

# Accelerating Maven by Delaying Dependencies

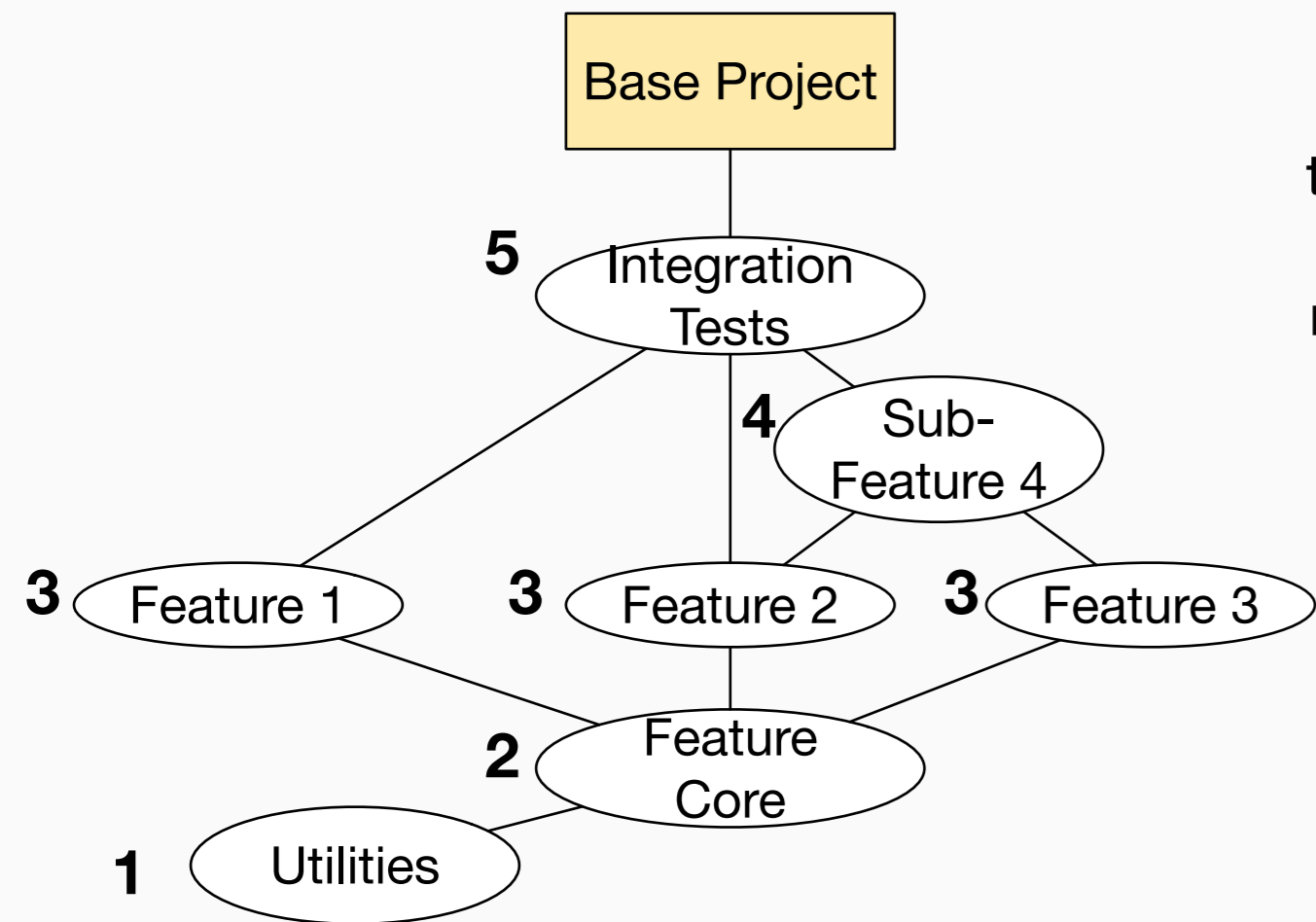
Jonathan Bell\*, Eric Melski+, Gail Kaiser\* and Mohan Dattatreya+

\*Computer Science Department, Columbia University

+Electric Cloud, Inc

## The Problem: Parallel builds can hit a granularity wall

### Maven builds often have many modules, limiting parallelism



This complicated web of module dependencies is typical of maven projects. The project itself is composed of many modules (each of which may represent some feature), each of which depend on some core functionality, such as utilities. An ideal build order is shown. At best, we can only build 3 modules in parallel.

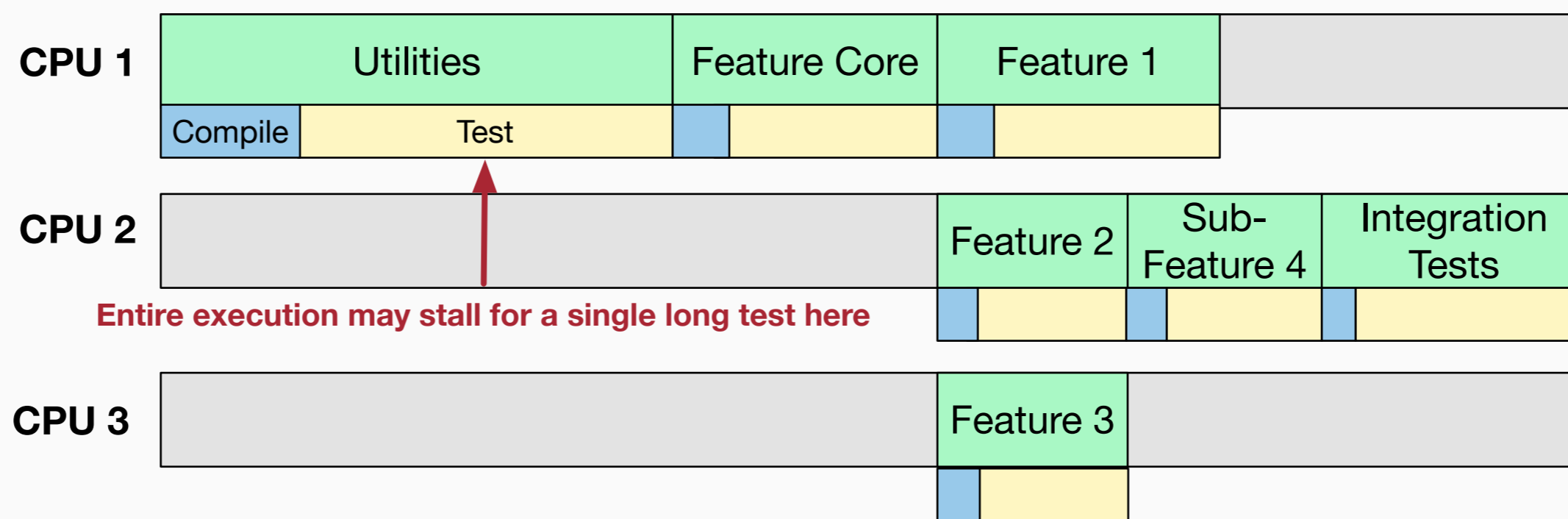
### Many modules contain unit tests, which dominate build time

With test cases split across many modules, parallelizing them (within a module) may not be very effective if they are imbalanced in execution time.

Project	Built Time (Mins)	Testing Time	Modules w/ tests
titan	380.77	95%	13/15
camel	359.57	85%	195/271
mule	198.87	93%	57/72
spring-data-mongodb	123.17	99%	3/3
cdap	110.62	97%	19/33
hadoop	108.03	98%	27/36
opennms	120.73	77%	122/220
ks-final-milestone	124.23	71%	17/46
mongo-java-driver	74.92	99%	1/1
netty	67.63	92%	16/19

### Imbalanced module execution time can restrict build parallelism

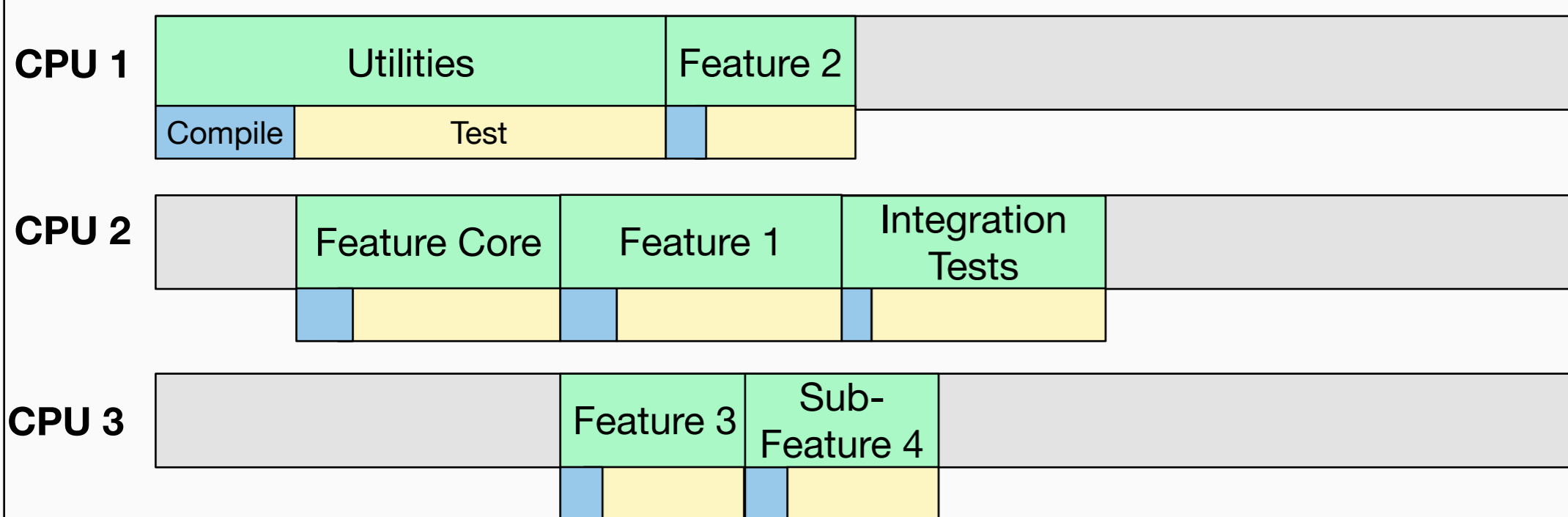
When parallelizing modules across multiple CPUs, we may experience pipeline stalls as we wait for long tests to complete from each module.



## Our Solution: Delayed Maven Dependencies

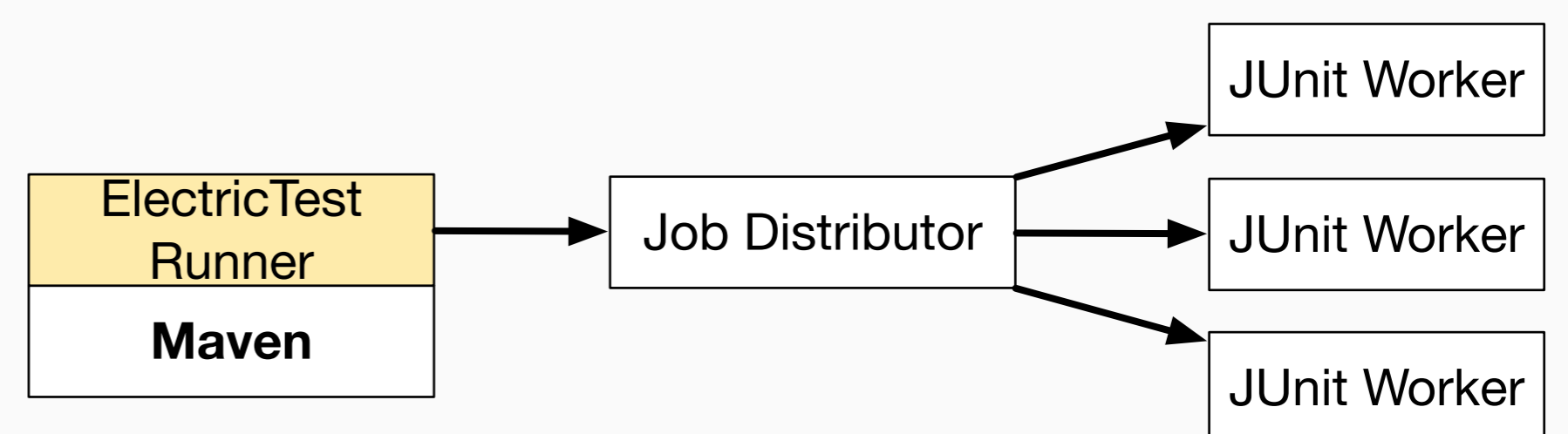
### Modify Maven to ignore dependencies on test execution

Since test execution in Maven typically does not produce artifacts that can be depended on by later modules, there is no need to wait for a module's test to finish running before beginning to build other modules that depend on it.



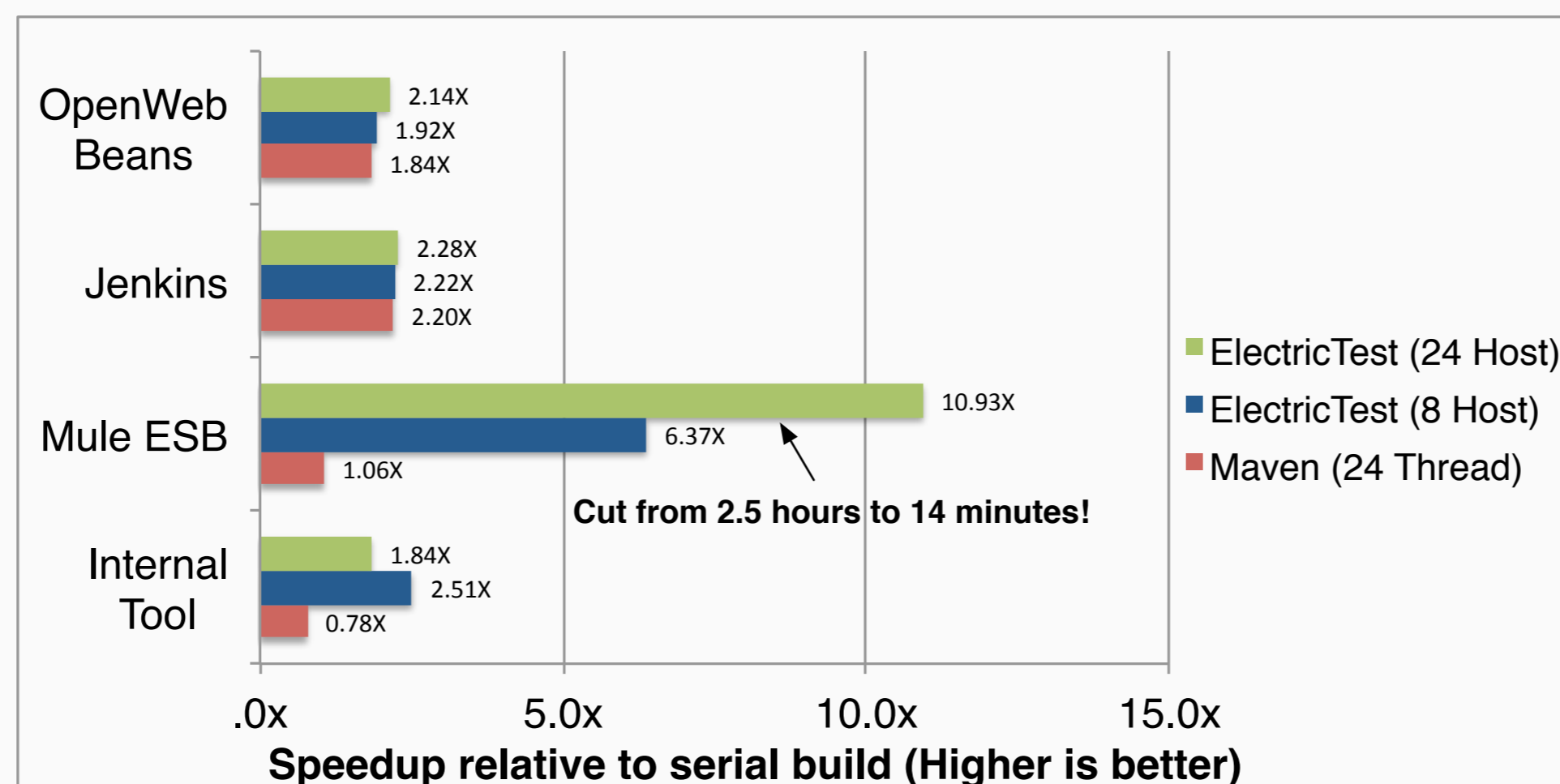
### Delaying dependencies without invasive changes to Maven

Maven allows for plug-in test runners. Our plug-in test runner runs tests in parallel, and asynchronously. Tests are shipped to a job distributor (running in a separate process) and the ElectricTest runner immediately returns to Maven, indicating that the build should continue. Test results are compiled at the end of the (complete) build.



All tests can run in parallel

### This approach can dramatically reduce build times



When comparing the speedups provided by our approach to the parallelism built-in to Maven, we found our approach significantly faster even when using the same number of compute nodes. Even in projects with few modules (OpenWebBeans and Jenkins), ElectricTest still shows a small improvement.

