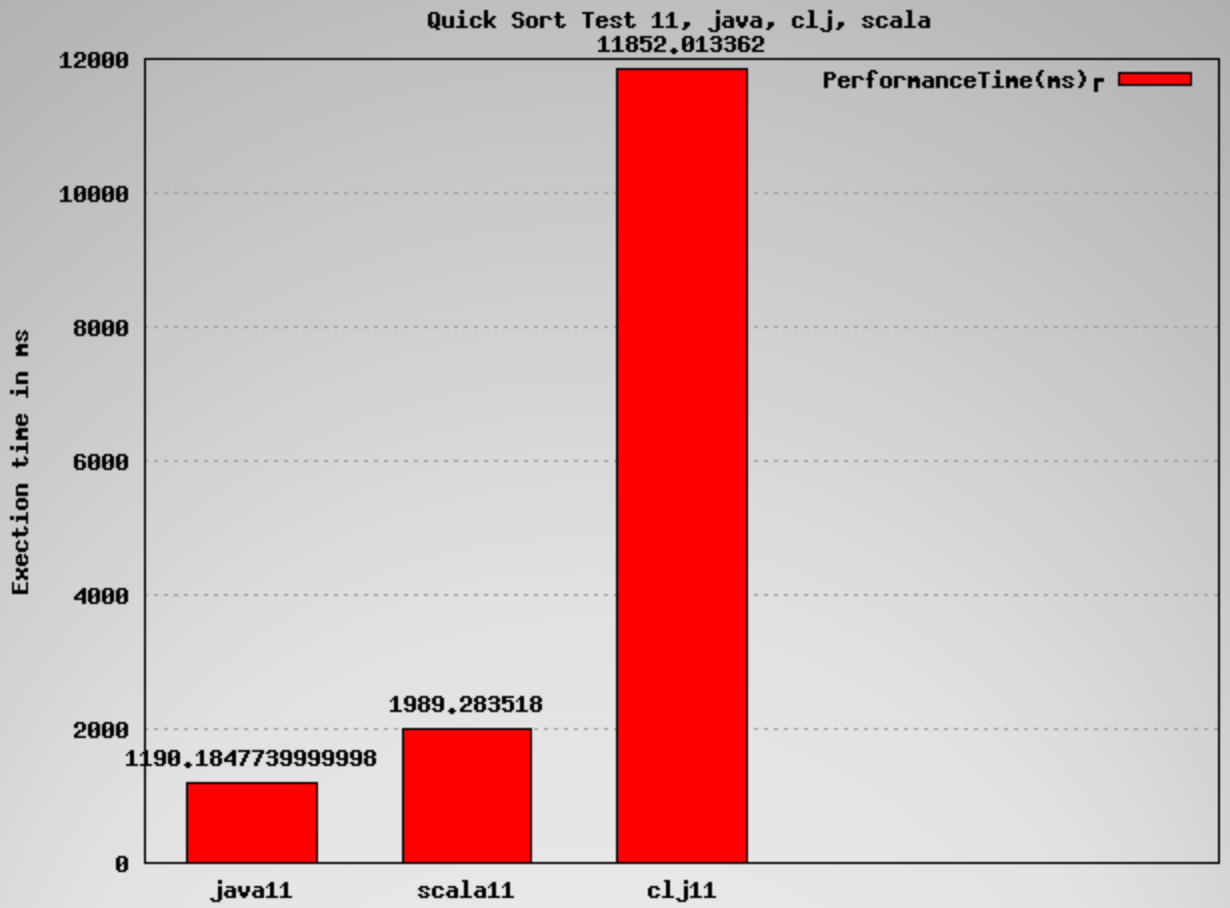
Team Team Lambda

Introducing Clojure

Clojure Is...

- Lisp dialect
- Declarative and dynamic
- Built on JVM
- Designed for concurrency



Clojure Performance

http://berlinbrowndev.blogspot.com/2009/07/jvm-notebook-basic-clojure-java-and-jvm.html

Haskell	Clojure
1 + 2	(+ 1 2)
[]	() or (list) or '()
[1,2,3]	(list 1 2 3) or '(1 2 3)
head ls	(first ls)
tail ls	(rest ls)
head (tail ls)	(second ls)
map fn ls	(map fn ls)
"Hello, " ++ "World!"	(str "Hello, " "World!")
<- readFile "test.txt"	(slurp "test.txt")
writeFile "test.txt" "Howdy"	(spit "test.txt" "Howdy")

Clojure Basics

Clojure Concurrency

- Agents store the concurrent state
- Use “send” to call a function with an agent in a new thread
- Use “await” to wait for a dispatched thread to finish

```
(let [count (agent 1)]
          (send a increment) ; increment defined elsewhere
          (await count))
```

- Use “dosync” when reading/writing shared memory