

Linux 内核性能测试框架的实现与优化

开题报告

杨 扬

指导教师：王生原 陈渝

清华大学计算机科学与技术系

2013 年 3 月 22 日

1 问题定义及背景

2 相关工作

3 框架设计

4 目前进展

5 工作计划

1 问题定义及背景

2 相关工作

3 框架设计

4 目前进展

5 工作计划

Linux 内核每版本 patch 数量

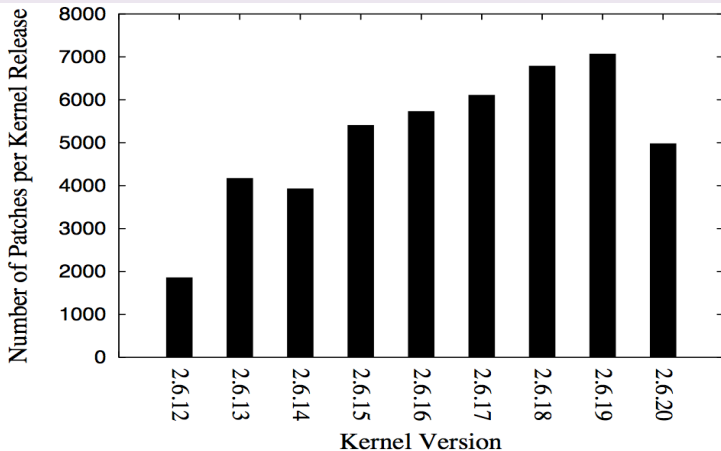


Figure : 各版本 patch 数量

Linux 开发模式

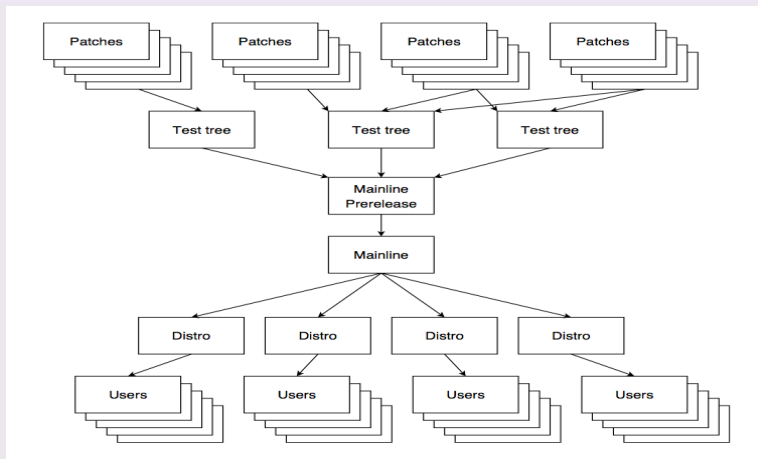


Figure : Linux 内核的开发模式

随着 Linux 内核开发的全球化, Linux 开发的步伐逐渐加快, 开发的规模也逐渐扩大, 内核开发有以下的特点

- ① 越来越多的新功能被添加到内核中来, 导致内核结构越来越复杂
- ② 内核复杂的结构使得任何一点改动对内核性能造成影响的可能性加大
- ③ Linux 内核进行官方测试的间隔一般比较大 (一般只在有新版本发布的时候才进行比较完整的测试)
- ④ 一旦有性能损失的问题, 这个问题就会随着各大 Linux 发行版扩散到广大的用户当中并被使用

性能退化 (Performance Regression) 指的是某些代码提交之后, 造成了的性能下降。

开发一套能够进行 Linux 性能测试的框架, 主要有以下特征:

- ① 较快并准确地进行 Linux 性能测试
- ② 在出现 performance regression 之后, 多次运行进行问题的确认
- ③ 确认 performance regression 之后, 能够最快地定位到出现问题的代码提交
- ④ 在定位到问题代码之后, 将相关的测试数据及出现问题的代码通知给这段代码的作者

1 问题定义及背景

2 相关工作

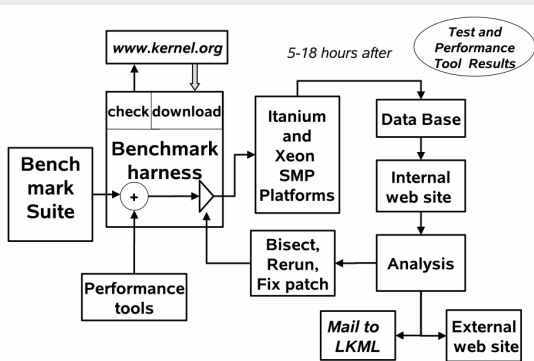
3 框架设计

4 目前进展

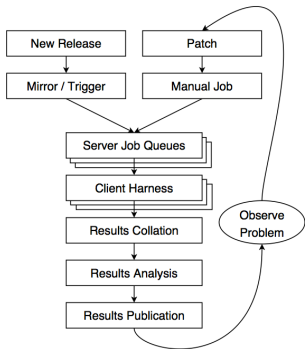
5 工作计划

Keeping Kernel Performance from Regressions

本文作者为 Intel 公司的员工，并在文中提到一种比较效率的查找内核性能下降的测试框架已经发挥了很大的作用，这个测试框架使用 Linux 的主线代码来进行测试。



Fully Automated Testing of the Linux Kernel



文中提出了对 Linux 内核进行完整的，全面的测试的测试系统所需要具备的条件：

- 代码的静态分析
- 新功能测试
- 性能测试
- 压力测试

同时，文章中还提到了如何进行有效的测试管理。

1 问题定义及背景

2 相关工作

3 框架设计

4 目前进展

5 工作计划

总体框架设计

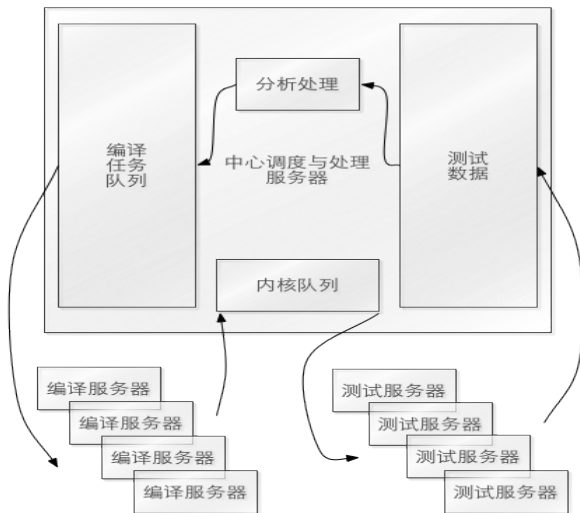


Figure : 总体框架图

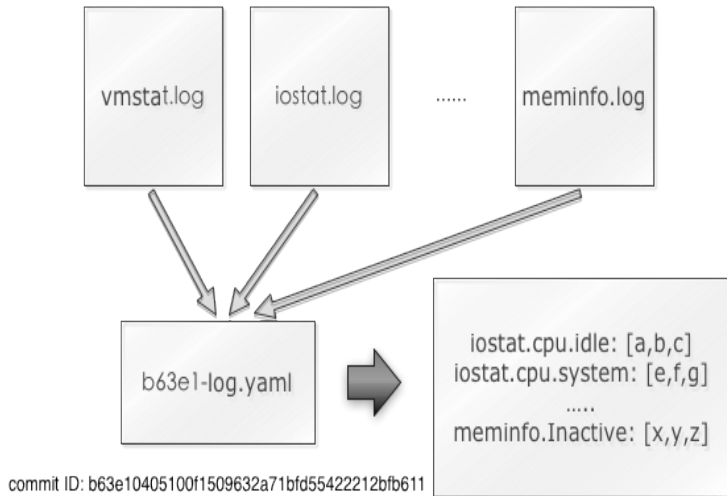


Figure : 数据流处理

有问题代码的定位 (10-Bisect)

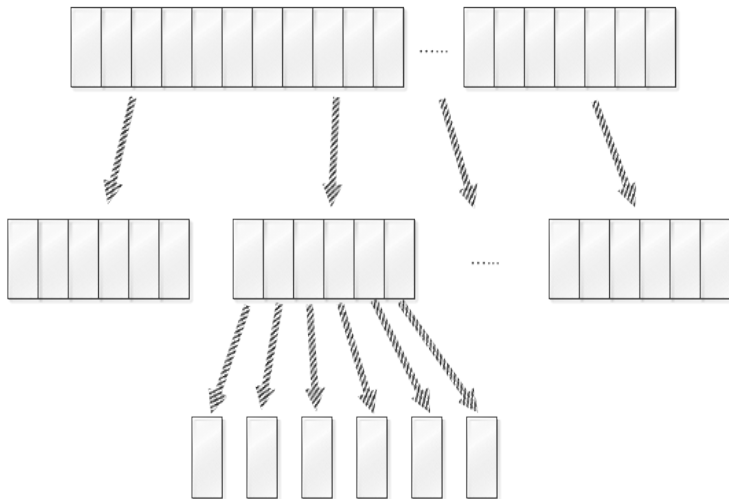


Figure : 10-bisect 算法

1 问题定义及背景

2 相关工作

3 框架设计

4 目前进展

5 工作计划

目前，我们在该项目中与 Intel 进行合作，已有的进展是：

- ① Intel 工程师吴峰光初步完成了前期框架
- ② 已完成大部分系统监视器的编写
- ③ 已完成大部分系统监视器输出的分析脚本
- ④ 初步了解了总体框架的运行机制

1 问题定义及背景

2 相关工作

3 框架设计

4 目前进展

5 工作计划

在我的毕业设计中，我的工作重点主要有以下几点：

- 理解已有的前期框架
- 完成系统监视器及相关数据提取
- 进行数据流格式化及分析
- 实现用于定位问题代码的 10-bisect 算法

在完成了上面的工作之后，就需要对整个系统进行进一步的优化，使之具有更高的性能和准确度。

工作计划—时间表

时间段	工作内容
1-4 周	开题调研及初步设计
5-6 周	完成系统监视器及相关数据提取
7-8 周	实现数据流格式化及分析
9-12 周	实现并调试 10-bisect 算法
13-15 周	完成最后的系统调试和优化
16 周及以后	撰写毕业论文

Thank you!
Q&A?