UNIVERSITEIT GENT

FACULTY OF ENGINEERING

Co-evolution of source code and build systems

Bram Adams

Supervisors: Herman Tromp & Wolfgang De Meuter (VUB)

GH-SEL

INTEC

Ghent University
Department of Information Technology
Sint-Pietersnieuwstraat 41
9000 Ghent, Belgium
http://www.intec.ugent.be

bram.adams@ugent.be

http://users.ugent.be/~badams/makao

Research Question

Build system:
- captures file-level dependencies
  - source-level #include's
  - composition of binaries, libraries, etc.
- "intelligent" decision what to build ➜ incremental build
  - time stamp, MD5 checksum, ...
- configuration of source code and build system

build tool

configuration

Strong link with source code:
- out-of-sync ➜ no or inconsistent build
- rigid build system ➜ less freedom to restructure/refactor code

Claim: build system co-evolves with source code

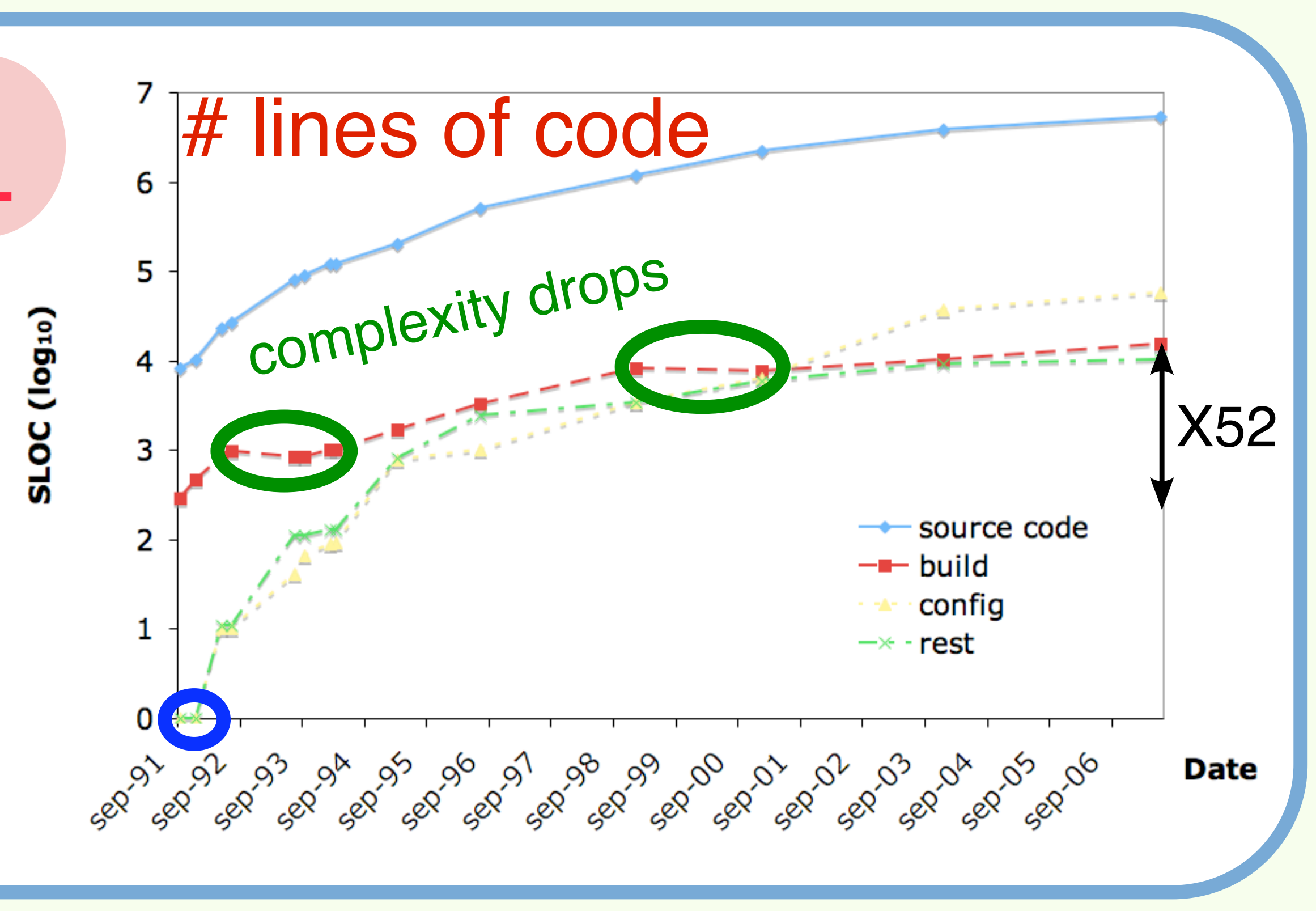
Case Study: Linux kernel

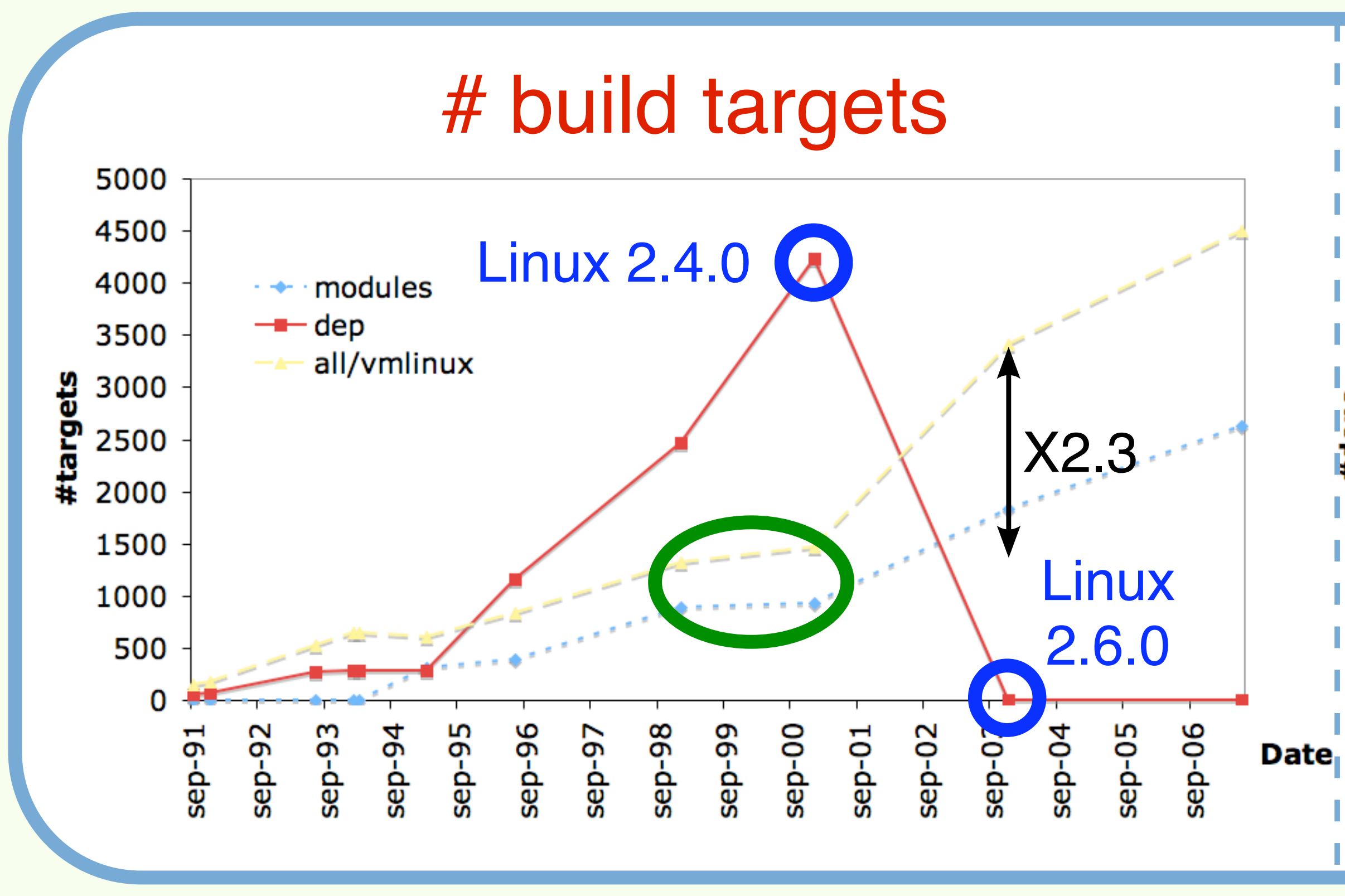
Case:
- Linux kernel from version 0.01 to 2.6.21.5

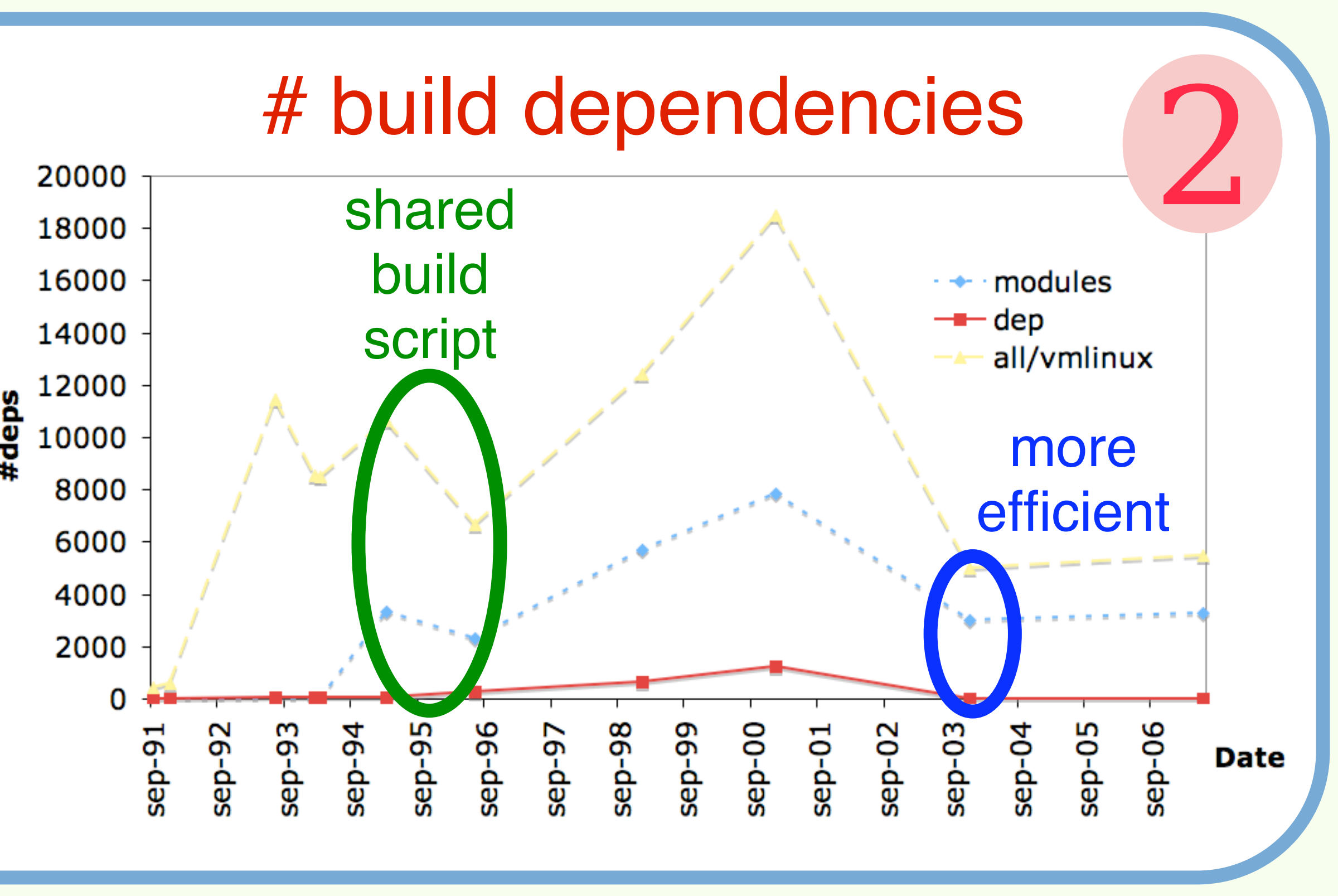
Measurements:
- physical lines of code (SLOC) of source code and build system 1
- complexity of build dependency graph 2
- detailed analysis of big evolution step in build process 3

Tools:
- SLOCCount (*http://www.dwheeler.com/sloccount/*)
- MAKAO:
  - re(verse)-engineering framework for build systems
  - dependency graph of concrete build as underlying model
  - nodes and edges are freely manipulable objects (in OO sense)

1

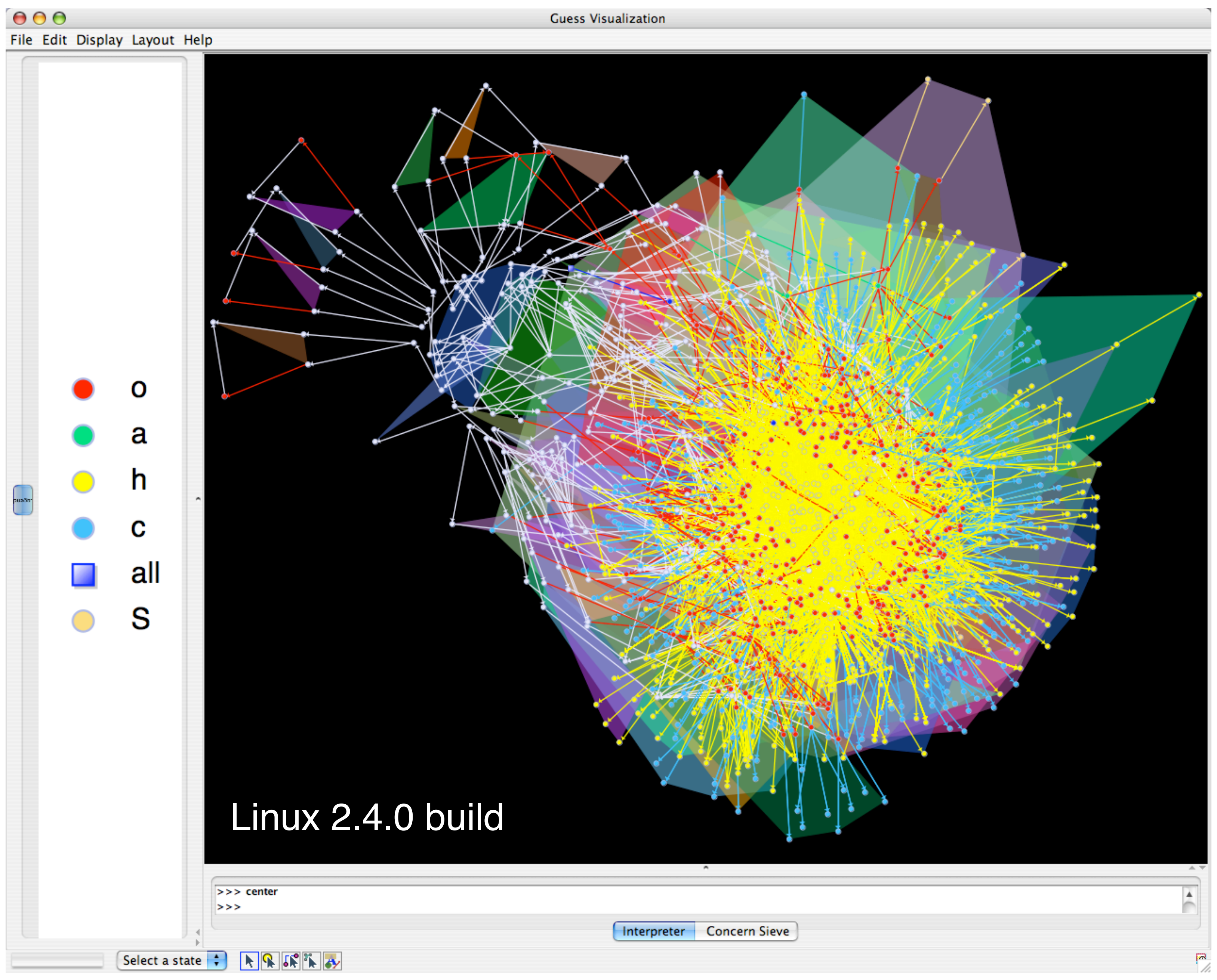






2

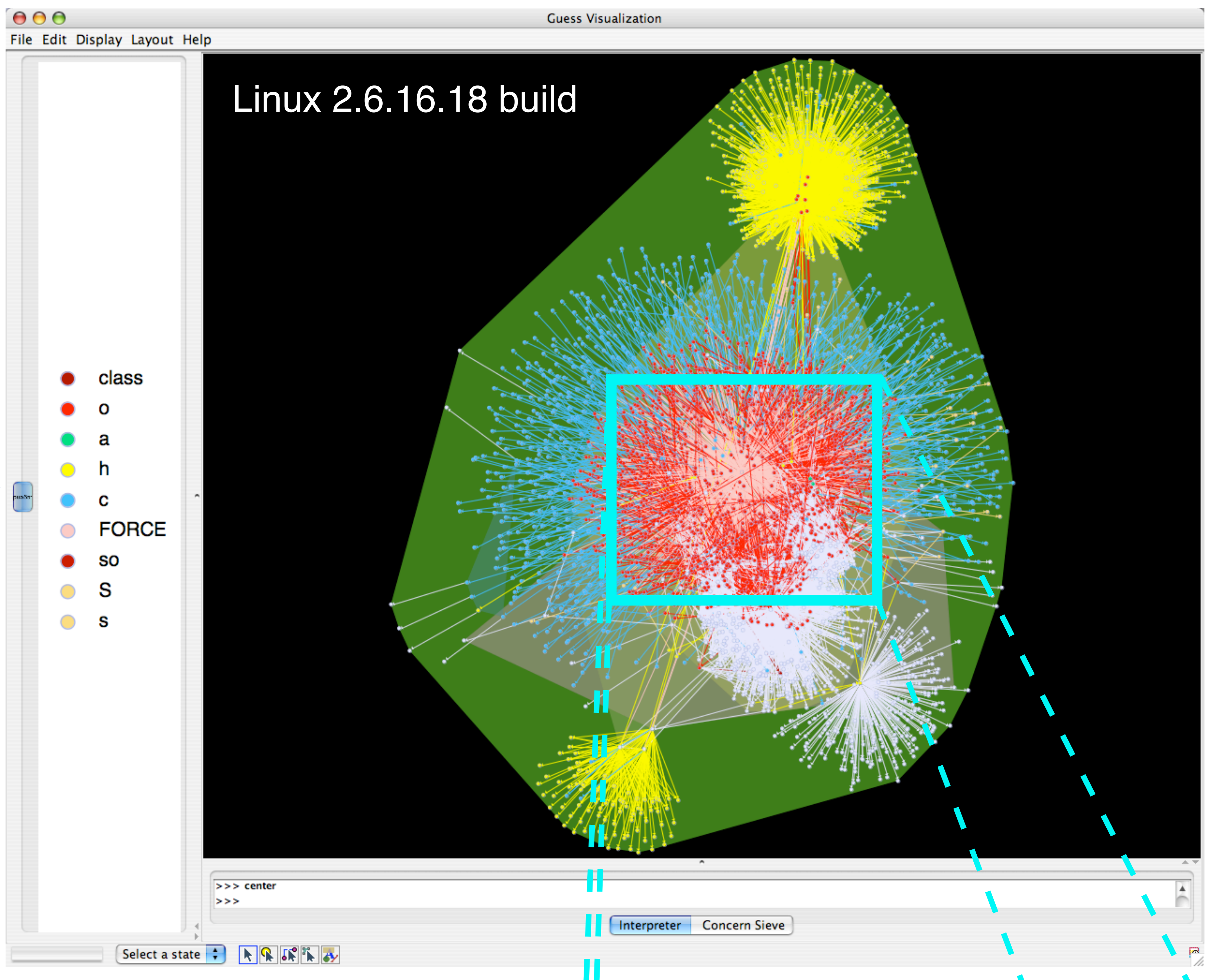
3



```
>>> center
>>>
```

big evolution step

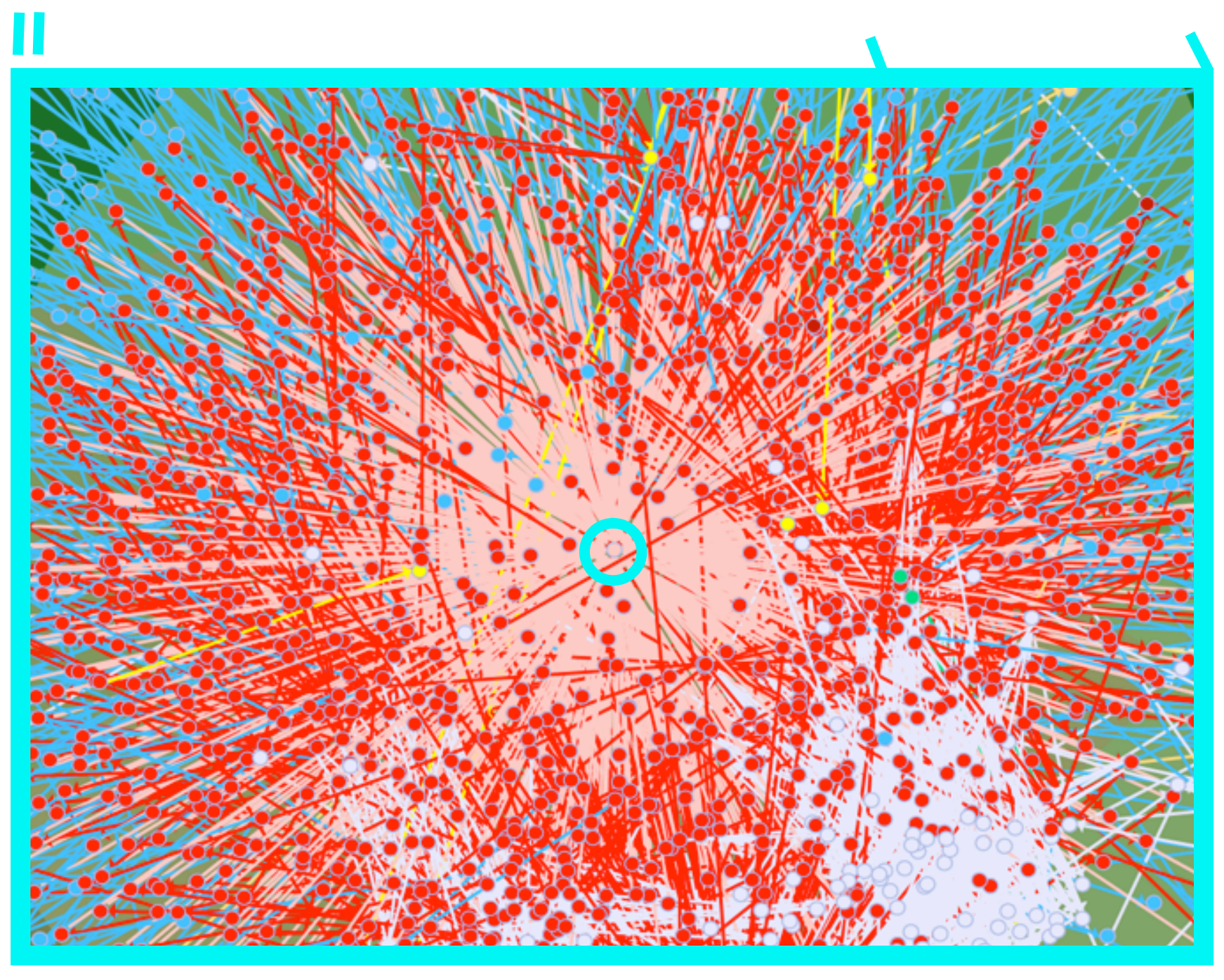
in build process

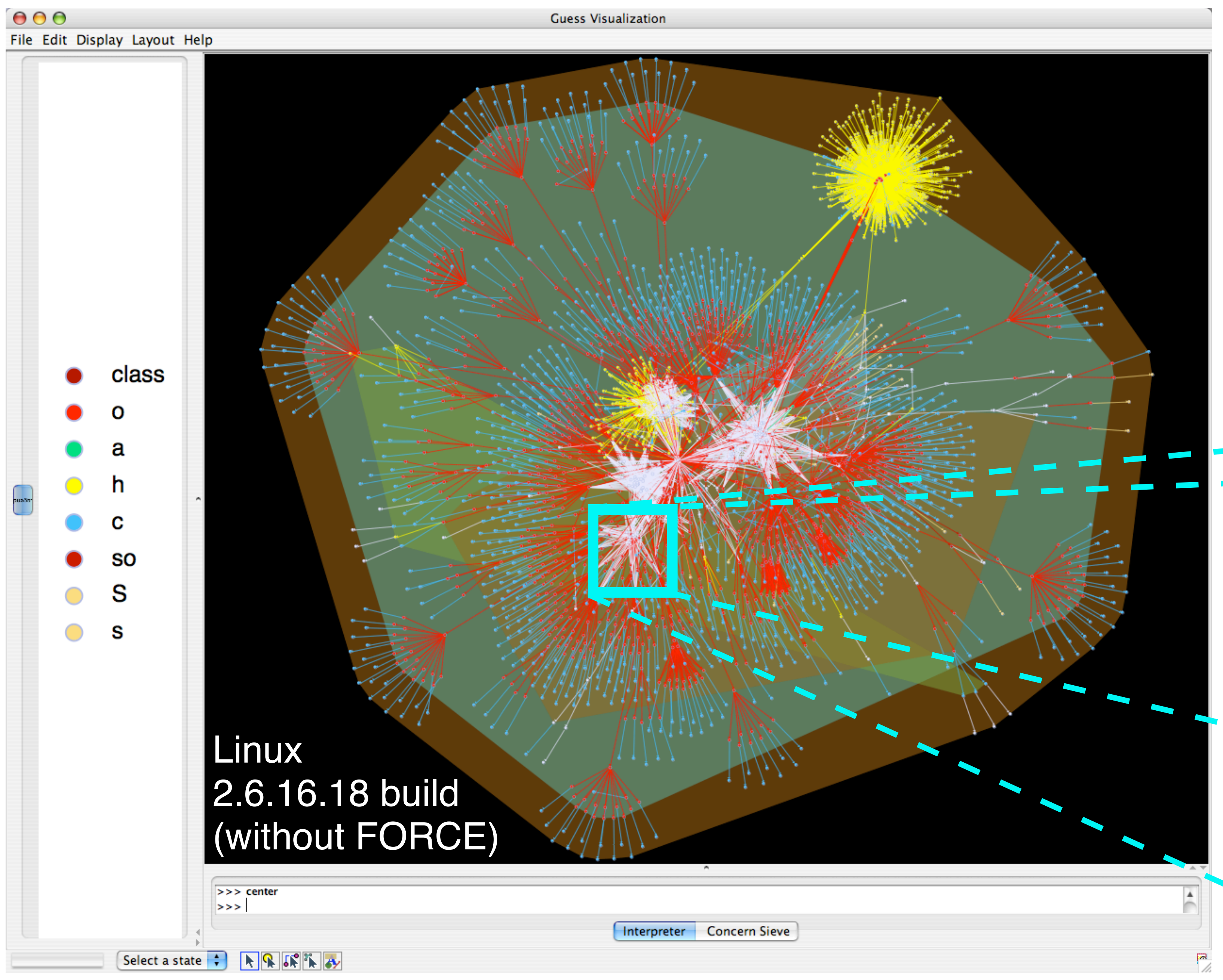


```
>>> center
>>>
```

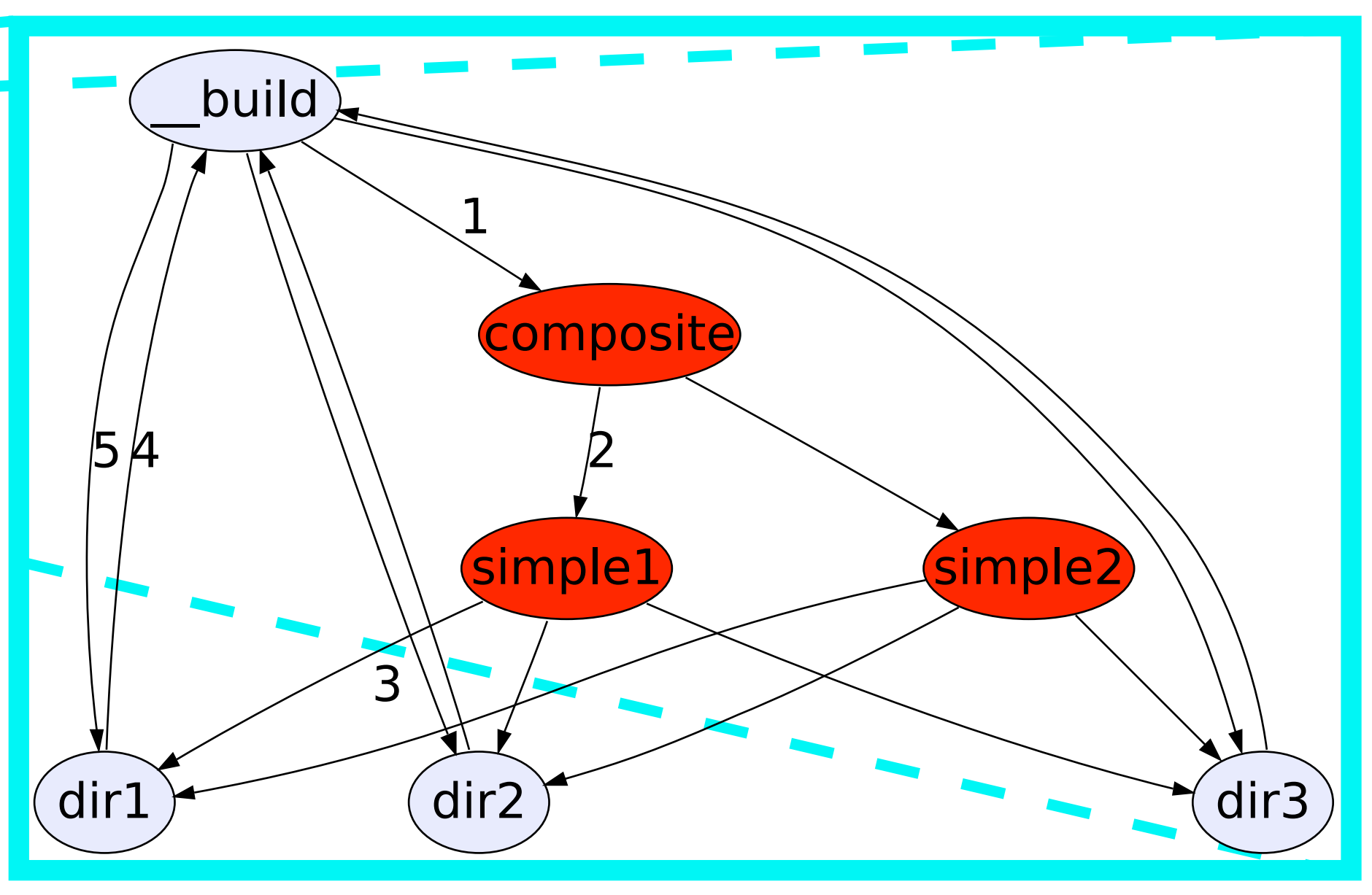
remove FORCE node

FORCE idiom for custom incremental compilation





```
>>> center
>>>
```



circular dependency chain fixes recursive make flaws

Conclusion

Lehman's laws of software evolution apply:
- build system evolves
- during evolution complexity increases
- maintenance performed to manage this

Future work:
- link with code re-engineering activities
- configuration-aware study