



Pivotal.

Hystrix Isolation Strategies

Isolation Strategies

- Thread Pool
- Semaphore

Thread Pool

- Default
- Dedicated Thread Pool per Key (Command)
- Commands use Queues and Futures to handle command execution and timeout detection

Thread Pool - Benefits and Trade-Offs

- Benefits:
 - Provides highest level of isolation
 - Can walk away from timed out requests (hard timeouts)
- At cost of:
 - Higher number of threads in a container
 - Performance penalty
 - Thread Safety

Thread Pool - Benefits and Trade-Offs

- Use when:
 - Netflix preferred default given their use cases and demonstrated suitable overhead with thread pools
 - Hard timeouts required (where no timeout facility exists within Command method)

Semaphore

- Must be configured
- Shares Thread Pool with Container
- Command runs on originating ThreadLocal
- Logical isolation via Semaphores that track concurrent command requests

Semaphore - Benefits and Tradeoffs

- Benefits:
 - Reduced Thread Resources
 - Thread Safety
- At cost of:
 - Thread Isolation
 - Cannot walk away from timed out requests (hard timeout)

Semaphore - Use When...

- Use when:
 - Netflix recommendation for high throughput applications (Nx100 rps) or where thread creation may be expensive
 - Spring Cloud recommendation when running Spring Security, Spring Scope, or other dependencies on ThreadLocal
 - If command wraps HTTP or TCP calls handled gracefully with socket timeouts

What Happens When Threads Get Stuck?

- Thread Pool (Default)
 - Hystrix will attempt to interrupt the command thread...
 - ... But command method must be equipped to handle interrupted exception
 - If command method is not equipped to handle interrupted exception, thread pool will deplete, and shed load to fallback. Container restart required to clear condition
- Semaphore Strategy
 - Nothing. The thread will be stuck until container restart.
 - Could result in container's thread pool being depleted, and liveness issues.

Hystrix Configuration

Configuration	Description	Default
<code>hystrix.threadpool.HystrixThreadPoolKey.maximumSize</code>	The maximum thread-pool size. This is the maximum amount of concurrency that can be supported without starting to reject HystrixCommands	10
<code>execution.isolation.semaphore.maxConcurrentRequests</code>	sets the maximum number of requests allowed to a <code>HystrixCommand.run()</code> method when you are using <code>ExecutionIsolationStrategy.SEMAPHORE</code>	10
<code>hystrix.command.default.execution.isolation.thread.interruptOnTimeout</code>	indicates whether the <code>HystrixCommand.run()</code> execution should be interrupted when a timeout occurs.	true

See here for full set configuration: <https://github.com/Netflix/Hystrix/wiki/Configuration>