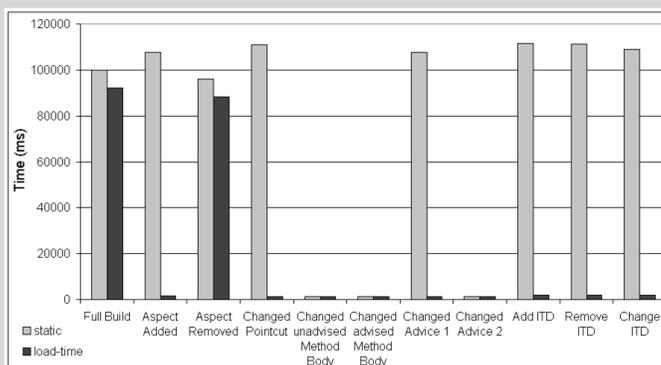


Problem: When Should You Weave?

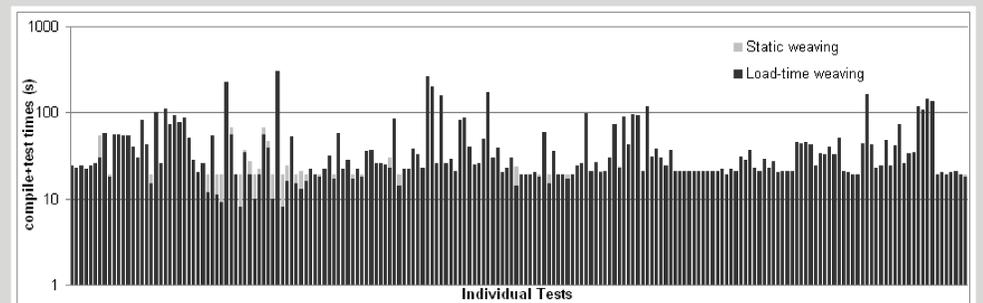
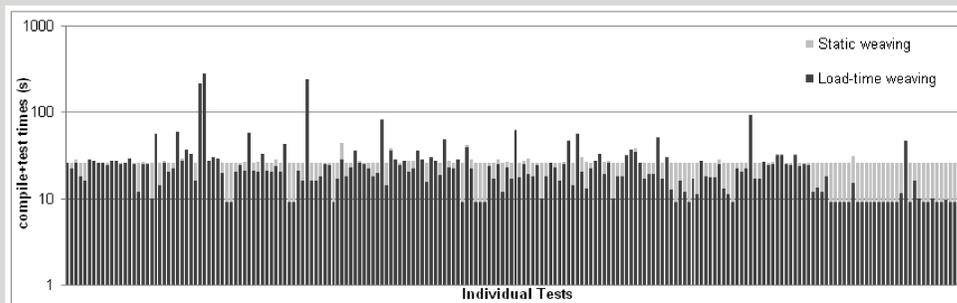
- ▶ Common development practice: make small changes to code, re-compile, run unit tests to verify the changes
- ▶ Aspect-oriented approaches offer several deployment models: static (compile-time), load-time, and run-time
- ▶ Static deployment typically induces longer (re-)compile times
- ▶ Load-time deployment typically induces longer execution times
- ▶ The impact of the deployment model on the edit+compile+test cycle is not understood: when should you weave?

Case Study I: Incremental Compilation

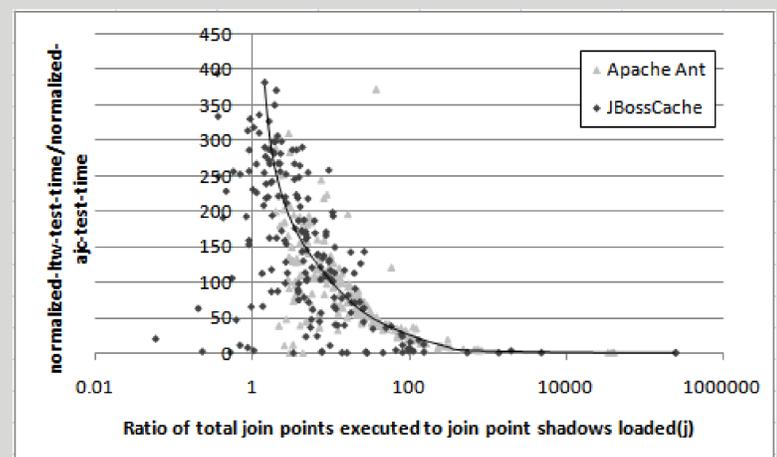
- ▶ Measured incremental compilation time of AspectJ's compiler (ajc) after small changes
- ▶ Studied large-scale systems: Azureus (2,000 source files) and Eclipse (13,000 source files)
- ▶ Results were similar for both projects: load-time deployment incurs fewer full builds compared to static deployment (1/10 versus 7/10, respectively)

Case Study II: Compile+Test Time

- ▶ For individual tests, with respect to compile+test times, load-time deployment performs better for the majority of the testcases of Ant, whereas static deployment performs better for the majority of the testcases of JBossCache which is a smaller project with larger testcases



- ▶ As the number of classes loaded increases, the test time taken by load-time deployment increases faster than static deployment
- ▶ As the number of join points executed (including repetitions) increases, the test time taken by load-time deployment decreases faster than static deployment
- ▶ For regression testing, the percentage of join points covered has no role in determining which deployment model performs better

**Contributions**

- ▶ Analysis of ajc's incremental compilation process
- ▶ A rigorous analysis of factors that contribute to the speed of a test-driven development technique using static and load-time deployment models
- ▶ Insights into scenarios where load-time deployment can be successfully used for mitigating the effects of increased AO incremental compilation time

Future Work

- ▶ Identify and measure the impact of additional parameters on the edit+compile+test cycle
- ▶ Determine the impact of the parameters when using a run-time weaving deployment model
- ▶ Build a tool that will use program analysis techniques to estimate which deployment model is best suited for a specific project+test suite