Source Reference

Noddy Benchmark Script

This documents a noddy network latency benchmark script.

Revision	Date	Author	Note
1	2007-04-10	Kx Systems	Initial content

Contents

NoddyBenchmark File Documentation

benchmark.q File Reference

Functions

- **benchmark** ()

    The "Benchmark" code to run.

- **echo** (val)

    A function to print to a value to stdout.

- **asDestination** (host, port)

    A function to convert a hostname and port number into a destination that we can connect to using hopen.

- **asFile** (filename)

    A function to convert a filename into a file descriptor we can open using hopen.

- **haveArgument** (arg)

    A function to check whether we were passed a certain command line argument.

- **rpcOnce** (handle)

    Performs an RPC call to execute one call to the benchmark function.

- **createHandleOrDie** (server, port)

    Try to connect to the remote server on the given port.

- **executeRemoteCalls** ()

    A function to open a connection to the server and execute 'iterations' number of RPC calls to the server.

Variables

- **iterations** = 10000

    The number of iterations to execute by default.

- **port** = 5000

    The default port to use in the server process.

- **argv** = .z.x

    The command line arguments.

- **argc** = count .z.x

    The number of command line arguments.

- **argvAsDictionary** = .Q.opt .z.x

    The command line arguments as a dictionary.

- **port** = value first argvAsDictionary['port]
- **iterations** = value first argvAsDictionary['iterations]

NoddyBenchmark File Documentation

Function Documentation

asDestination (host, **port**)

A function to convert a hostname and port number into a destination that we can connect to using hopen.

Parameters:

    host The hostname of the remote machine

    port The remote port to connect to

Returns:

    A symbolic descriptor for the remote host:port

asFile (filename)

A function to convert a filename into a file descriptor we can open using hopen.

Parameters:

    filename The filename to make into a descriptor

Returns:

    The symbolic file descriptor

benchmark ()

The "Benchmark" code to run.

createHandleOrDie (server, **port**)

Try to connect to the remote server on the given port.

If we fail, then complain and exit.

Parameters:

    server The server to connect to

    port The port on which to connect

Returns:

    The open connection handle

echo (val)

A function to print to a value to stdout.

Parameters:

    val The value to print

executeRemoteCalls ()

A function to open a connection to the server and execute 'iterations' number of RPC calls to the server.
Save the timing data in the log file if requested.

haveArgument (arg)

A function to check whether we were passed a certain command line argument.

Parameters:

arg The symbolic form of a command line argument we are checking

Returns:

true if we were passed this argument, otherwise false

rpcOnce (handle)

Performs an RPC call to execute one call to the benchmark function.

Parameters:

handle an open handle to the remote server

Returns:

returns whatever the benchmark server returns

Variable Documentation

argc = count .z.x

The number of command line arguments.

argv = .z.x

The command line arguments.

argvAsDictionary = .Q.opt .z.x

The command line arguments as a dictionary.
"-flag arg" is processed so 'flag is a key and arg is the associated value.

iterations = value first argvAsDictionary['iterations]

iterations = 10000

The number of iterations to execute by default.

port = value first argvAsDictionary['port]

port = 5000

The default port to use in the server process.