

一、 概述

`ns4_gear_watchdog` 属于轻量级应用，是 NS4 框架的守护进程，守护并管理 NS4 进程。其实现对 NS4 远程启动和停止。

`ns4_gear_watchdog` 实时监控 NS4 进程的健康状态、内存消耗、CPU 使用、内部线程；收集 NS4 实现的业务日志轨迹、业务内部实时流转的业务数据，达到实时对 NS4 进程在线上的运行状态、实现的业务以及业务数据的流转状态等方面的监控，并精准、快速、便捷地定位出异常以及异常点、CPU、线程等运行状态。提供日志轨迹功能，便捷寻找以及精准定位异常问题。

二、 设计理念

一个在线运行的应用，它的健康状态是我们关心的重点。如何能在异常情况发生前或者刚发生时及时的发现并准确定位，这是系统维护人员面对的一个很大的问题。

通常影响一个已经通过各种测试并在线上生存应用的健康因素，主要有三个因素：

1、 应用生存的环境因素：

环境因素主要包括系统层面的内存消耗、CPU 使用、负载、线程等。这些是每个线上生存应用的最基础因素，掌握了这些信息的数据，可以更清楚的了解当前进程的健康状态。

2、 实现功能的代码因素：

在系统运行过程中，代码因素是最直接的因素，代码的健康程度会更直接更猛烈的影响系统每个应用进程的健康程度，其所造成的错误或异常对系统正常运行往往是致命的，所以，我们要针对系统级别的错误要时刻重视。

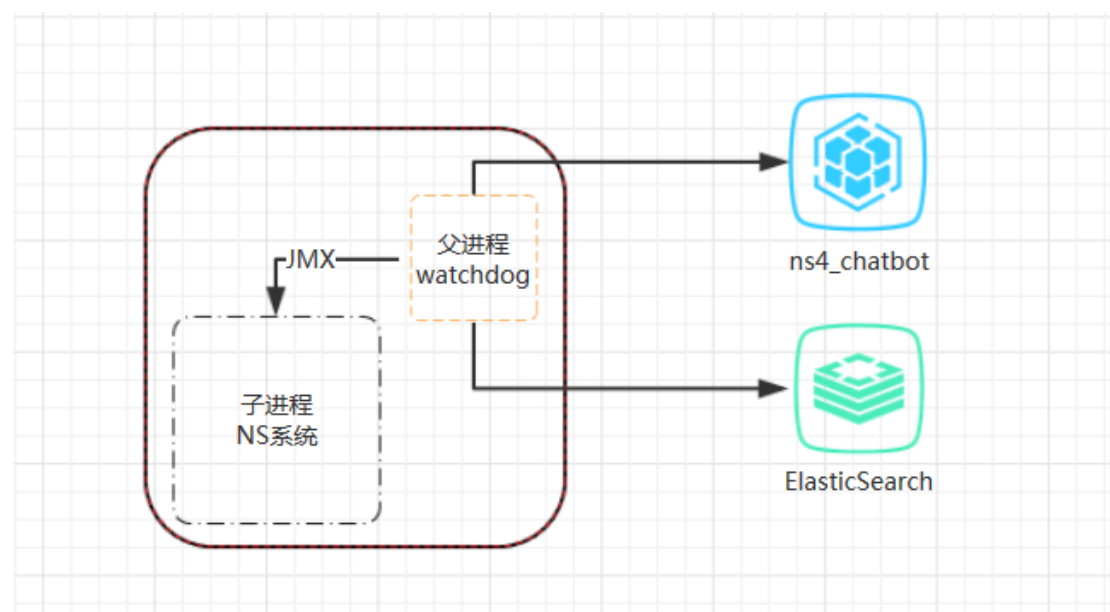
3、 业务因素：

系统正常运行是基础，而业务则是线上应用的价值所在。如何能在第一时间发现业务流程发生异常情况，并降低该由于业务问题而造成的损失，这对于系统

的服务对象来说尤为重要。在问题发生后第一时间通知相关人员及时处理该问题并解决，这样就能极大提高系统服务对象的体验。

针对以上三个主要因素，我们诞生了 watchdog。watchdog 是作为父进程存在的，通过父进程启动目标项目（子进程），并针对子进程的三大因素进行实时监控。父子进程通过 jmx 方式进行通讯，并获采集三大因素数据，将这些数据保存到 elasticsearch 中，进一步通过分析数据和通过现实运行情况总结制定出的指标相结合，将该三大因素信息通过微信机器人实时通知提醒相关负责人。

三、 基本架构图



四、 启动方法

watchdog 是一个 java 项目，watchdogServer 作为入口 `com.creditease.ns4.gear.watchdog.monitor.WatchdogServer`，在该类中启动 main 方法即可。