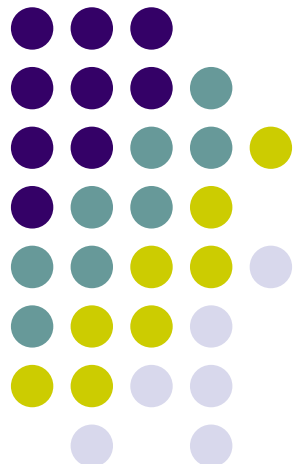
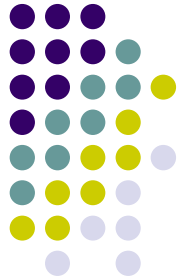
Using HotSwap for Implementing Dynamic AOP Systems

Shigeru Chiba, Yoshiki Sato
(Tokyo Institute of Technology)
Michiaki Tatsubori (IBM)

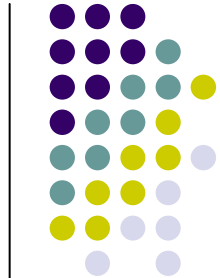


Dynamic AOP, Why?



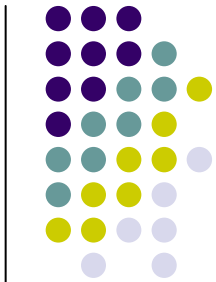
- Bug fixing by performance profiling
  - On-site
  - Real work load
  - Try-and-Error
    - Insert profiling code for monitoring particular actions, remove it, insert different profiling code, and …
  - Low performance penalty
    - Don’t ship the products with a large amount of profiling code.

Another application



- Adaptive caching for Web app. servers
  - Caching database-query results and file contents
    - Selectively for better hit ratio.
  - Crosscutting concern
    - Caching and invalidation cut across multiple classes.
  - Dynamic join points
    - Static AOP should be slow.
    - Runtime contexts turns caching on and off.

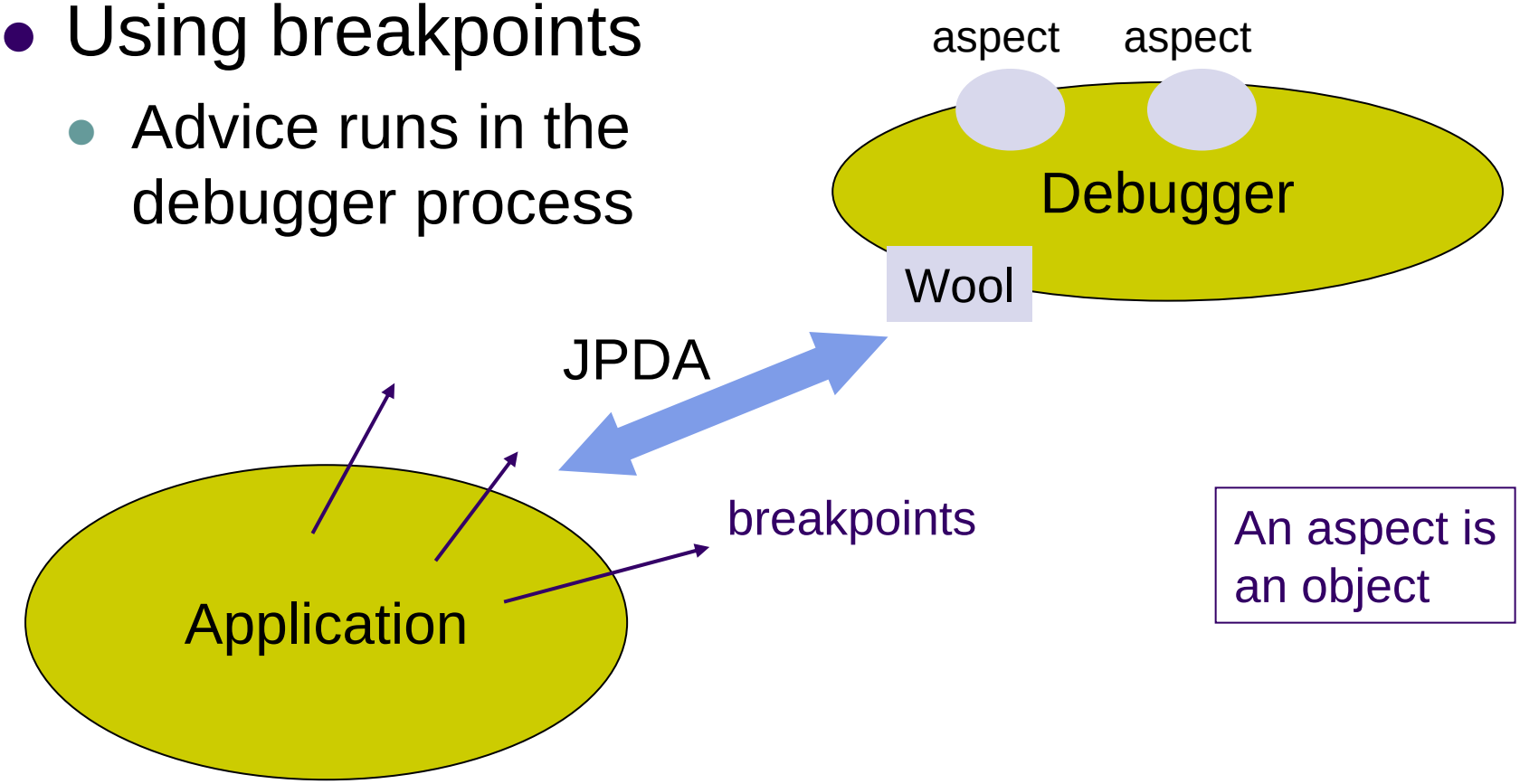
Wool



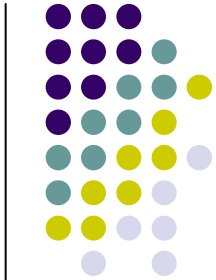
- Java Dynamic AOP system
  - Using the Java Platform Debugger Architecture (JPDA) of JDK 1.4
  - Intercepting method invocation by:
    - Breakpoints
    - HotSwap
      to replace a method including hooks for interception
  - Wool is a Java class library.
    No AOP language.
    No introduction.

First stage

- Using breakpoints
  - Advice runs in the debugger process



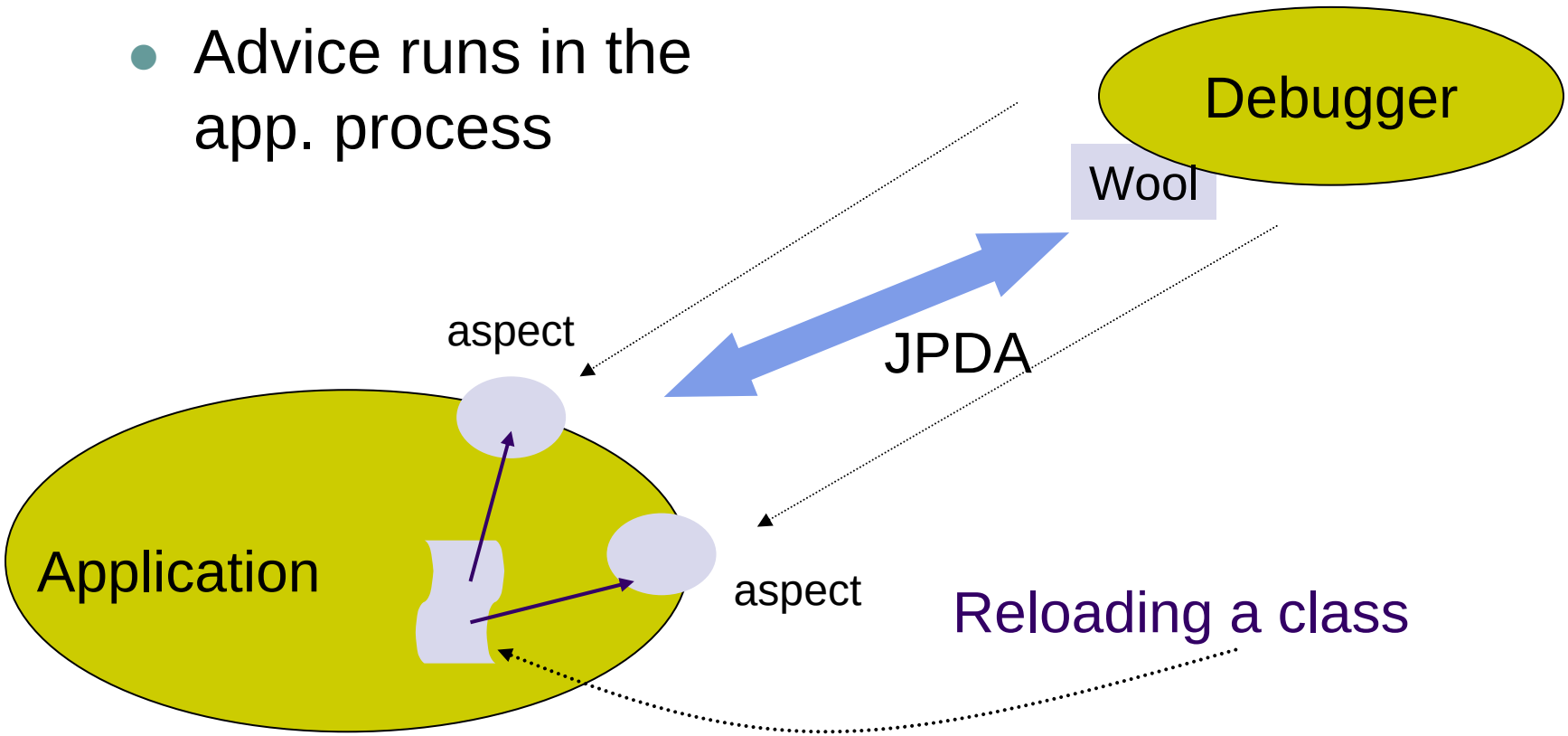
Joinpoint object



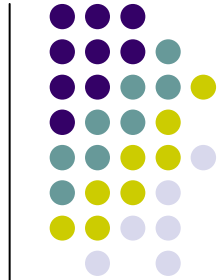
- A parameter to advice
- It represents the runtime contexts of the join point.
  - Method prameters, target object, etc…
- Abstraction for accessing the application process from the debugger process.

Second stage

- Using HotSwap
  - Advice runs in the app. process



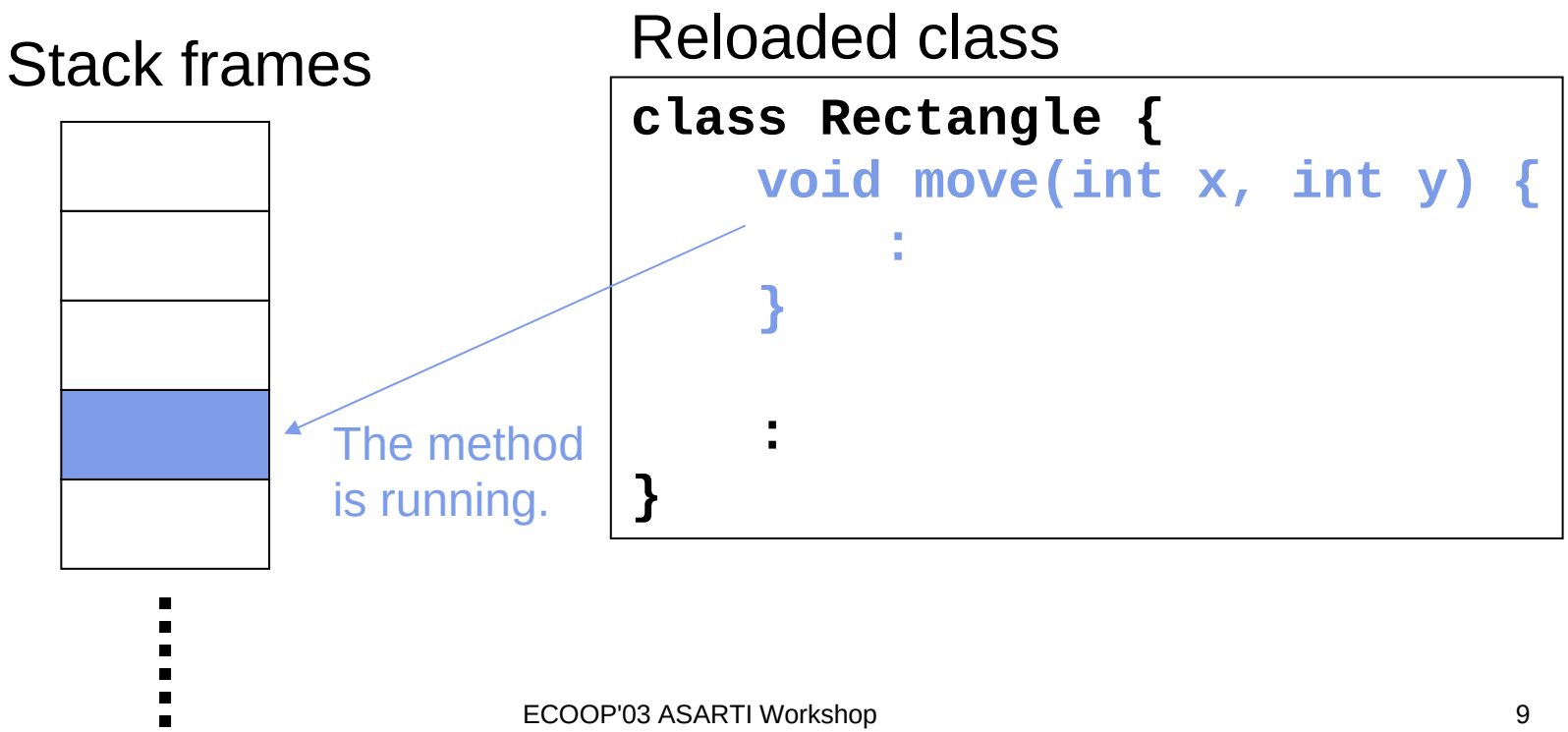
Using both breakpoint and HotSwap, Why?



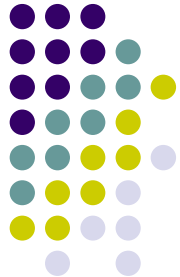
- Breakpoint
  - Installation is fast.
  - Every interception takes long time.
- HotSwap
  - Reloading a class takes long time.
  - Every interception is fast.

Also, …

- HotSwap does not allow reloading a class if …



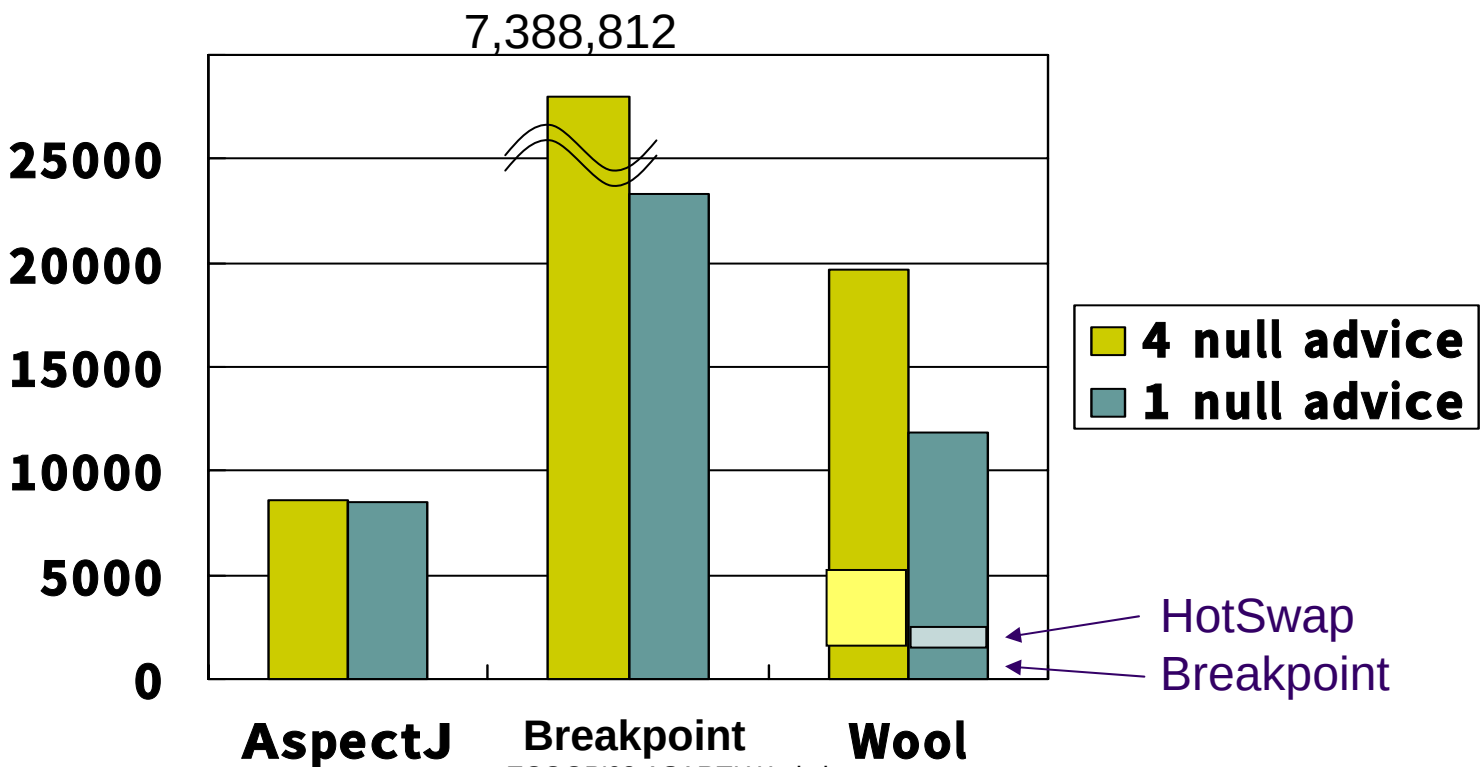
Pair of before&after advice



- They must be woven simultaneously.
  - Wool allows the users to control this.
- Example.
  - Before advice records current time.
  - After advice records the elapsed time.
    - The after advice does not work if the before advice has not been installed.

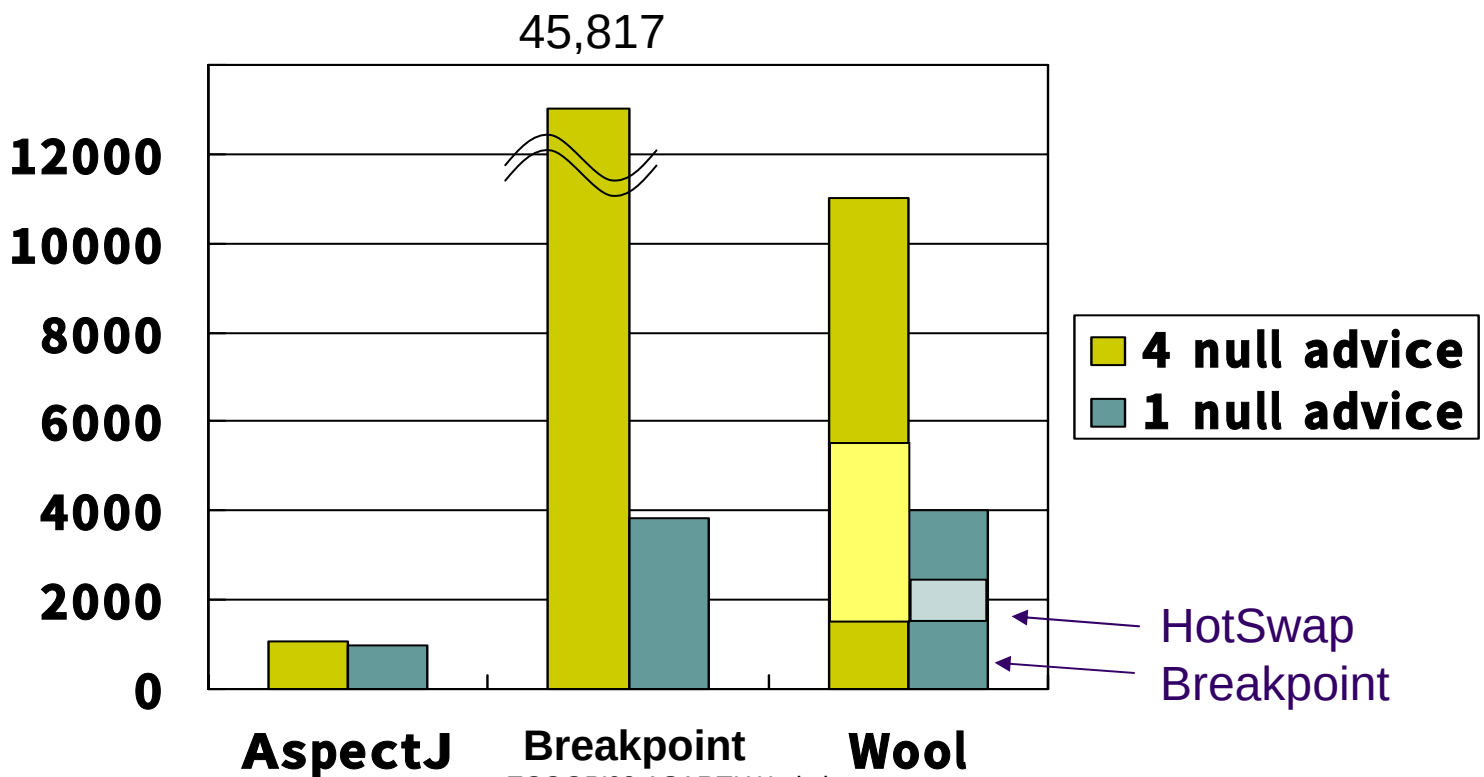
Jess in SPECjvm98

- Elapsed time (Number puzzle)

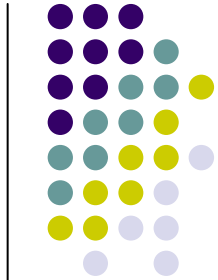


Jess in SPECjvm98

- Elapsed time (Monkey banana)



Summary



- Wool
  - Java-based Dynamic AOP system using JPDA
    - Mixing Breakpoints and HotSwap
    - No need to use a custom JVM
  - Overheads due to dynamic AOP is not negligible against static AOP.
    - Constructing a Joinpoint object would be a major source of runtime penalty.