3 protocol at github.com/redis/redis-specifications/blob/master/protocol/RESP3.md.

Continuous availability / fault tolerance

HPE NSIMC is implemented as an HPE NSPP. The backup process takes over when the primary process fails and becomes the new primary. There is no loss of the stored data elements. Figure 2 shows the HPE NSPP architecture. For data transfer, the primary uses file system-interprocess communication (FS-IPC) toward the backup.

The primary and backup data processes are able to sense the failure of the other. The backup assumes the role of the primary (when the primary fails), and it has complete access and control over the data set that was being managed by the primary. The HPE NSPP infrastructure in HPE NonStop facilitates that a new instance is automatically started up if the primary or a backup fails.

[2] redis.io/docs/data-types/

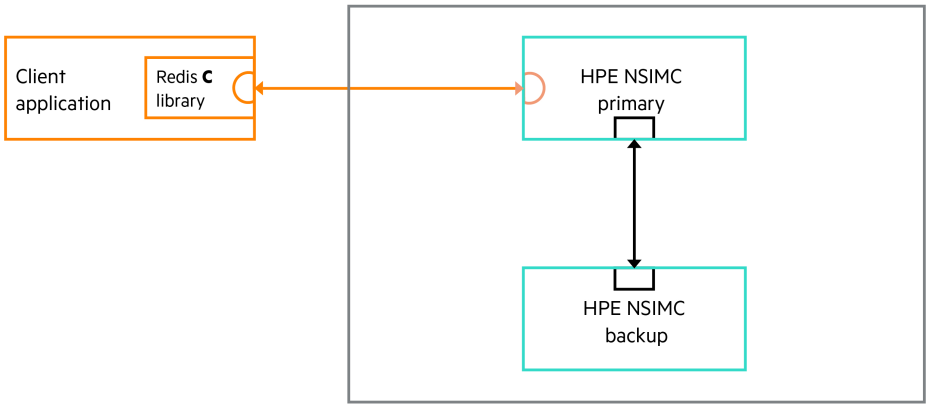

Figure 2. HPE NSIMC—HPE NSPP

HPE NSIMC has implemented performance improvements in the HPE NSPP mechanism. Primary only sends those commands from the clients that will result in modifications of the stored data set. For example, commands that read the data structure store are not sent to the backup. This results in better performance than otherwise. The primary collects the contents of the entire command and applies it on the backup at a faster rate.

Persistence of stored data elements

Redis provides two methods to persist the cache into nonvolatile (that is, disk) storage. One is Redis database (RDB) and the other is append-only file (AOF). RDB stores a snapshot of the cache contents at specified intervals. AOF method writes every write operation received by the Redis server into a log file. Both methods have their own pros and cons with respect to performance of the HPE NSIMC server—ease of reconstructing the cache contents after recovering from a loss and the accuracy of the backed-up data compared to the data in the volatile memory when the loss occurred. HPE NSIMC supports the RDB method. AOF is not supported. There is no data loss in HPE NSPP mode of operation when one instance of the HPE NSIMC server pair fails because the backup instance takes over seamlessly. Thus, AOF is not necessary.

Eviction of old/stale data elements

LRU and LFU algorithms

As time passes, the cache store may get filled up with data elements that have not been accessed for a long time. Such elements can be removed (evicted) to make space for newer elements. HPE NSIMC supports both the algorithms that Redis 7.0.8 provides to evict old/stale data elements from the cache. It provides an approximation of theoretical LRU and LFU algorithms. In both cases, you can set the maximum memory size beyond which, when a new data element is attempted to be stored, the algorithm tries to evict the LRU or the LFU element from the store. There are policies in both algorithms you can define to govern their behavior. For example, remove elements LRU, remove elements at random but only if they have an expiry time set, and so on. The number of samples to be picked up for applying the algorithms can also be configured. More the number of samples better the accuracy, but keep in mind that the cache server's response time toward the clients will increase.

Real-time messaging (pub-sub) between clients

Clients take the role of publishers and subscribers of messages. Publishers push messages into channels from which the subscribers receive them. Publishers can push messages into multiple channels. Similarly, subscribers can receive from multiple channels. The reply carrying the message to a subscriber is in array reply format of the RESP protocol. This operation does not store any data in the volatile cache. Publisher-submitted data disappears after subscribers consume it.

System requirements of the host

- HPE Integrity NonStop X, HPE Virtualized NonStop, or HPE Virtualized Converged NonStop servers
- HPE NonStop operating system release version L19.08 or later Open System Services (OSS) environment
- HPE NonStop TCP/IPv6 or IP-CIP for networking

Integration with HPE NonStop Measure

HPE NSIMC is integrated with HPE NonStop Measure. Each command invoked by a client on the HPE NSIMC server is pegged as an HPE NonStop Measure counter. The HPE NonStop Measure counter can be of ACCUM[3] or QUEUE[4] type. These counters prove very helpful in analyzing statistics such as

- Number of times a particular Redis command was issued by the client application to the HPE NSIMC server
- How much time, on average, elapses for running a command
- How much time a command spends in the HPE NSIMC Server queue before getting served

In addition to the Redis commands being counted, there is another counter command, which measures the overall total run time for all the commands that were sent by a client in one message. Users can view and analyze the latency in serving the requests and the length of the queue where the requests from different clients wait for implementation. This feature of HPE NonStop Measure counters can be turned on or off.

[3, 4] To know about ACCUM and QUEUE counters, see HPE NonStop Measure Reference Manual at hpe.com/info/nonstop-ldocs.

Transactions in HPE NSIMC

The client can send multiple commands serially to the HPE NSIMC Server, which will not be run but queued. The client can then ask the server to run all the queued commands in one atomic operation. Either all commands are run or nothing is run. A client can also request discarding of a transaction in the middle of sending the commands to be queued or just before running the transaction. There are possibilities like commands getting rejected at the queuing time itself, commands failing at the time of implementation, and so on. The HPE NSIMC Server handles these scenarios such that there is no inconsistency in the data set stored in the cache. Note that HPE NSIMC transactions are not same as HPE NonStop Transaction Management Framework (TMF) transactions.

Lua scripting

HPE NSIMC provides commands that can run scripts written in Lua version 5.1 from within the context of the HPE NSIMC Server. HPE NSIMC converts Lua data types and error responses to Redis types. HPE NSIMC uses the Lua interpreter to run the scripts. Running of a script is atomic. While a script is being run, no other HPE NSIMC command or a script is implemented.

When the data elements are stored in the nonvolatile memory by running the “SAVE” command, any Lua scripts that have been part of the HPE NSIMC installation are also stored as part of the RDB method implementation. The stored Lua scripts are loaded back into the memory when the HPE NSIMC starts up. Thus, the Lua scripts are available across process restarts. This functionality is specific to HPE NSIMC and is not provided by the open-source Redis software.

The Lua scripts are also propagated to the backup server in the HPE NSPP mode of operation.

Master-replica configuration

HPE NSIMC supports master-replica configuration. Figure 3 shows this configuration. The whole data set managed by the master is replicated in all the configured slave nodes. You can configure **n** masters with **m** replicas. The data is duplicated from a master to a replica when the client commands a change to the data set managed by that master. However, the duplication takes place asynchronously. The client that initiated the change in data receives its response from the master before the duplication to the replica is completed. Hence, there is a finite probability that the data is not in sync between the master and the replica. This probability depends on the frequency of replication and is a compromise between performance and consistency. Clients can also directly communicate with replicas. They can run all commands on the master but can run only **read** commands on the replica.

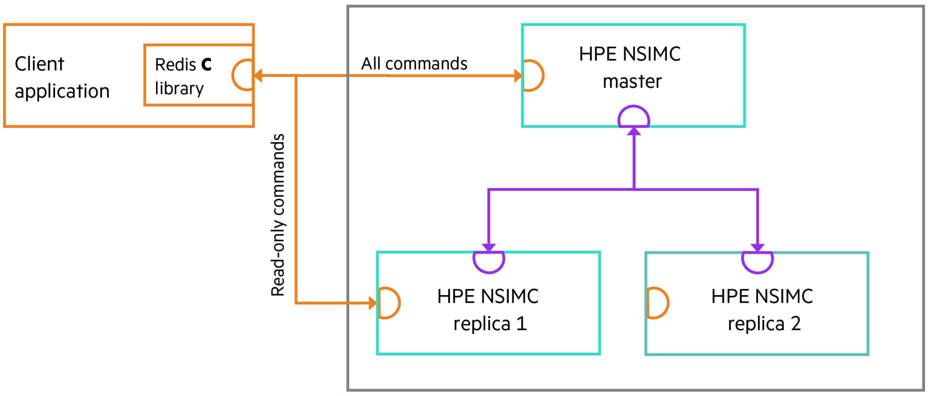

Figure 3. HPE NSIMC in Redis master-replica mode of operation

The master-replica configuration can be used in conjunction with the HPE NSPP architecture. The master and replica each have its primary and backup processes. Figure 4 shows this arrangement. However, this arrangement does not add much value in the HPE NonStop environment. HPE NSPP architecture by itself provides enough availability and data consistency for mission-critical applications against single points of failure.

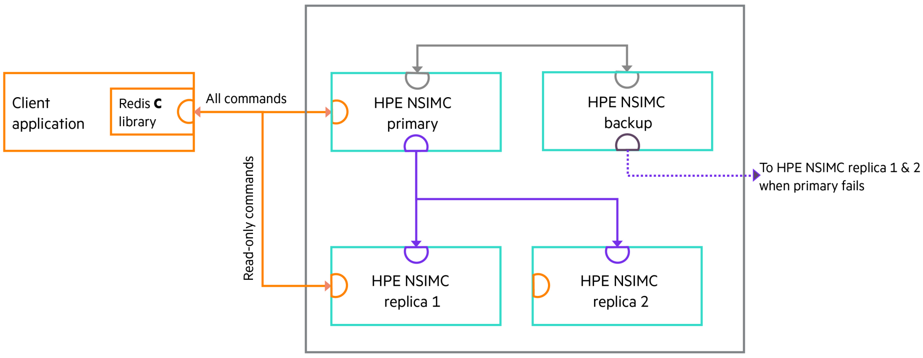

Figure 4. HPE NSIMC simultaneously working in HPE NSPP and Redis master-replica modes

Security in HPE NSIMC—SSL protection

HPE NSIMC 4.0 provided only a rudimentary security. It was strongly advised to deploy the application clients in the same HPE NonStop server as the HPE NSIMC was on. HPE NSIMC server is not meant to be exposed to networks outside of its host machine. If the client had to be placed on another machine, then it is advised to install an SSL proxy in front of the HPE NSIMC Server to protect the connection. The need for a proxy has been obviated now. From version 7.0 onward HPE NSIMC supports SSL connection between clients and the HPE NSIMC server. Thus, no additional component is required to protect the data exchanged by clients with the HPE NSIMC Server.

Security in HPE NSIMC—ACLv2[5]

In HPE NSIMC 4.0 you could enable client authentication by suitably configuring a password in the HPE NSIMC configuration file. Note that the password in the configuration file is stored as clear text. From HPE NSIMC 7.0, the ACLv2 lets you define a set of capabilities for each of the set of users/clients. A user/client is only entitled to use the set of commands, keys including patterns of keys that are configured for that particular user/client. ACLv2 along with the SSL protection of data in transit significantly enhance the security of the HPE NSIMC 7.0 server and the data it manages.

Modules

You can write your own functions as run-time libraries to the HPE NSIMC Server. These are called modules. You can define RESP commands for invoking those functions. When the HPE NSIMC Server receives those commands from a client via the RESP protocol, it runs the corresponding module, fetches the result, and returns it to the client. Modules provide the flexibility to the business application to implement methods that are not available in the HPE NSIMC core itself. It is also possible to create custom data types that are not provided by HPE NSIMC.

Modules are usually written in C language, but it is possible to write them in other languages such as C++, which have C binding functionalities. Modules are designed to work with different versions of the open source Redis software. So once written, modules need not be modified or recompiled to work on different versions of HPE NSIMC.

When deployed in HPE NSPP mode, the modules are also loaded on to the backup HPE NSIMC server. Hence, there is no break in the functionalities provided by the modules and the changes to module specific data elements are not lost during process failures.

[5] ACL was first released in Redis 6.0. It was enhanced to ACLv2 in Redis 7.0.

Cluster formation by client-side sharding

HPE NSIMC supports scalability of the cache server, which is the amount of data that can be stored, by a mechanism called client sharding. You may start multiple instances of the HPE NSIMC Server and provide the client applications with their connectivity details. The clients have to be capable of **client-side sharding**. This functionality is available in open-source libraries, for example, Jedis (for Java clients) and Redis shard (for Python-based clients). The clients then balance the data elements stored across the multiple instances using their sharding capable libraries.

Client-side caching

To reduce the number of times a client has to fetch data from the HPE NSIMC Server and thus improve the performance of both the client and the server, HPE NSIMC 7.0 supports client-side caching feature. A client can inform the HPE NSIMC server to send an "invalidation" message whenever the value pointed to by a key is changed or the key itself is deleted. The client then may request for the new value of the cached data. In the meantime, the client could store the data value in its local process memory. This means the client does not have to fetch the data every time the application needs it, thus saving fetching time, network traffic, and also the processing cycles of the HPE NSIMC server. However, there are several aspects associated with this functionality that need careful consideration and design. See the "Client-side caching in Redis" section in Redis documentation for the details.

Product acquiring information

HPE NSIMC is available on the HPE NonStop Site Update Tape (SUT) starting from HPE NonStop Release Version Update (RVU) L23.08. It is delivered as T1300 in the HPE NonStop Operating System SUT. The HPE NSIMC User Guide is available in HPE Support Center. See also Redis Documentation for more information on the functionalities described in this data sheet.

Related HPE offerings

Customer technical training

Gain the skills you need with training from Hewlett Packard Enterprise. Accelerate your technology transition, improve operational performance, and get excellent returns on your HPE investment. Training is available when and where you need it, through flexible delivery options and a global training capability. hpe.com/ww/learnnonstop

HPE Services

HPE Services leverages our strength in infrastructure, partner ecosystems, and the end-to-end lifecycle experience, to accelerate powerful, scalable IT solutions to provide you the assistance for faster time to value. HPE Services provides a comprehensive portfolio including advisory, transformational, professional, and operational services to help accelerate your digital transformation.

Operational services

- **HPE Complete Care Service:** Our most comprehensive support solution tailored to meet your specific data center support requirements. It offers a wide choice of proactive and reactive service levels to cover requirements ranging from the most basic to the most business-critical environments. **HPE Complete Care** Service is designed to scale to any size and type of data center environment while providing a single point of contact for your support needs for HPE as well as selected multivendor products.
- **Critical Service:** High-performance reactive and proactive support designed to help minimize downtime. It offers an assigned support team, which includes an account support manager (ASM). This service offers access to HPE Global NonStop Solution Center, 24x7 hardware and software support, six-hour call-to-repair commitment, enhanced parts inventory, and accelerated escalation management.
- **Proactive 24:** Provides proactive and reactive support delivered under the direction of an ASM. It offers 24x7 hardware support with four-hour on-site response, 24x7 software support with two-hour response, and flexible call submittal.
- **HPE Foundation Care:** Support for HPE servers, storage, networking hardware, and software to meet your availability requirements with a variety of coverage levels and response times.

Advisory and transformation services—HPE Services designs the transformation and builds a road map tuned to your unique challenges including hybrid cloud, workload and application migration, Big Data, and the edge. HPE leverages proven architectures and blueprints, integrates HPE and partner products and solutions, and engages professional and operational services teams of HPE Services as needed.

Professional services—HPE Services creates and integrates configurations that get the most out of software and hardware and works with your preferred technologies to deliver the optimal solution. Services provided by the HPE Services team, certified channel partners, or specialist delivery partners include installation and deployment services, mission-critical and technical services, and education services.

Learn more at

HPE.com/info/NonStop

Visit **HPE GreenLake**

Chat now (sales)

Hewlett Packard Enterprise

a00007548ENW, Rev. 2