Challenges in Mining Whole Software Universe

Katsuro Inoue
Osaka University

Analyzing Evolution of `kern_malloc`

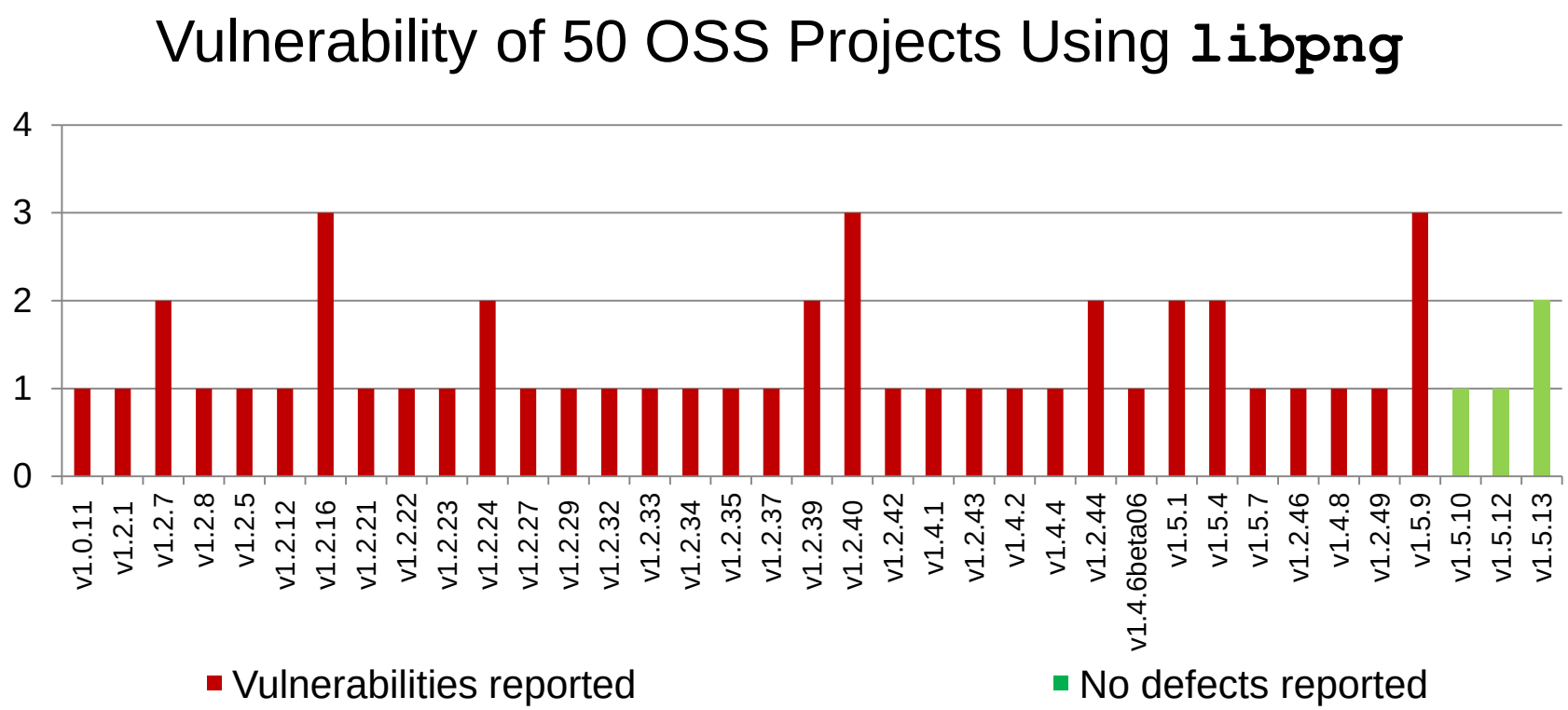
1	Lites 1.0	(G)	28-33	Kame	(G)
2	Kernel Source Archive - CMU Mach 3.0	(K)	34-36	SimOS	(K)
3	Lites 1.1.u3	(G)	27-46	Kame	(G)
4	Lites 1.1-950808	(G)	47	Netnice	(G)
5	The Rio (RAM I/O) Project	(K)	48	Kame	(G)
6	ftp in The University of Edinburgh	(G)	49-50	Psumip	(G)
7	Mip-summer98	(G)	51	Netnice	(G)
8	freeBSD/SPARC	(G)	52	Reflexprotocol	(G)
9-12	ftp in Stockholm University	(G)	53	Netnice	(G)
13	freeBSD-cam2.1.5R	(G)	54	NetBSD v1.105	(K)
14-15	SonicOSX	(K)	55	OpenBSD PV Xen	(G)
16	Labyrinth BSD(labyrinthos)	(K)	56	OpenBSD v1.73	(K)
17	Oskit	(G)	57	Pmon	(G)
18	Psumip	(G)	58-62	Proyecto A.T.L.D. GNU/hurd(extremelinux)	(K)
19	Mach	(G)	63	OpenBSD v1.74	(K)
20-22	Savannah	(G)	64	Pmon	(G)
23	Unofficial OSKit source	(K)	65	774	(G)
24-26	Unofficial OSKit source(oskit)	(K)	66	Chord-ns3	(G)
27	ftp in Stockholm University	(G)	67	Openbsd-loongson-vc	(G)

Results by G(Google Code Search) and K(Koders)

: File in New BSD License

: File in Original BSD License

Analyzing Reuse of Outdated Libraries



Result from Google Code Search and Koders

Experience and Concern

Mining source code repositories, e.g.,
SourceForge, Github, Open Hub, Google Code, Marven, ...
(BlackDuck)

- Outcomes heavily depend on repository contents
- Aren't we mining a small world?
- There may be many other source code contents in the universe


sourceforge


fedora
codeBeamer JavaForge
Gna!
GITGO
CloudForge by CollabNet
FusionForge
Apache maven
GitHub
berliOS
Entdecke Open Source
GitLab.com
OH
our project.org
SEUL.org
sourceforge
CodePlex
Google Developers
ALioth.debian.org
OW2
assembla
Bitbucket
freepository.
GITORIOUS
Tigris.org
Open Source Software Engineering Tools
launchpad

Whole Software Universe U

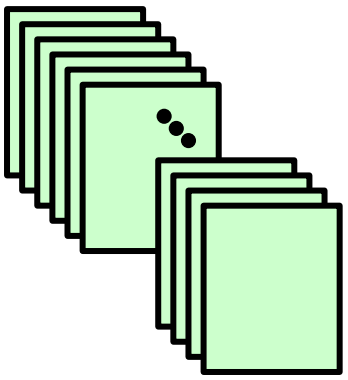
- Whole Software Universe

$$U \equiv Collection\ of\ All\ Software\ Developed\ by\ Human\ in\ the\ Past$$

  - Open source software
  - Personally-developed software
  - Proprietary software

  ... any others

- P : Set of all meaningful software (a countable infinite set)
- $U \subseteq P$



Questions for U

A) How do we get U?

B) What do we mine from U?

C) How do we mine U?

D) Why do we mine U?

A) How Do We Get U?

- No one knows actual U
- So we would collect many repositories, and construct a subset $U' \subseteq U$
- U' should be as large as possible, of course
- U' should reflect characteristics of U
- Challenges
  - Collecting and unifying different repositories into U'
    - Duplication, coherence, ...
  - Performance and capacity for U'
  - Updating and maintaining U'

B) What Do We Mine from U?

Examples

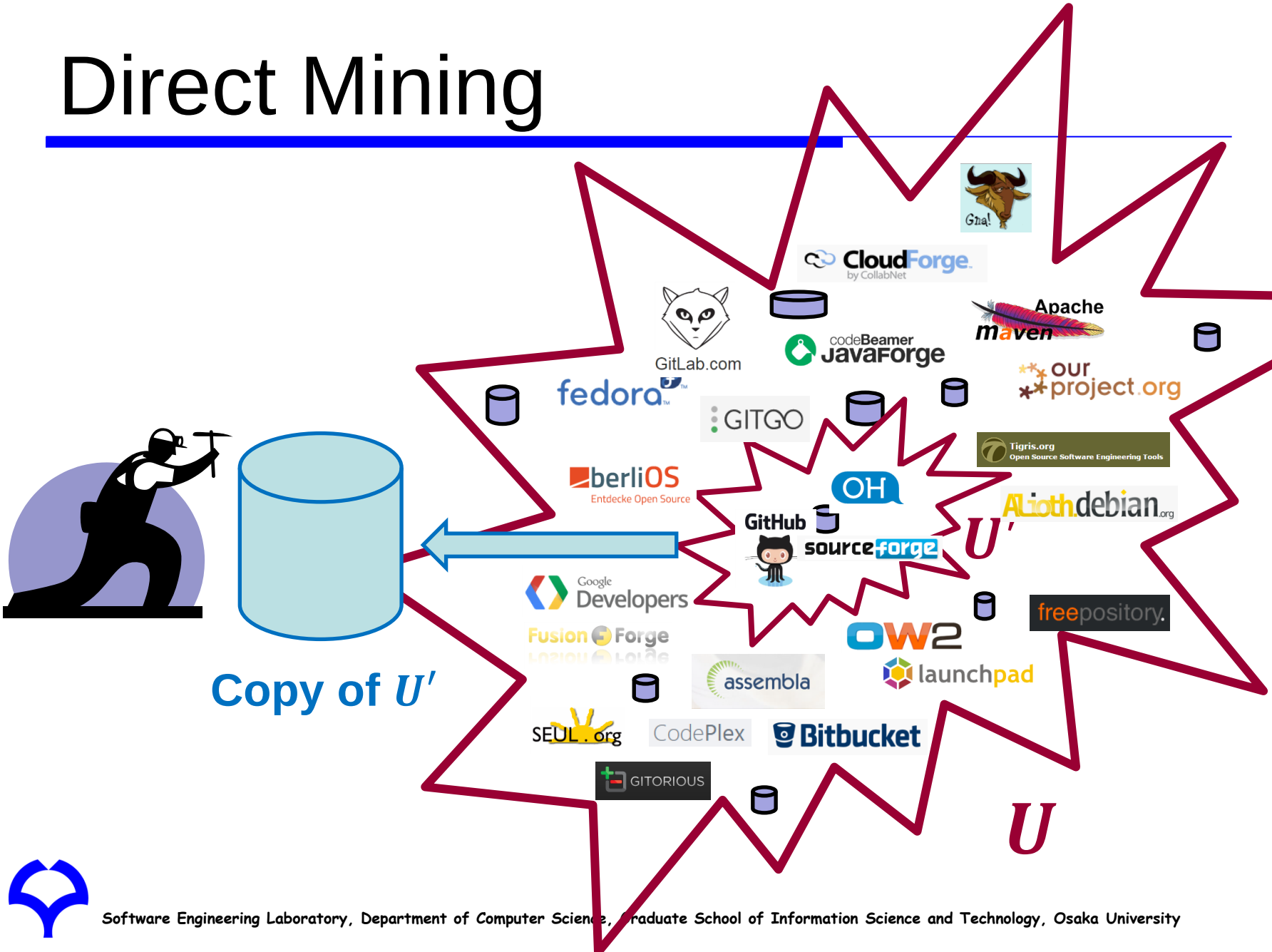
- Simple metrics of U over history
  - Size $|U|_{t1}, |U|_{t2}, \ldots$
  - Language usage

  …
- Density of U with respect to P
- History and evolution of code c in U
  - Origin version of c
  - Closely related code c' (clone, variation, family, ...)
  - Future prediction for c

C) How Do We Mine U (U')?

1. Direct mining
   - Good model
   - Powerful machine
2. Indirect mining
   - Use external services
   - Reconstruct mining result from those external services

Direct Mining

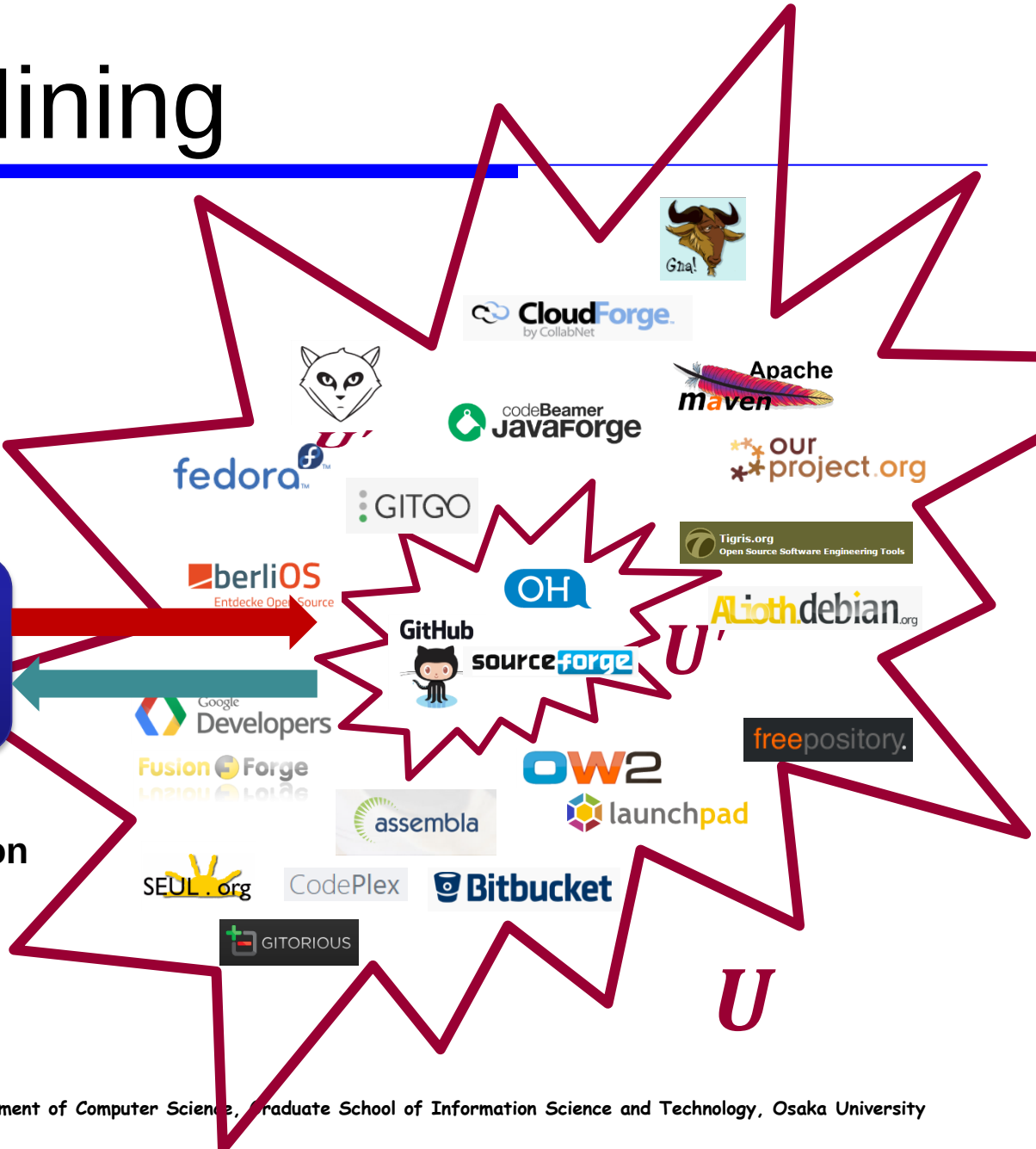


Indirect Mining

Want to know about U'

Mashup Engine

Query Decomposition and Result Composition



D) Why Do We Mine U?

Objectives of mining U

- Reuse and knowledge transfer
  - We do not want to reinvent the *wheel*
- Historical Archive
  - Frontier's wisdom

...

Discussion!

- Is it interesting research topics?
- Can we get useful research results?
- Is it feasible research target?

Thank you