UNIVERSITY OF TARTU

Static Analysis for Dynamic Updates

Oleg Šelajev
@shelajev

Agenda

- Problem Introduction
  - Dynamic System Updates
  - Analysis of updates

Agenda

- Main discovery
  - Challenges to overcome
    - packaging and resource management
    - classloading simulation
  - event system
    - event opposition
  - dynamic update policies
    - HotSwap
    - LiveRebel

Agenda

- Results
  - case study
    - zt-zip
- Conclusion & Future work

Dynamic System Updates

- Updating a program without interruption
  - long running computation
  - 24 x 7 availability

Dynamic System Updates

Dynamic System Updates

- 40 years of research
  - scientific apps
  - web-scale
- Kernel / User space
- Full / partial solutions
- Process restart + state migration
- On-the-fly patching

Analysis of updates

- Almost non-existent
- Requirements are obvious & natural
- No de-facto solution in industry

Basic Questions

- What exactly is changed?
- Do we support these changes?

Theory of updates

- Application version
  - identifier
  - archive
- Update
  - old version => new version
  - static analysis vs. using runtime info

Existing DSU solutions (Java)

- HotSwap
- Play! Framework
- PROSE system
- LiveRebel

Static Analysis

- Scan archives: old vs. new
- Determine where they differ
  - Folders
    - Scan further
  - Nested archives
    - Extract and analyse it
  - Class files
    - Analyse structure and members
  - Resources

Static Analysis: details

- liverebel.xml
  - application name: apples vs. oranges
  - application version
- Mark every archive
  - modules
  - libraries

Problem 1: packaging and resources

- Zip + meta-information
  - jar
  - war
  - ear
- Complexity grows
  - more managed components
  - everything can be updated

Problem 2: classloading simulation

- Hierarchy of types
- Special components in program
  - must be updated
- Supertypes
- Must follow runtime logic

Solution 1: event system

- general archives events
- special entries events
- class level events
- method related events
- fields events
- inner classes related events

Event opposition

- new class added
  - class removed
- method body change
  - method body change

Dynamic update policies

- Every DSU solution is different
- Compatible
  - Compatible with warnings
- Incompatible

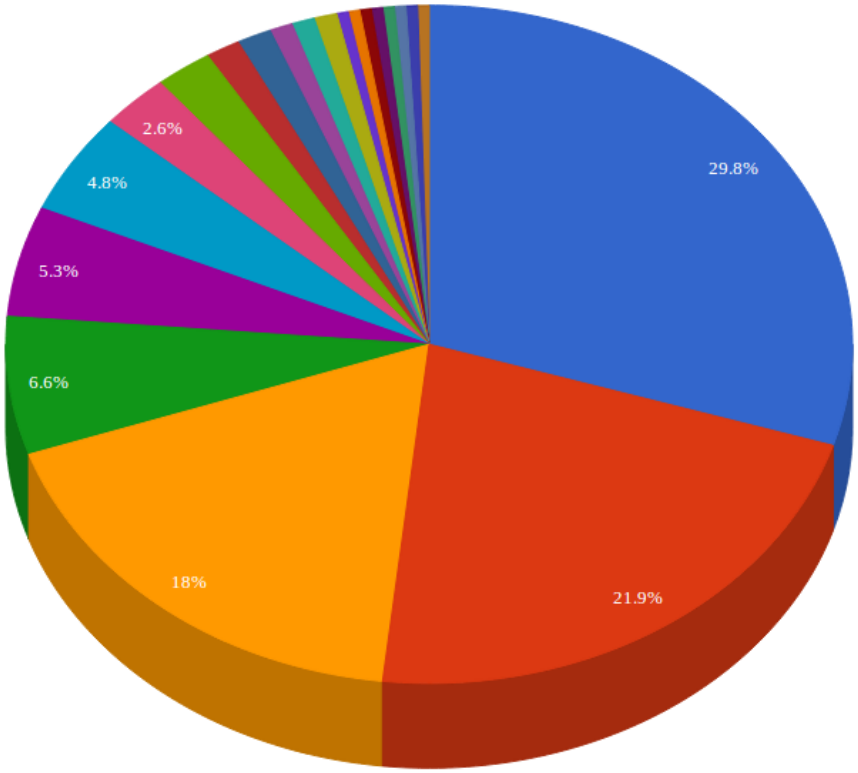
Case study: HotSwap vs. LiveRebel

Changes to Class structure		
Changes to method bodies	✓	✓
Adding/removing Methods	✓	✗
Adding/removing constructors	✓	✗
Adding/removing fields	✓	✗
Adding/removing classes	✓	✗
Adding/removing annotations	✓	✗
Changing static field value	✓	✗
Adding/removing enum values	✓	✗
Changing interfaces	✓	✗
Replacing superclass	✗	✗
Adding/removing implemented interfaces	✗	✗

Case study: results

- Real world projects:
  - zt-zip
    - OSS library for zip manipulations
    - 50KB

- Changed method body
- Changed class
- New method
- New class
- Removed method
- Changed resource
- Changed class from non-final to final
- Changed constructor body
- New constructor (Existing objects will not be affected.)
- Removed instance field
- New instance field (New instance field will not be initialized on the existing objects.)
- Removed constructor (Existing objects will not be affected.)
- Removed static field
- Anonymous inner class changed the implemented interface
- Changed method visibility from package to private
- Changed method visibility from public to private
- Changed static initializer (The static initializer will be reinvoked)
- Changed static initializer (The static initializer will not be reinvoked (but will be when changing back to the old version))
- New directory
- New static field
- New static field with constant value



case study: results

- HotSwap:
  - Compatible: 1 (out of 6)
  - Incompatible: 5 (out of 6)
- LiveRebel
  - Compatible: 3 (out of 6)
    - Compatible with warnings: 3 (out of 6)
  - Incompatible: 0

Case study: fun fact

- zt-zip 1.2 => 1.3 differences:
  - META-INF/MANIFEST.MF
  - liverebel.xml

Questions again

- What exactly is changed?
- Do we support these changes?

Conclusion

- Changelog-like diff
  - answers to what is changed
- Compatibility policy file
  - answers if update is supported

Future work

- Engine extensibility
- Analysis
  - more queries
  - understanding real world updates
  - build tools to consume output
- Runtime information