Slider HealthCheck

Requirement

检查一个deployed applications是否health

1) a simple URI Pattern will suffice for many components

2) an even simpler "check the port" for being open probe works for IPC

Slider servicemonitor中包含了从Hadoop 1.0 HA daemons的Service monitors，通过一个bootstrap 进程来监控service health，在独立的thread中运行。

KeyPoint

1）app\_config.json定义health的方式

2）SliderAppMaster启动HealthCheck进程

3）SliderAppMaster以REST的方式提供healthCheck的结果（可用于Slider APP State） 4）实现probeReportHandler，根据report结果进行app运行状态的更新

Design

1) definition

定义文件app\_config.json中，格式如下：

{

"schema":"http:xxx",

"metadata":{},

"global":{},

"healthchecks":{}

}

下面是Storm的例子：

"healthChecks": {

"protocol":"TCP", //监听端口

"exportGroup":"quicklinks",

"export":"nimbus.host\_port", //信息来源于PublishedExport中的quicklinks的 nimbus.host\_port

"probeInterval":3000, //监听间隔

"probeTimeout":3000, //监听超期时间

"reportInterval":3000, //向ReportHandler汇报间隔

"bootStrapTimeout":30000 // ProbeWorker的启动超期时间

}

2) SliderAppMaster中添加Health模块

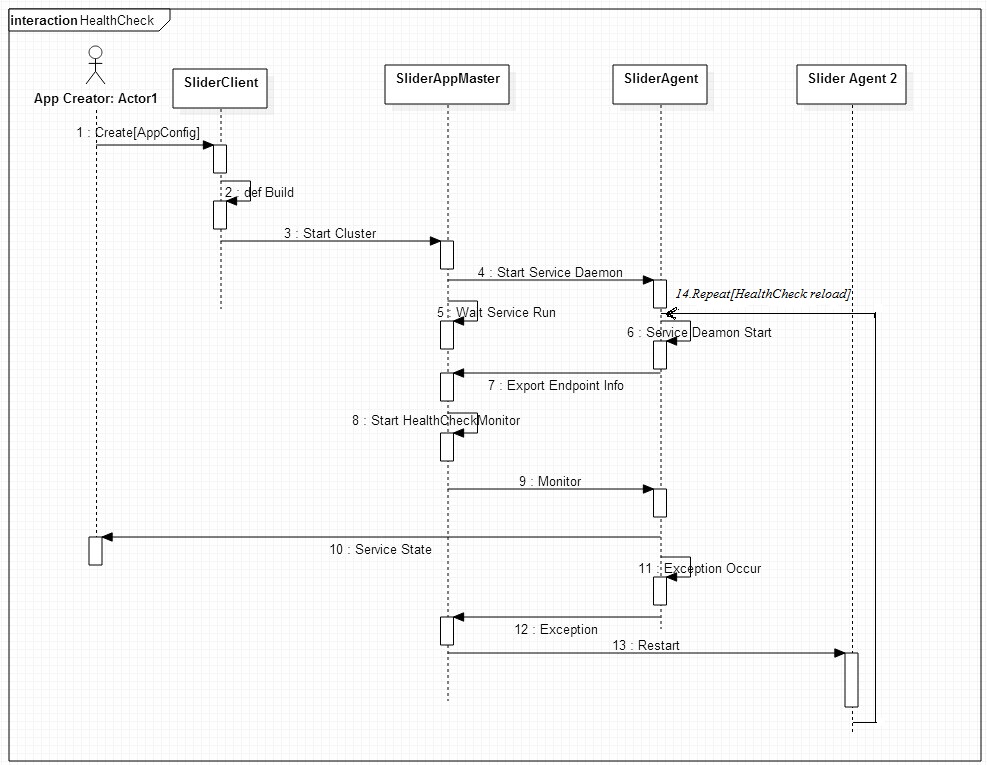
a) 增加ApplicationHealthCheck类，根据App运行状态获取监听端口或者URL信息，生成ProbeWoker，并启动，同时将结果写入MetricsAndMonitoring（显示结果）

b）需要实现ProbeReportingHandler实现类ApplicationStateReportHandler用于根据状态进行决策，目前仅接收结果，并未做进一步的处理。下一步要根据结果进行App State的更新，App的LifeCycle管理等

总体框架如下图所示：

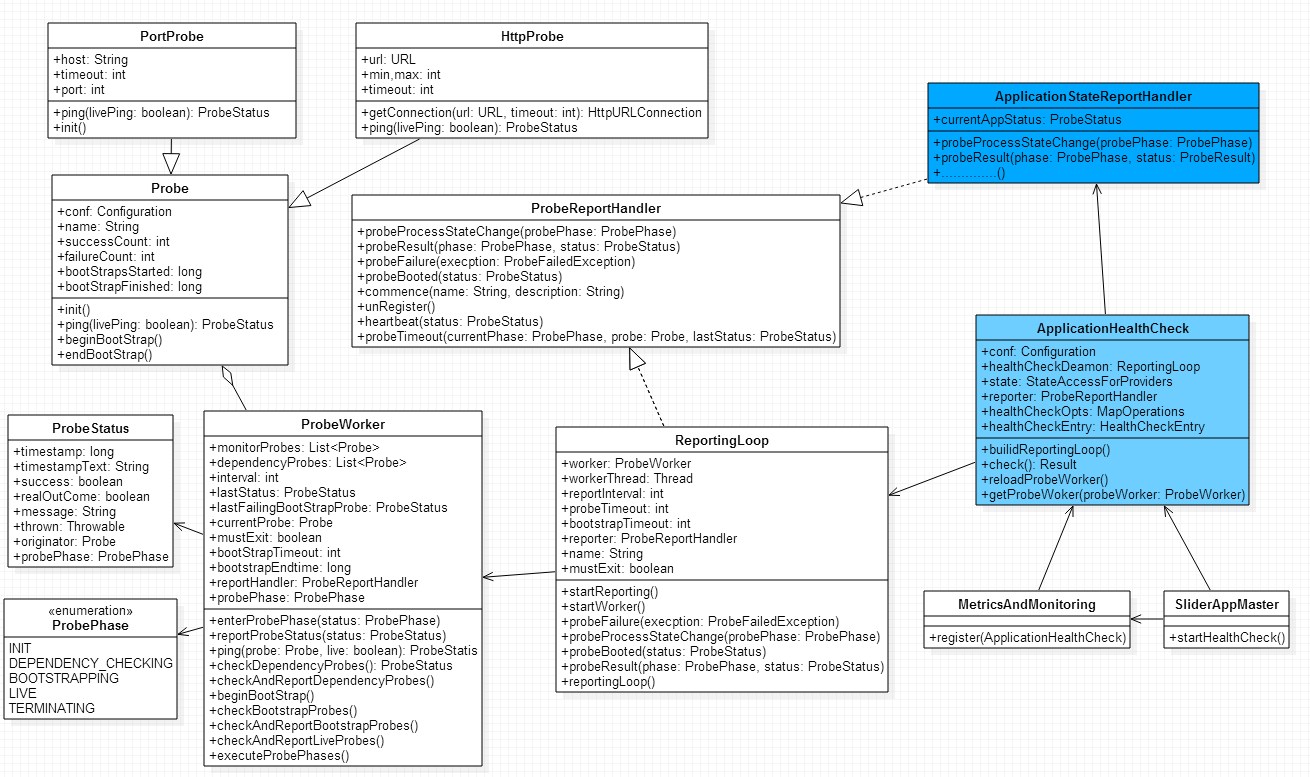


3) 执行流程



附录：

ApplicatioHealthCheck实现类图如下



附录2：MonitorKeys

|  |  |
| --- | --- |
| key | 描述 |
| service.monitor.report.classname |  |
| service.monitor.report.interval |  |
| service.monitor.probe.interval |  |
| service.monitor.probe.timeout |  |
| service.monitor.bootstrap.timeout |  |
| service.monitor.dependency.dfslive | 根据DFS being live来做monitor |

|  |  |
| --- | --- |
| service.monitor.portprobe.enabled | tpc连接 monitor |
| service.monitor.portprobe.port |  |
| service.monitor.portprobe.host |  |
| service.monitor.portprobe.connect.timeout |  |
| service.monitor.portprobe.bootstrap.timeout |  |

|  |  |
| --- | --- |
| service.monitor.lsprobe.enabled | 基于ls操作的monitor |
| service.monitor..lsprobe.path |  |
| service.monitor.lsprobe.bootstrap.timeout |  |

|  |  |
| --- | --- |
| service.monitor.webprobe.enabled | 基于url的访问来进行monitor |
| service.monitor.webprobe.url |  |
| service.monitor.webprobe.min |  |
| service.monitor.webprobe.max |  |
| service.monitor.webprobe.connect.timeout |  |
| service.monitor.webprobe.bootstrap.timeout |  |

|  |  |
| --- | --- |
| service.monitor.jtprobe.enabled |  |
| service.monitor.jtprobe.bootstrap.timeout |  |

|  |  |
| --- | --- |
| service.monitor.pidprobe.enabled |  |
| service.monitor.pidprobe.pidfile |  |