SimplePerformanceTests

Table of Contents

1 What ConsumerMain/ProducerMain Does

ProducerMain injects a finite number of messages at a certain rate into a queue named "test queue". ConsumerMain retrieves messages from queue "test queue". Both programs report how fast they were able to complete their tasks.

For ConsumerMain and ProducerMain, arguments can be omitted from the right, and sensible defaults will be used. Since simple positional argument decoding is used, there's no way of providing a value for argument 4, say, while omitting any of arguments 1 through 3.

2 What MulticastMain Does

MulticastMain producer-parts send messages to the named exchange. MulticastMain consumer-parts create anonymous queues and bind them to the named exchange. This permits experimentation with the performance of direct, fanout and topic exchanges, in configurations where each consumer is to receive its own copy of the messages sent. Contrast this with ProducerMain/ConsumerMain, in which each consumer reads from a shared queue.

Programs	Deliveries per 1 publication	Copy style
MulticastMain	N	fanout
ProducerMain / ConsumerMain	1	round-robin

3 Command Line for ConsumerMain

```
bash runjava.sh com.rabbitmq.examples.ConsumerMain \
  $brokerhostname \
  $brokerportnumber \
  $writestats
```

where

- $brokerhostname: defaults to "localhost"
- $brokerportnumber: defaults to 5672
- $writestats: defaults to "true"
  - "true", to cause this instance of ConsumerMain to capture latency statistics, and write them out in a gnuplottable form, or
  - "false", to ignore latency information for this ConsumerMain instance.

4 Command Line for ProducerMain

```
bash runjava.sh com.rabbitmq.examples.ProducerMain \
  $brokerhostname \
  $brokerportnumber \
  $ratelimit \
  $messagecount \
  $sendcompletion \
  $commitevery \
  $sendlatencyinfo \
```

where

- $brokerhostname: defaults to "localhost"
- $brokerportnumber: defaults to 5672
- $ratelimit: defaults to 100000 messages-per-second (essentially "unlimited").
- $messagecount: defaults to 60000 messages sent in total.
- $sendcompletion: defaults to "false".
  - "true" for sending an extra message after the main batch to cause the consumers to exit.
  - "false" to omit the extra message.
- $commitevery: defaults to -1.
  - set to -1 to use transient-mode, without transactions.
  - set to 0 to use persistent-mode, without transactions.
  - set to >0 (e.g. 1 or 100, for instance) to use persistent-mode with fully synchronous transaction commits every N messages.
- $sendlatencyinfo: defaults to true.

5 Command Line for MulticastMain

MulticastMain differs from the other two, in both what it does and how it is invoked. All arguments are provided using command-line switches, which can be in any order. All switches are optional.

```
bash runjava.sh com.rabbitmq.examples.MulticastMain \
  -h hostName \
  -p portNumber \
  -t exchangeType \
  -e exchangeName \
  -i samplingInterval \
  -r rateLimit \
  -x producerCount \
  -y consumerCount \
  -m producerTxSize \
  -n consumerTxSize \
  -a \
  -s minMsgSize \
  -d maxRedirects \
  -z timeLimit \
  -f [mandatory|immediate|persistent]
```

The hostName and portNumber default to "localhost" and 5672.

The -a option is a simple switch: if it is present, AMQP auto-acknowledgements are used; if it is absent, explicit acknowledgement messages are sent to the broker.

The -f option can be specified multiple times. The argument to -f adds a single flag to the set of flags used for each Basic.Publish operation:

- "mandatory" is the mandatory flag on Basic.Publish
- "immediate" is the immediate flag on Basic.Publish
- "persistent" sets the DeliveryMode to 2

The exchangeType is used when declaring any exchanges used by the program, and defaults to "direct".

The exchangeName defaults to the exchangeType (ie. usually "direct", unless overridden), if an exchangeName is not supplied. The producer part of the application both declares and publishes to the named exchange. The consumer part of the application both declares and binds a queue to the named exchange.

The samplingInterval specifies how often, in seconds, statistics are to be dumped to the console and reset. The default is every 1 second.

The rateLimit is the number of messages per second that each producer part should publish, and defaults to 0, meaning unlimited (i.e. as fast as it can).

The producerCount and consumerCount control how many separate threads (and thus separate TCP socket connections to the broker) should be started as producers and as consumers, respectively. Both values default to 1.

The producerTxSize controls the number of publishes each producer sends before calling txCommit(). If the value is 0, TX-transaction mode is not used. The consumerTxSize, similarly, controls transaction batching for consumers in the same way, mutatis mutandis. Both controls default to 0, i.e. not using TX-transaction mode.

minMsgSize can be used to force the payload part of each published message to be longer than the application's minimum of 12 bytes. (The 12 bytes are used to carry a sequence number and a nanosecond tick counter, used for computing latency.) It defaults to 0, i.e. the messages sent will be as small as possible.

maxRedirects controls the number of redirects accepted during connection establishment. It defaults to 0, meaning that the broker is not permitted to redirect the client to a different address. This can be used when connecting to a broker cluster, to distribute connections between the different listening sockets in the cluster.

timeLimit controls how long this instance of MulticastMain should run for, in seconds. The default value of 0 means that it should continue running indefinitely.

(last modified Sat, 16 Feb 2008 15:23:52 GMT by tonyg@lshift.net)