

底层parker group为pika 360



Zookeeper：为分布式应用提供一致性服务，封装复杂的服务，将简单易用的接口提供给客户

AD：

1. 海量存、吞吐大、可扩
2. 高可用，一个节点down不影响，RTO毫秒
3. 离线批量导入（历史数据）
4. 跨机房数据同步（？通道）

SLB: server load balancer 根据转发策略将流量分发到多台云服务器。提供proxy的负载均衡，对外暴露一个VIP

Proxy：计算层，数据路由、数据聚合、协议转换。与每个节点维护一个连接池，proxy leader（zookeeper）操作集群、更新路由信息、检底层group存活。转请求到ds

Group数据分片，负责若干slot数据。M+多s

路由存在zookeeper里，router中有group信息，master和slave的ip和port。Slots到group的映射。正在迁移的slots。

根据权重分摊请求到group内的各个节点。

proxy与每个后端ds都存在连接池，ds对读写做流控

Ds存数据，采用rocksdb（持久性kv）存。

Ms通过binlog复制同步数据

异步复制，只保证最终一致性。

Read after write

备份单个ds数据存在S3系统中，从系统拉出，恢复到一致性。

扩容：迁移slot到新的group，达到负载均衡。pl后台线程并发迁移。

一机房写，多机房读。存节点会cache一份路由表，异步写入对端集群。

Ds在group中，存在主从复制关系。

Pika 解决redis内存不足，持久化存储磁盘ssd盘。18线程

等于是55 上的 data0上两个pika data1上两个pika（开一个proxy连个

Group0，G1）

56上重复55操作（开一个proxy 连g2，g3）

两边各放400G数据，每个group 4096个slot

Curl 访问网络资源 类似爬虫

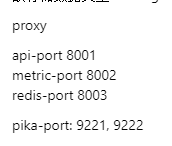
测试数据metrics

P99和P999，分位数表示



Ulimit

选项 含义-a 显示当前系统所有的limit资源信息。 -H 设置硬资源限制，一旦设置不能增加。例如：ulimit – Hs 64；限制硬资源，线程栈大小为 64K。-S 设置软资源限制，设置后可以增加，但是不能超过硬资源设置。例如：ulimit – Sn 32；限制软资源，32 个文件描述符。-c 最大的core文件的大小，以 blocks 为单位。例如：ulimit – c unlimited； 对生成的 core 文件的大小不进行限制。-f 进程可以创建文件的最大值，以blocks 为单位.例如：ulimit – f 2048；限制进程可以创建的最大文件大小为 2048 blocks。-d 进程最大的数据段的大小，以Kbytes 为单位。例如：ulimit -d unlimited；对进程的数据段大小不进行限制。-m 最大内存大小，以Kbytes为单位。例如：ulimit – m unlimited；对最大内存不进行限制。-n 可以打开的最大文件描述符的数量。例如：ulimit – n 128；限制最大可以使用 128 个文件描述符-s 线程栈大小，以Kbytes为单位。例如:ulimit – s 512；限制线程栈的大小为 512 Kbytes。-p 管道缓冲区的大小，以Kbytes 为单位。例如ulimit – p 512；限制管道缓冲区的大小为 512 Kbytes。-u 用户最大可用的进程数。例如 limit – u 65536；限制用户最多可以使用 65536个进程。-v 进程最大可用的虚拟内存，以Kbytes 为单位。ulimit – v 200000；限制最大可用的虚拟内存为 200000 Kbytes。-t 最大CPU占用时间，以秒为单位。ulimit – t unlimited；对最大的 CPU 占用时间不进行限制。-l 最大可加锁内存大小，以Kbytes 为单位。



Zookeeper地址



A memcache/redis NoSQL traffic generator and performance benchmarking tool.

Connection and General Options:

-s, --server=ADDR Server address (default: localhost)

-p, --port=PORT Server port (default: 6379)

-S, --unix-socket=SOCKET UNIX Domain socket name (default: none)

-P, --protocol=PROTOCOL Protocol to use (default: redis). Other

supported protocols are memcache\_text,

memcache\_binary.

-a, --authenticate=CREDENTIALS Authenticate using specified credentials.

A simple password is used for memcache\_text

and Redis <= 5.x. <USER>:<PASSWORD> can be

specified for memcache\_binary or Redis 6.x

or newer with ACL user support.

--tls Enable SSL/TLS transport security

--cert=FILE Use specified client certificate for TLS

--key=FILE Use specified private key for TLS

--cacert=FILE Use specified CA certs bundle for TLS

--tls-skip-verify Skip verification of server certificate

--sni=STRING Add an SNI header

-x, --run-count=NUMBER Number of full-test iterations to perform

-D, --debug Print debug output

--client-stats=FILE Produce per-client stats file

--out-file=FILE Name of output file (default: stdout)

--json-out-file=FILE Name of JSON output file, if not set, will not print to json

--show-config Print detailed configuration before running

--hide-histogram Don't print detailed latency histogram

--cluster-mode Run client in cluster mode

--help Display this help

--version Display version information

Test Options:

-n, --requests=NUMBER Number of total requests per client (default: 10000)

use 'allkeys' to run on the entire key-range

-c, --clients=NUMBER Number of clients per thread (default: 50)

-t, --threads=NUMBER Number of threads (default: 4)

--test-time=SECS Number of seconds to run the test

--ratio=RATIO Set:Get ratio (default: 1:10)

--pipeline=NUMBER Number of concurrent pipelined requests (default: 1)

--reconnect-interval=NUM Number of requests after which re-connection is performed

--multi-key-get=NUM Enable multi-key get commands, up to NUM keys (default: 0)

--select-db=DB DB number to select, when testing a redis server

--distinct-client-seed Use a different random seed for each client

--randomize random seed based on timestamp (default is constant value)

Arbitrary command:

--command=COMMAND Specify a command to send in quotes.

Each command that you specify is run with its ratio and key-pattern options.

For example: --command="set \_\_key\_\_ 5" --command-ratio=2 --command-key-pattern=G

To use a generated key or object, enter:

\_\_key\_\_: Use key generated from Key Options.

\_\_data\_\_: Use data generated from Object Options.

--command-ratio The number of times the command is sent in sequence.(default: 1)

--command-key-pattern Key pattern for the command (default: R):

G for Gaussian distribution.

R for uniform Random.

S for Sequential.

P for Parallel (Sequential were each client has a subset of the key-range).

Object Options:

-d --data-size=SIZE Object data size (default: 32)

--data-offset=OFFSET Actual size of value will be data-size + data-offset

Will use SETRANGE / GETRANGE (default: 0)

-R --random-data Indicate that data should be randomized

--data-size-range=RANGE Use random-sized items in the specified range (min-max)

--data-size-list=LIST Use sizes from weight list (size1:weight1,..sizeN:weightN)

--data-size-pattern=R|S Use together with data-size-range

when set to R, a random size from the defined data sizes will be used,

when set to S, the defined data sizes will be evenly distributed across

the key range, see --key-maximum (default R)

--expiry-range=RANGE Use random expiry values from the specified range

Imported Data Options:

--data-import=FILE Read object data from file

--data-verify Enable data verification when test is complete

--verify-only Only perform --data-verify, without any other test

--generate-keys Generate keys for imported objects

--no-expiry Ignore expiry information in imported data

Key Options:

--key-prefix=PREFIX Prefix for keys (default: "memtier-")

--key-minimum=NUMBER Key ID minimum value (default: 0)

--key-maximum=NUMBER Key ID maximum value (default: 10000000)

--key-pattern=PATTERN Set:Get pattern (default: R:R)

G for Gaussian distribution.

R for uniform Random.

S for Sequential.

P for Parallel (Sequential were each client has a subset of the key-range).

--key-stddev The standard deviation used in the Gaussian distribution

(default is key range / 6)

--key-median The median point used in the Gaussian distribution

(default is the center of the key range)

WAIT Options:

--wait-ratio=RATIO Set:Wait ratio (default is no WAIT commands - 1:0)

--num-slaves=RANGE WAIT for a random number of slaves in the specified range

--wait-timeout=RANGE WAIT for a random number of milliseconds in the specified range (normal

distribution with the center in the middle of the range)