



开源社区（及代码）的 可持续性分析

周明辉

北京大学

zhmh@pku.edu.cn

<http://sei.pku.edu.cn/~zhmh>

2016-07-20

引子：软件数字考古学





软件开发活动数据

理解过去，
预测未来，
推荐实践，
支持决策。



- 获取/清洁数据
- 基于数据构造量度
- 对过去进行建模以预测未来
- 推荐最佳实践，支持开发决策



研究方法

□ 挖掘、分析和度量软件开发过程中留下的数据

- 版本控制系统 (**CVS, SVN, HG, GIT**)
- 问题追踪系统 (**JIRA, BUGZILLA**)
- 邮件列表 (**Mailman, Ezmlm**)
- **Github, Stack Overflow**

□ 使用数据挖掘/分析技术，例如

- 原始数据处理: **shell scripts, perl, python**
- 定量分析: **R语言**
- 定性分析: **interview, survey, grounded theory**



研究对象

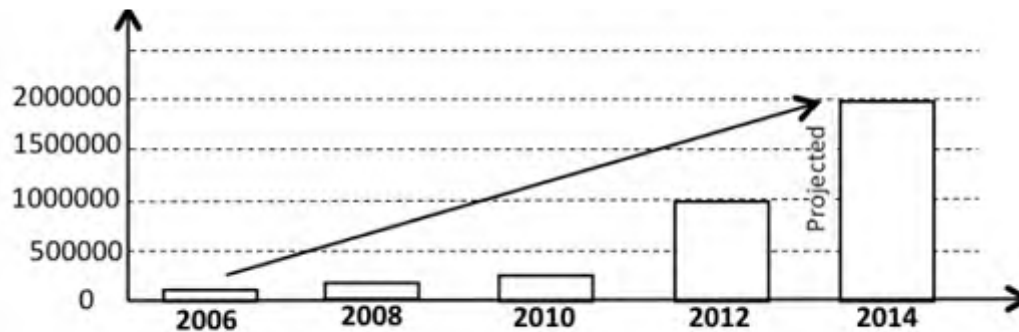
□ Anything relevant to software engineering

- 经典的复杂问题：例如，如何提高程序员效率？
- 有趣的时代问题：例如，开源生态系统如何形成？

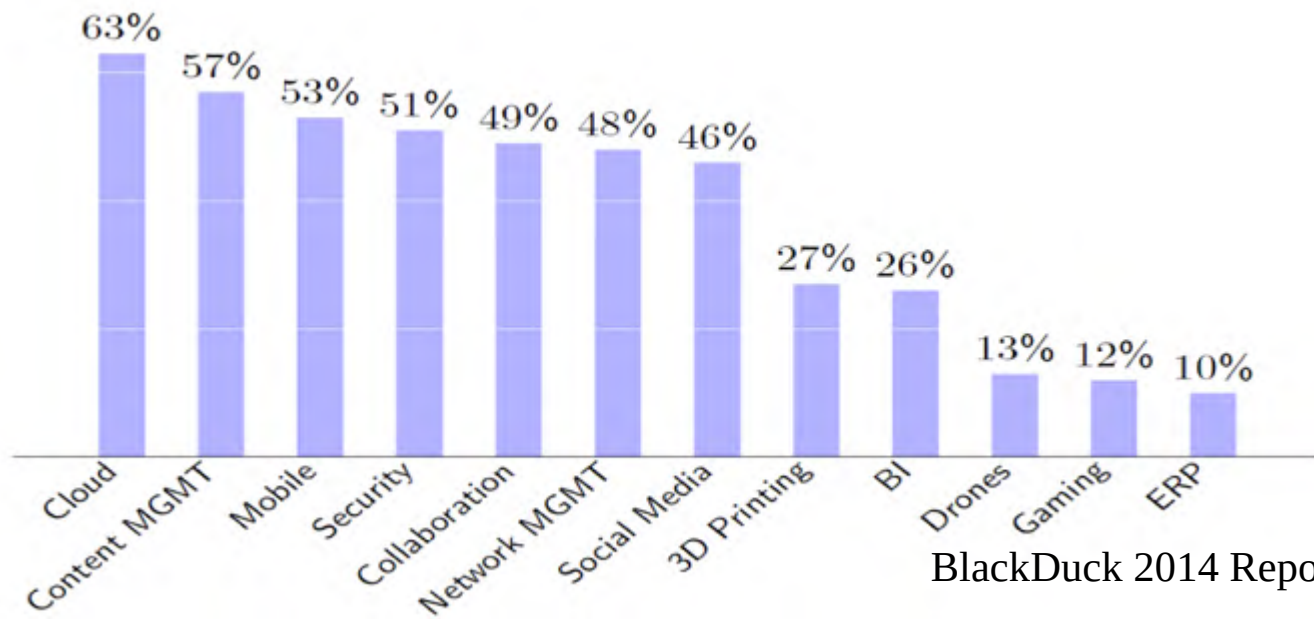
Open source is eating the world

80% of all Software Development is Open Source

-- Jim Zemlin, Linux Con 2014



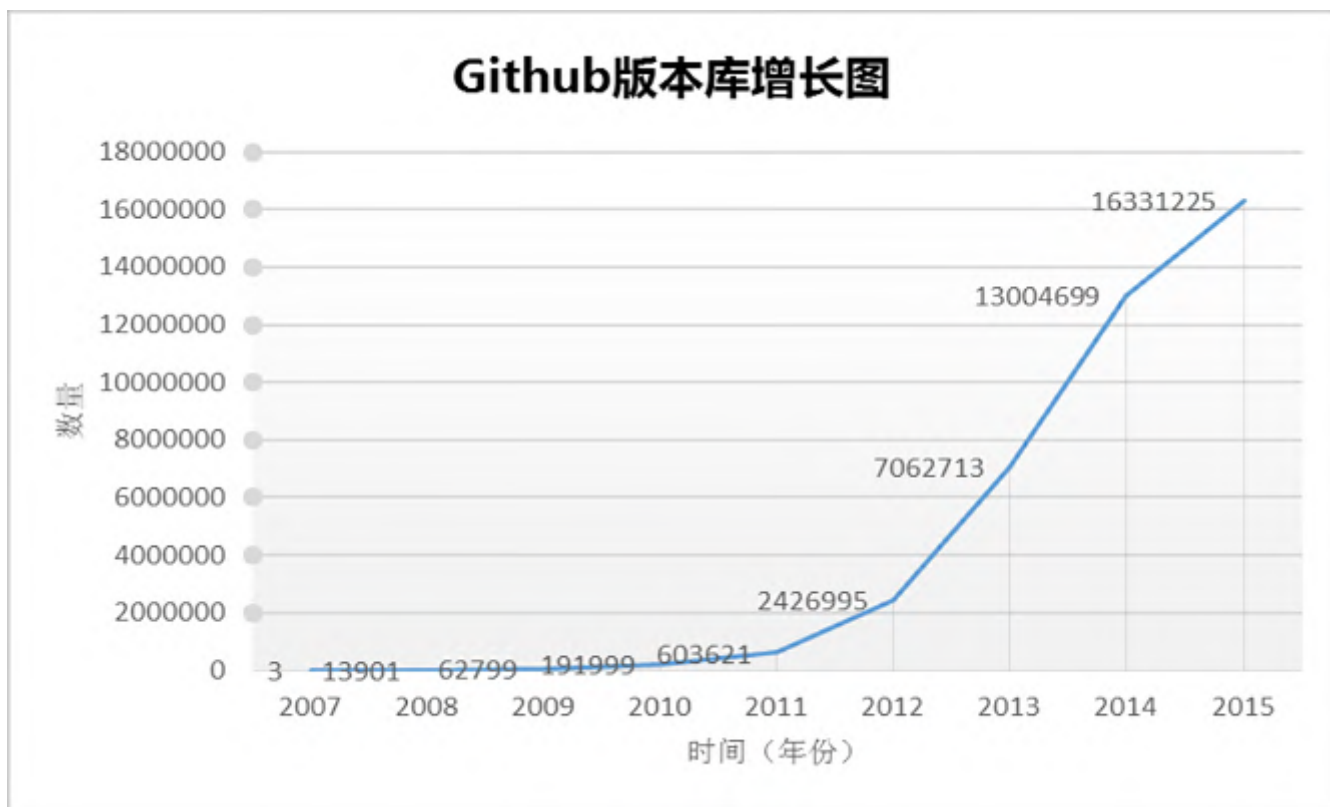
Over 1 Million Unique Open Source Projects Today



BlackDuck 2014 Report

软件及社区的持续性是关键

- 古往今来成功的项目很多
- 失败的项目更是层出不穷





开源社区/代码的生存和演化?

A, 生态社区如何持续，存活下来的社区跟消失的社区的区别是什么？

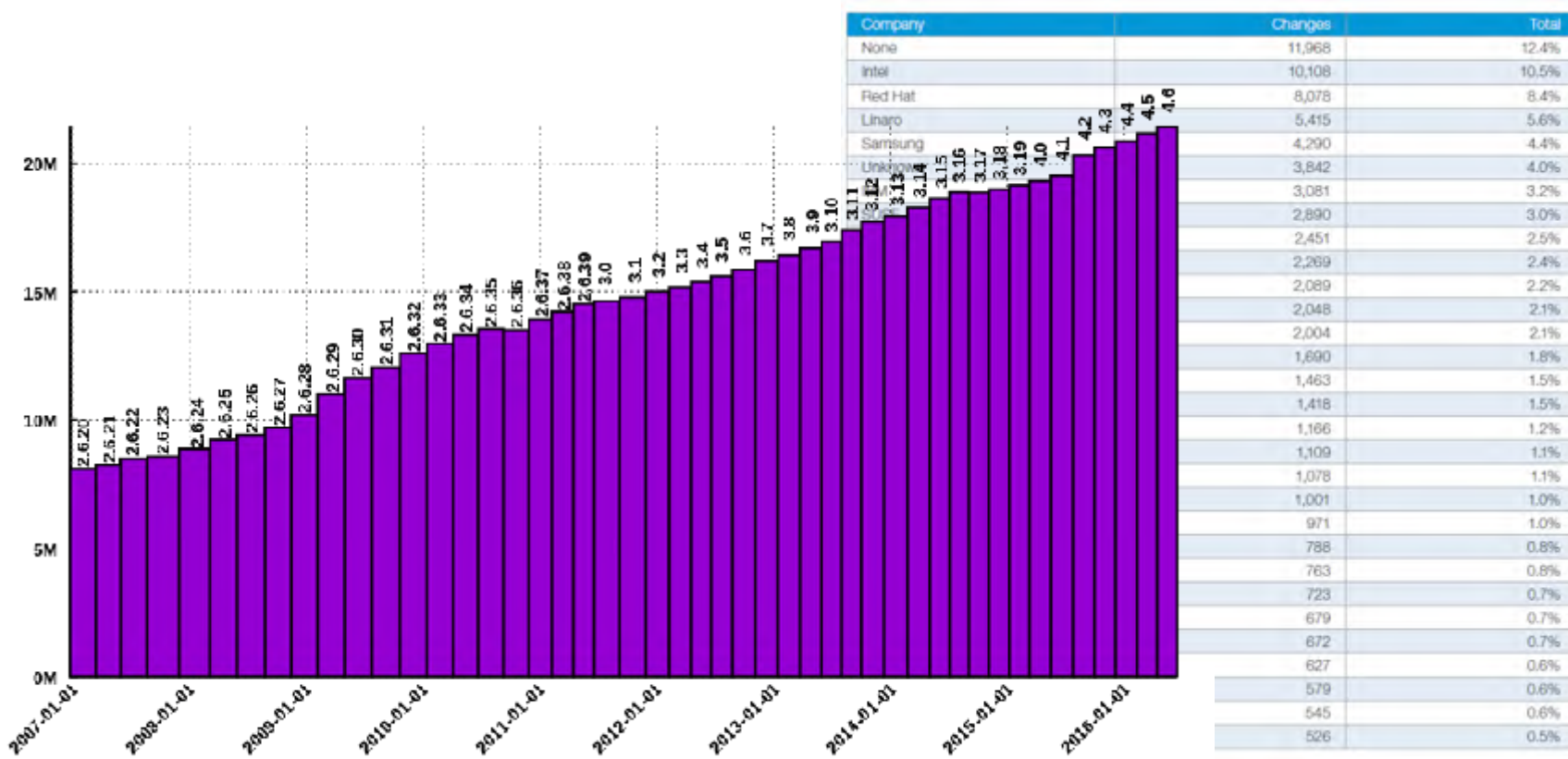
b, 复杂code是怎么组织的，怎么传播的，怎么延续并发挥影响力的？



Linux kernel



- ❑ 1991年10月发布的0.02版本有10,239 LOC， 2015年6月发布的version 4.1有 19.5 million LOC, 14000贡献者
- ❑ 企业贡献巨大：从2005年，有来自200个公司的3700位贡献者；2015年，Intel 贡献了10.5% 提交数, RedHat 8.4% (2015)
- ❑ Volunteer数量变化：2013/14.6=>2014/13.6=>2015/12.4





Linux kernel的开发是怎么组织的?

- ❑ 贡献者：数万个体，数百公司
- ❑ 功能：**driver, arch, fs, net, ...**, 因为贡献者特征对隔离性和可扩展性要求很高，形成相对独立的子系统
- ❑ 代码质量保障：**maintainer**制度

Linux is evolution, not intelligent design

— *Linus Torvalds*, 2005

有章可循吗?



Instead of a roadmap, there are technical guidelines. Instead of a central resource allocation, there are persons and companies who all have a stake in the further development of the Linux kernel, quite independently from one another: People like Linus Torvalds and I don't plan the kernel evolution. We don't sit there and think up the roadmap for the next two years, then assign resources to the various new features. That's because we don't have any resources. The resources are all owned by the various corporations who use and contribute to Linux, as well as by the various independent contributors out there. It's those people who own the resources who decide...

— Andrew Morton, 2005

方法

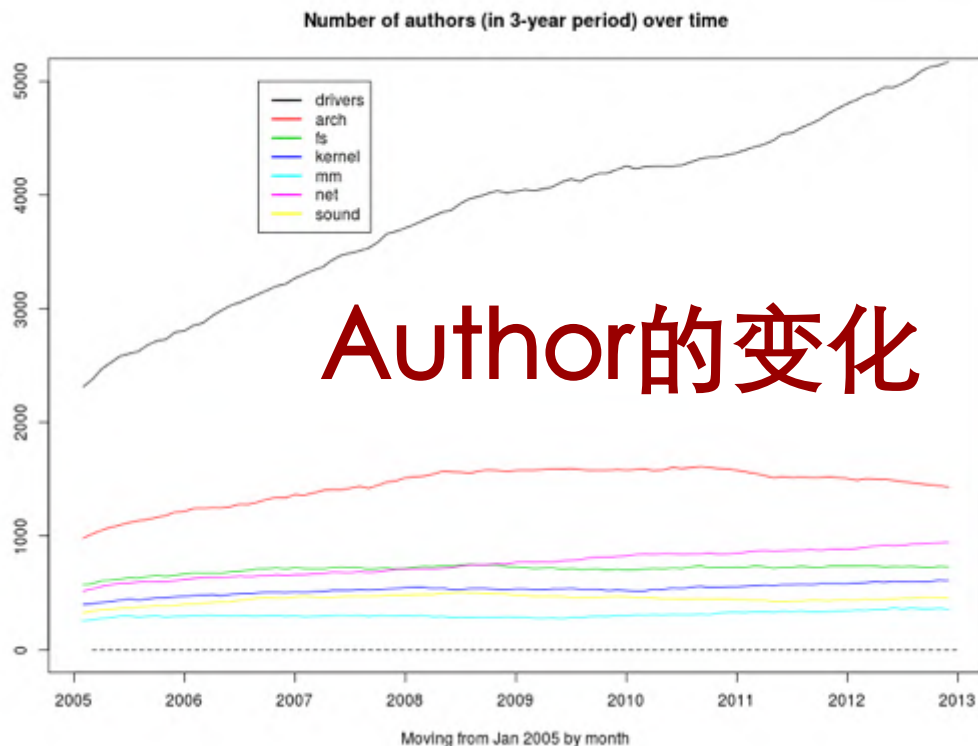
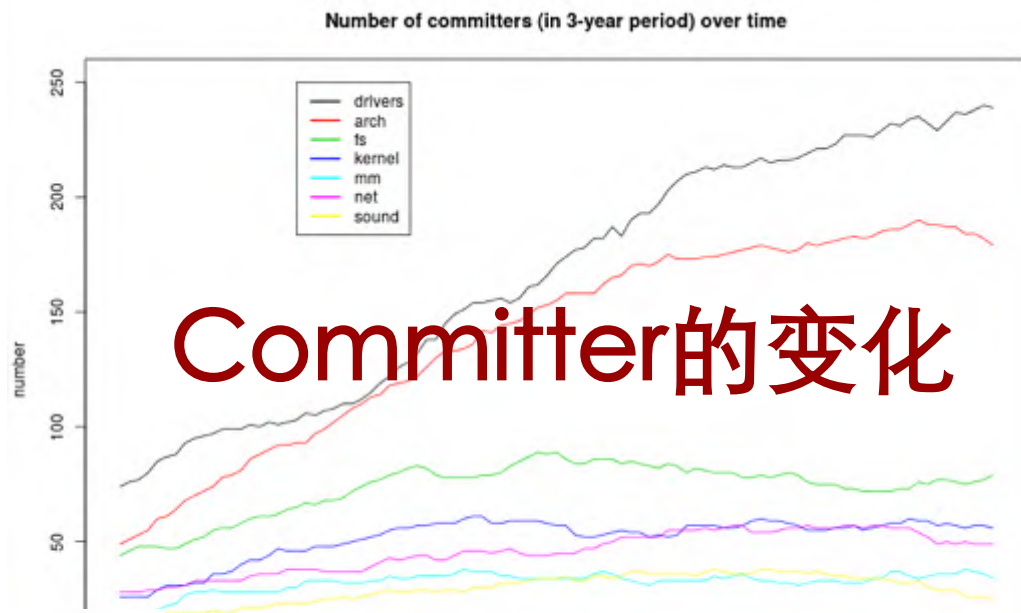
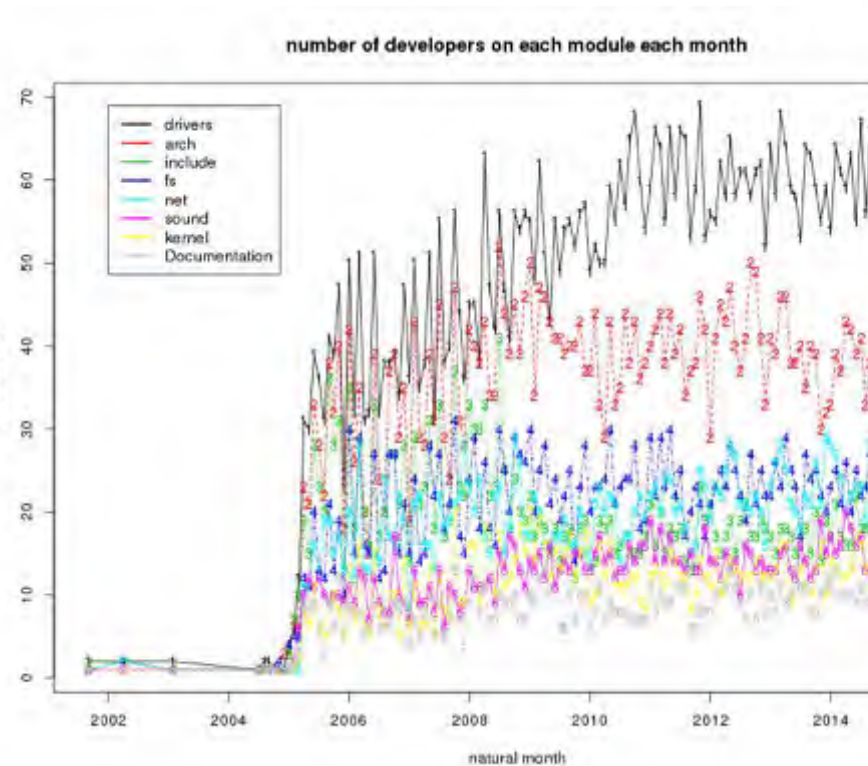
- ❑ 下载 **mainline repository** (由 **linus torvalds** 维护), 提取提交日志
- ❑ 度量和观察在不同子系统中贡献者的组织布局, 度量贡献者和代码的交互 -- 度量贡献者、任务及制品的属性及其关系
 - 合并重名: 以 **email** 或 **name** 为句柄? 姓名和名姓
 - 去除只为自己提交的 **committer**
 - 提交日志的第一层目录为子系统名称:

author	author time	committer	commit time	changed file	module
Minfei Huang	Nov 6 16:32:45 2015 -0800	Linus Torvalds	Nov 6 17:50:42 2015 -0800	kernel/kexec.c	kernel

- 与 **linux developer** 交互以验证数据、量度和结果



从贡献者的组织布局开始 度量和观察





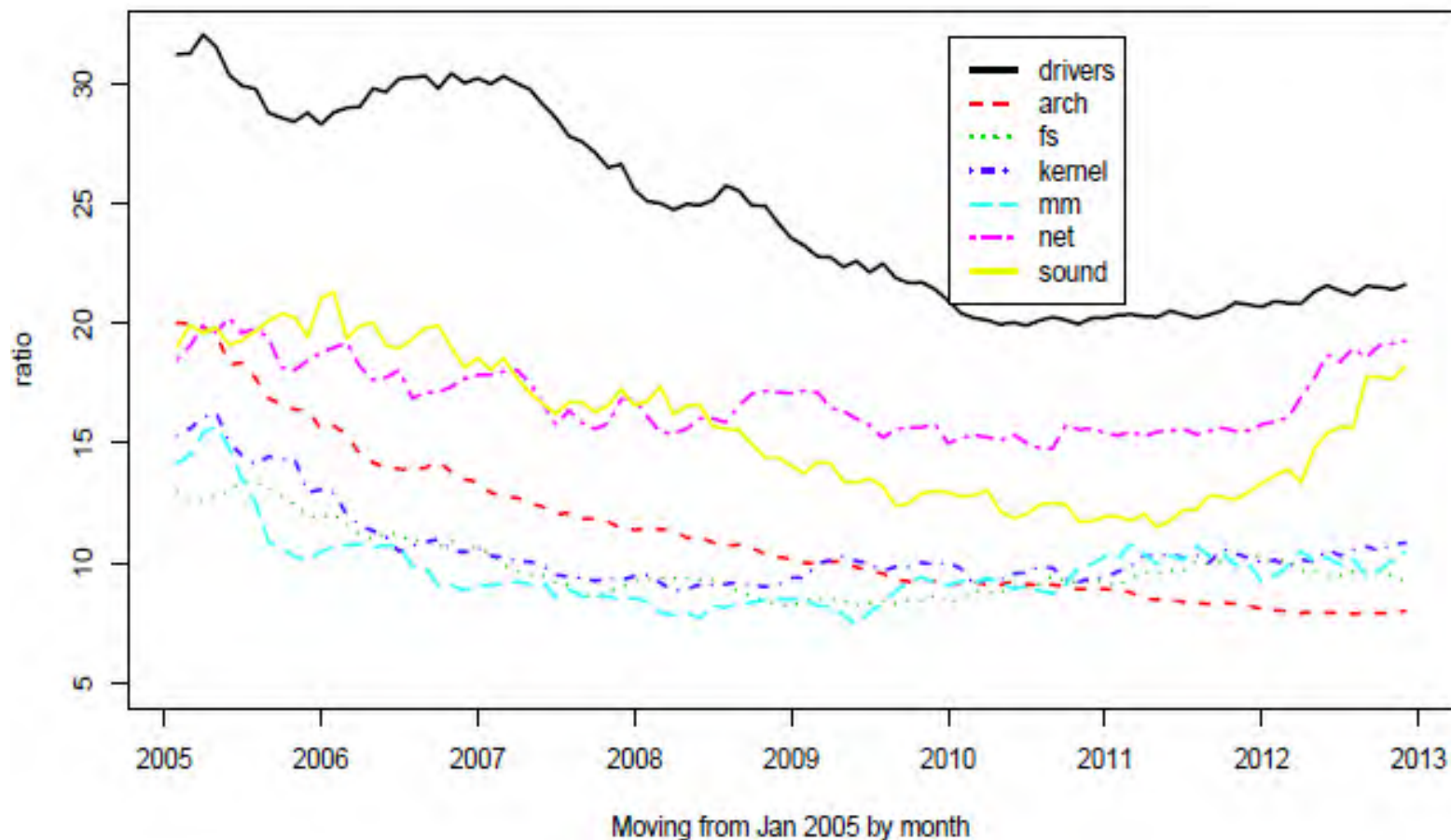
贡献者的组织布局: A2C ratio

❑ **author**对**committer**的比率: 代表很多内容.....

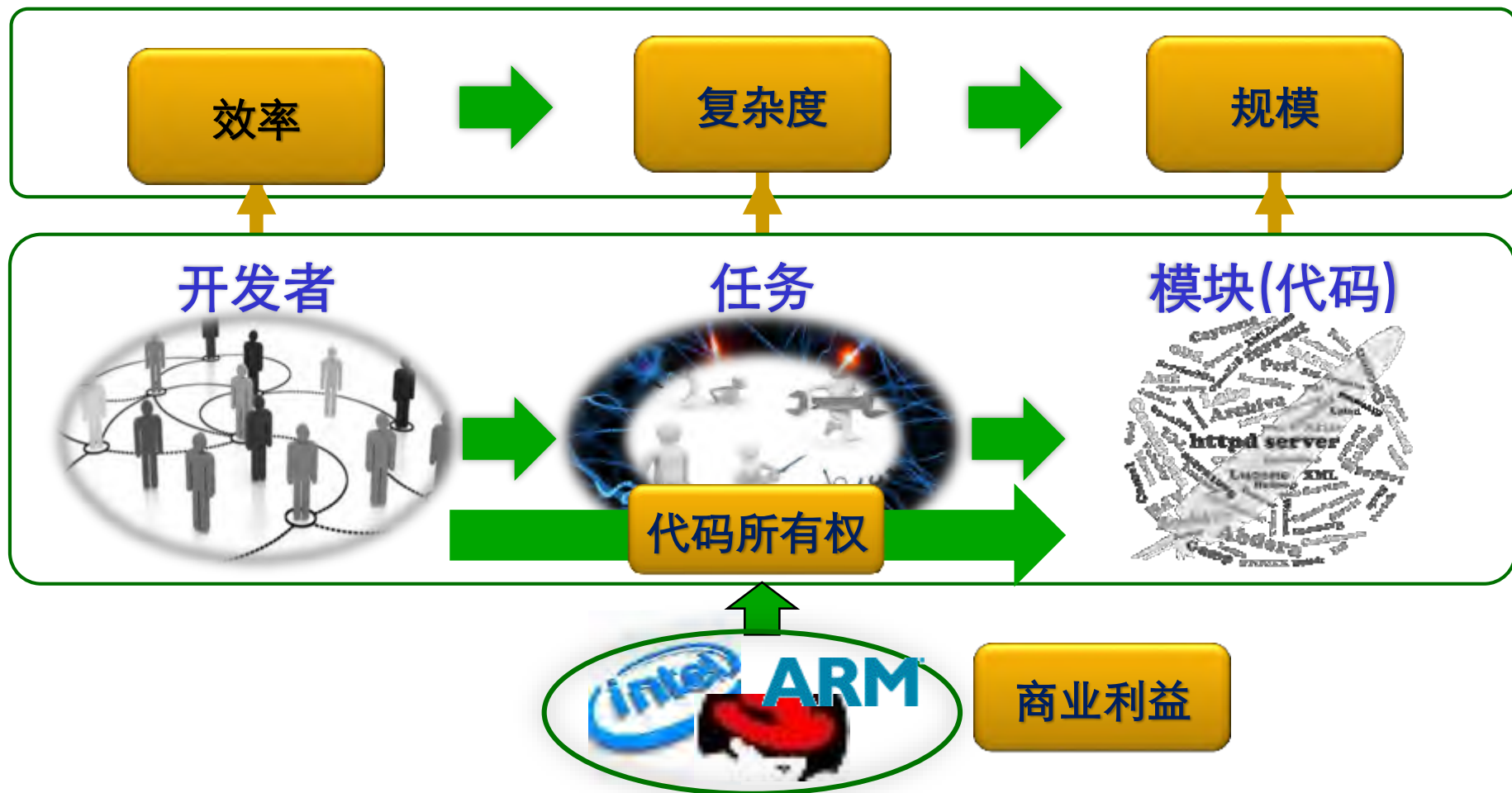
在不同子系统中, **A2C ratio**几近常量

❑ The ratio is a community phenomenon imposed by the underlying architectures.

Ratio of authors to committers (in 3-year period) over time



贡献者与制品交互形成依赖模式

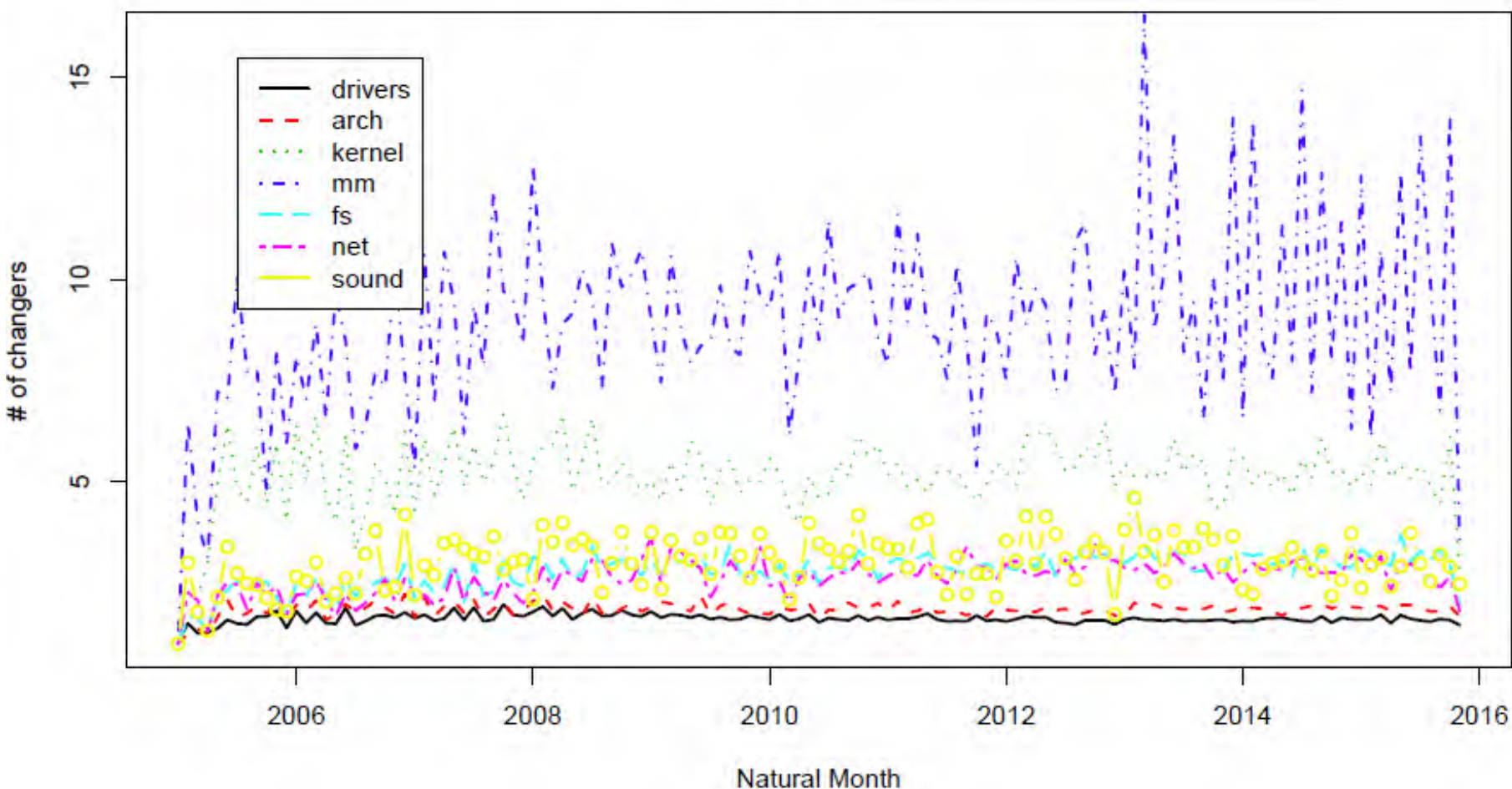




依赖模式：代码所有权

□ 不同子系统的代码所有权有清晰的层次

Average # of changers per file per month





依赖模式：商业兴趣

- 不同子系统被商业公司支持的力度有清晰的系统特征

	numMtnr	numSptr	numWord (avgWord)
drivers	366	258	637482 (33.05928)
net	14	7	23594 (14.75547)
sound	10	9	93504 (52.26607)
kernel	18	10	183874 (560.59146)
mm	0	0	1733 (16.50476)
fs	24	18	80719 (45.50113)
arch	61	40	186878 (11.47829)

子系统的分类

□ 特征变化，子系统分类保持不变

	log(nf)	log(loc)	group
drivers	9.866979	15.96378	2
net	7.377134	13.37935	1
sound	7.489412	13.43963	1
kernel	5.793014	11.98009	3
mm	4.653960	11.16445	3
fs	7.480992	13.60760	1
arch	9.697754	14.51425	2

子系统规模

	nf2	loc2	loc8chg	nf8chg	loc8an	nf8an	group
drivers	9.866979	15.96378	5.316161	2.278572	6.374748	6.430273	3
net	7.377134	13.37935	4.711986	2.246477	5.455766	5.374554	2
sound	7.489412	13.43963	4.998158	2.035159	6.031737	5.732996	2
kernel	5.793014	11.98009	3.971407	1.407670	4.541640	2.349141	1
mm	4.653960	11.16445	3.642324	1.314953	4.125898	1.945615	1
fs	7.480992	13.60760	4.795260	2.229750	5.685973	5.085726	2
arch	9.697754	14.51425	5.010073	2.444008	5.974197	7.042772	3

任务复杂度

贡献者效率

	nf2	loc2	loc8chg	nf8chg	loc8an	nf8an	avgdvpf	group
drivers	9.866979	15.96378	5.316161	2.278572	6.374748	6.430273	1.620133	2
net	7.377134	13.37935	4.711986	2.246477	5.455766	5.374554	2.631225	1
sound	7.489412	13.43963	4.998158	2.035159	6.031737	5.732996	3.023679	1
kernel	5.793014	11.98009	3.971407	1.407670	4.541640	2.349141	5.158408	3
mm	4.653960	11.16445	3.642324	1.314953	4.125898	1.945615	8.914708	3
fs	7.480992	13.60760	4.795260	2.229750	5.685973	5.085726	2.793329	1
arch	9.697754	14.51425	5.010073	2.444008	5.974197	7.042772	1.843785	2

代码所有权

商业兴趣...



Linux kernel的组织原则

□ 子系统们分为三类

- **Drivers cluster (arch)**, 子模块隔离性强, 贡献者独立开发, 任务相对简单, 商业参与多, **A2C ratio**在20以上
- **Kernel cluster (mm)**, 子模块依赖性强, 贡献者共享开发, 任务复杂度高, **A2C ratio**在10以下
- **Net cluster (sound, fs), in the middle**

□ 启示?

- 强大的人类总可以**make it work**, 但我们可以将隐式知识显示化, 推动文明进程



开源社区/代码的生存和演化?

A, 生态社区如何得以延续，存活下来的社区跟消失的社区的区别是什么？

b, 复杂code是怎么组织的，怎么传播的，怎么延续并发挥影响力的？



从最基本的问题开始

- 面向开源宇宙的版本库
 - 来自于下述项目的代码提交历史:
 - Github, googlecode, sourceforge, ...
- 从最基本的问题开始
 - What projects are most frequently reused?
 - What folder names are common?
 - How do people write code across projects?
- 应用价值
 - Suggesting the good quality code
 - Suggesting best practices for writing code

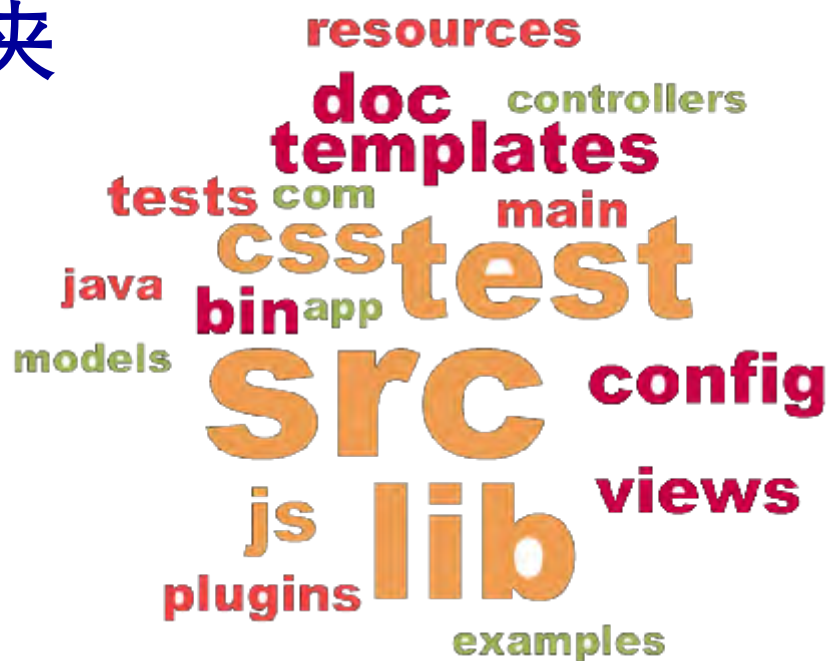


最经常复用的项目

Co-occur pair	Project	Application area
lib & mm, include & mm	linux kernel	OS
config & script, config & public	Ruby on Rails	web framework
js & langs, css & langs	TinyMCE	editor component
lib & feature	Cucumber	test framework
lib & spec	Rspec	test framework

最经常使用的文件夹

Top 20文件夹



微过程（模式）

- 跟程序语言和应用领域相关, e.g., *com* 和 *css*
- 涉及标准文件夹, e.g., *test*, *doc* 和 *examples*

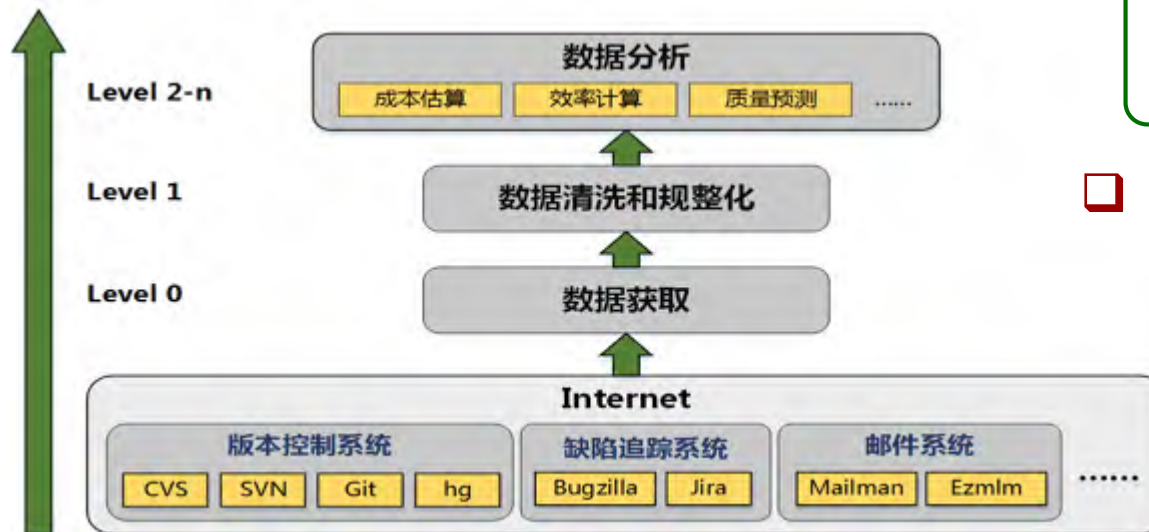


其他一些有趣的问题？

- ❑ 从mailing-list到pull-request: 代码贡献工具真的在进步吗？
- ❑ 许多工作采用很fancy的方法推荐代码reviewer，但实际项目只是简单地随机选择：简单方法够用吗？
- ❑ 技术问题vs.社会问题
- ❑ 多学科交叉应用：机器学习，高性能计算，...

软件工程大数据

- Internet上所有开源项目的版本控制系统数据总量约在70T
- Github.com有超过1千万的项目，SourceForge.net和GoogleCode分别超过30万
- Gnome每天有上百个新的缺陷提出，Mozilla每天有2万多条新的提交



大规模开源数据池

Source	Type	Files	File/Vol	Unique File/Vol	Downloads	From
Issue Data						
Gnome (517793), mozilla (620479), jboss (93149), apache (228879)	Issues	12.589K	12.589K	12.589K	1000	
Email Archives						
Gnome (1343372), jboss (1052291), apache (6323416)	Emails	1.11K	1.11K	1.11K	1000	
Version control data						
gnome, mozilla, jboss, apache, netbeans, openjdk, opensolaris, symbian, android, debian, github, githenel, gitorious, savannah, repo.or.cz	Repositories	1.11K	1.11K	1.11K	1000	

Passion-Lab
Software Engineering Institute
Peking University

Data Center

开源世界全镜像
目前 约80T 数据

硬件基础设施

- DELL R910(4U), 64GbRAM, 16-cores X7550,
- DELL MD3200, 12*2TB SAS
- DELL R710 * 4, 64GbRAM
- DELL R720, 12*3 TB SAS

<https://passion-lab.org>

谢谢

