日志错误代码分析

1. 统计崩溃类型数量

搭建hadoop平台，步骤见附件（hadoop集群配置）。搭建成

功后，通过将LogCount.java（见附件）打包成LogCount.jar，在hadoop平台上运行，详情见图1.2,、图1.2。

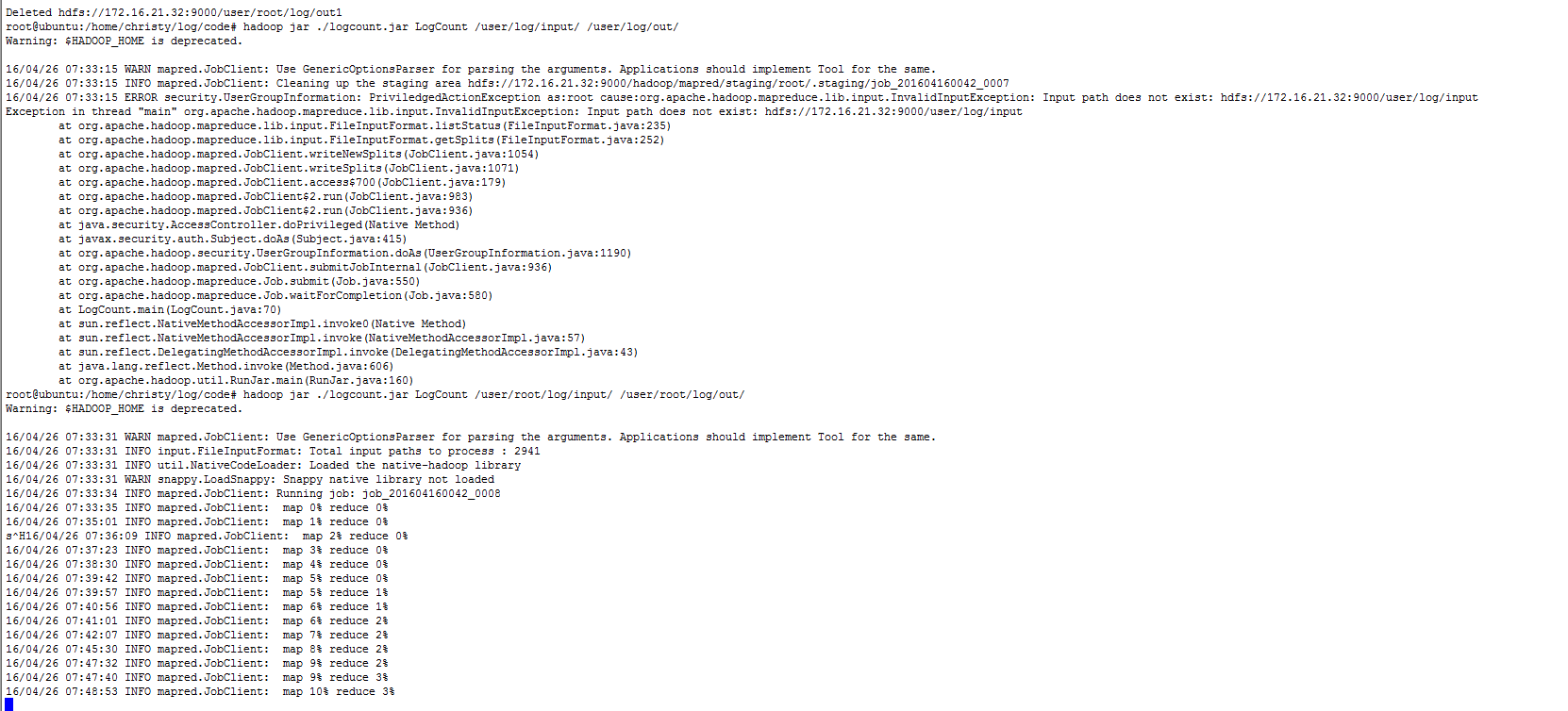


图1.1

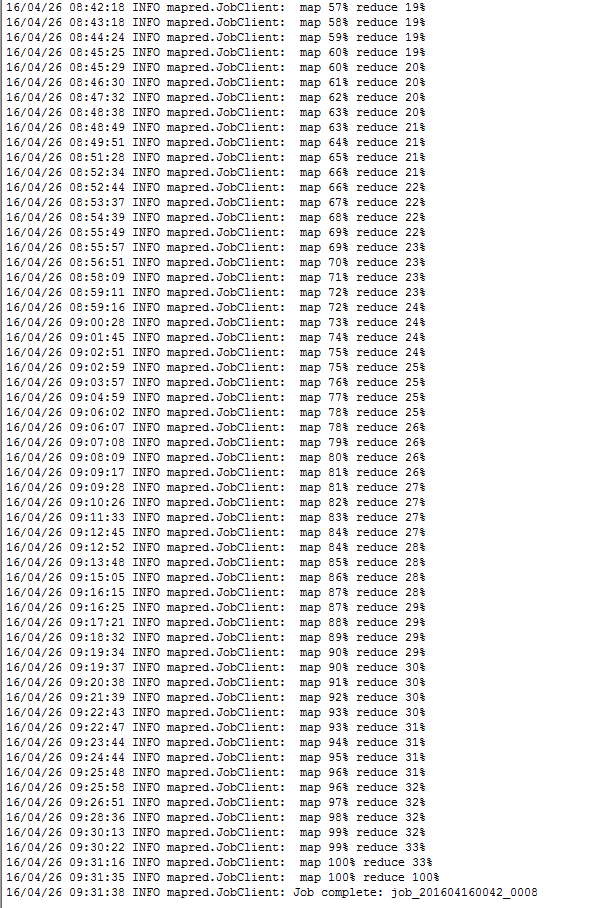


图1.2

统计结果见附件（part-r-00000）。

1. 崩溃日志产生原因

分析崩溃日志前，得先了解什么情况下会产生崩溃日志。产生崩溃日志:主要有两种情况：

1. 应用违反操作系统规则。

2.应用中有Bug。

违反iOS规则包括在启动、恢复、挂起、退出时watchdog超时、用户强制退出和低内存终止。

**Watchdog 超时机制：**

从iOS 4.x开始，退出应用时，应用不会立即终止，而是退到后台。但是，如果你的应用响应不够快，操作系统有可能会终止你的应用，并产生一个崩溃日志。

**用户强制退出**

iOS 4.x开始支持多任务。如果应用阻塞界面并停止响应， 用户可以通过在主屏幕上双击Home按钮来终止应用。此时，操作应用将生成一个崩溃日志。

**低内存终止**

在前台运行的应用拥有访问和使用内存的最高优化级。然而，这并不意味着该应用能使用设备的所有可用内存 ——每个应用只能使用一部分可用内存。

当内存使用达到一定程度时，操作系统将发出一个 UIApplicationDidReceiveMemoryWarningNotification 通知。同时,调用 didReceiveMemoryWarning 方法。

此时，为了让应用继续正常运行，操作系统开始终止在后台的其他应用以释放一些内存。所有后台应用被终止后，如果你的应用还需要更多内存，操作系统会将你的应用也终止掉，并产生一个崩溃日志。而在这种情况下被终止的后台应用，不会产生崩溃日志。

**应用中有Bug**

如你所想，大多数闪退都是由于应用中有Bug，因此大多数崩溃日志的产生都是因为应用中的Bug。Bug的种类的有很多。

**异常编码**

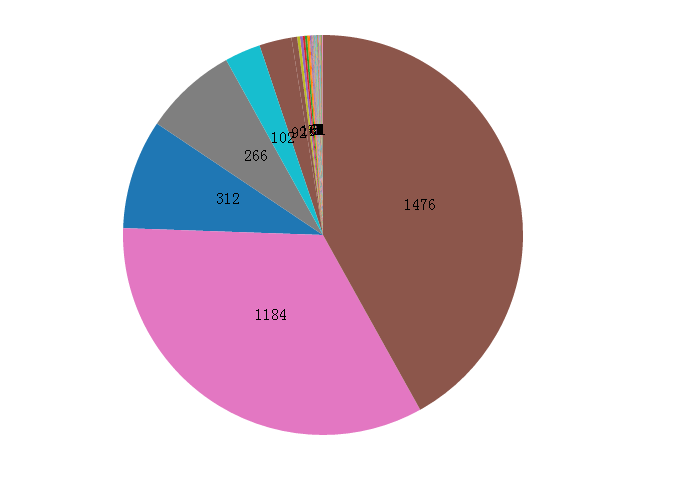
0x8badf00d: 读做 “ate bad food”!该编码表示应用是因为发生watchdog超时而被iOS终止的。  通常是应用花费太多时间而无法启动、终止或响应用系统事件。

0xbad22222: 该编码表示 VoIP 应用因为过于频繁重启而被终止。

0xdead10cc: 读做 “dead lock”!该代码表明应用因为在后台运行时占用系统资源，如通讯录数据库不释放而被终止 。

0xdeadfa11: 读做 “dead fall”! 该代码表示应用是被用户强制退出的。根据苹果文档, 强制退出发生在用户长按开关按钮直到出现 “滑动来关机”, 然后长按 Home按钮。强制退出将产生 包含0xdeadfa11 异常编码的崩溃日志, 因为大多数是强制退出是因为应用阻塞了界面。

1. 分析崩溃日志



Exception Subtype: WAKEUPS 1476

Exception Codes: 0x000000008badf00d 1184

Exception Codes: 0x0000000000000000, 0x0000000000000000 312

Exception Codes: 0x00000000bad22222 266

Exception Subtype: CPU 102

从数量统计上来看，主要的问题围绕与、**Watchdog 超时、向特定线程发送wakeup command过于频繁、以及缓存不足。**