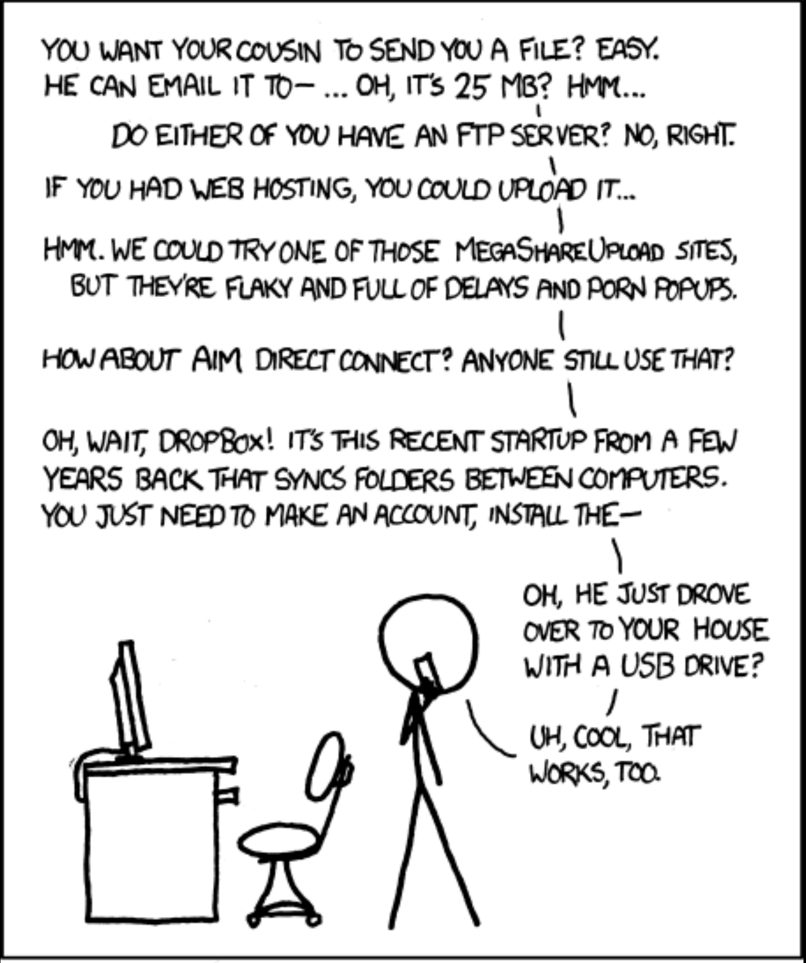

YOU WANT YOUR COUSIN TO SEND YOU A FILE? EASY. HE CAN EMAIL IT TO— ... OH, IT'S 25 MB? HMM...
DO EITHER OF YOU HAVE AN FTP SERVER? NO, RIGHT.
IF YOU HAD WEB HOSTING, YOU COULD UPLOAD IT...
HMM. WE COULD TRY ONE OF THOSE MEGASHAREUPLOAD SITES, BUT THEY'RE FLAKY AND FULL OF DELAYS AND PORN POPUPS.
HOW ABOUT AIM DIRECT CONNECT? ANYONE STILL USE THAT?
OH, WAIT, DROPBOX! IT'S THIS RECENT STARTUP FROM A FEW YEARS BACK THAT SYNCS FOLDERS BETWEEN COMPUTERS. YOU JUST NEED TO MAKE AN ACCOUNT, INSTALL THE—
OH, HE JUST DROVE OVER TO YOUR HOUSE WITH A USB DRIVE?
UH, COOL, THAT WORKS, TOO.

I LIKE HOW WE'VE HAD THE INTERNET FOR DECADES, YET "SENDING FILES" IS SOMETHING EARLY ADOPTERS ARE STILL FIGURING OUT HOW TO DO.

Building a Scala IO Library for Clarity

Tim Hausler & Suvamsh Shivaprasad

Clarity needs to do IO

Existing options:

- Java
- Scala

Java Options

- Java 6 IO:
  - General purpose standard IO library
  - java.io.File
- Java 7 NIO.2 (New IO):
  - NIO released in 1.4
  - NIO.2 part of Java 7 added java.nio.file.Path class
  - Custom FileSystem

Scala Options

- Scala IO library:
  - Standard library a wrapper around Java IO
- Jesse Eichar's scalax library:
  - Solo hobby project from Switzerland
  - Attempts to mimic NIO.2 by re implementing in Java 6 IO
  - Neat features such as Path and PathSets

Motivation

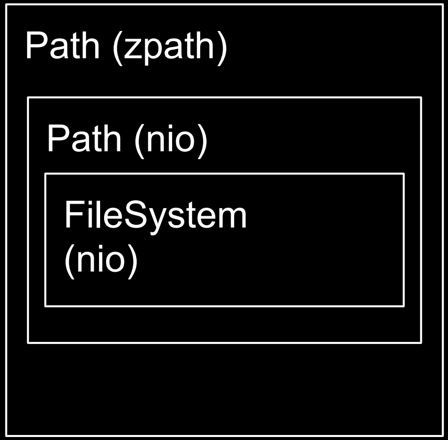
- scalax shortcomings:
  - One man army, not actively developed
  - Poor design decisions
  - Headed in a direction Clarity didn’t want
  - Time consuming to maintain
- Solution: Build custom Scala IO library

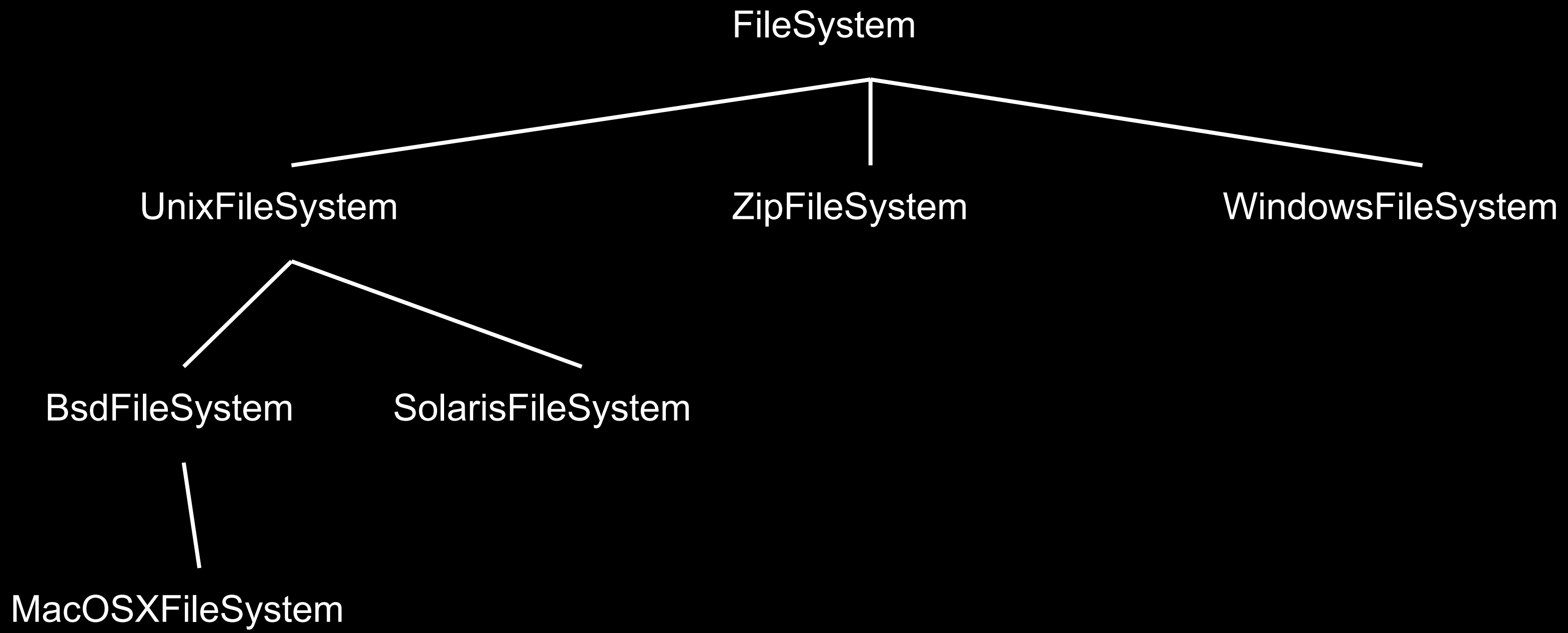
Design Goals

- Build API that mimics scalax
- Less maintenance
- Use NIO.2 underneath but provide same features as scalax
- Better performance

Top Level Classes

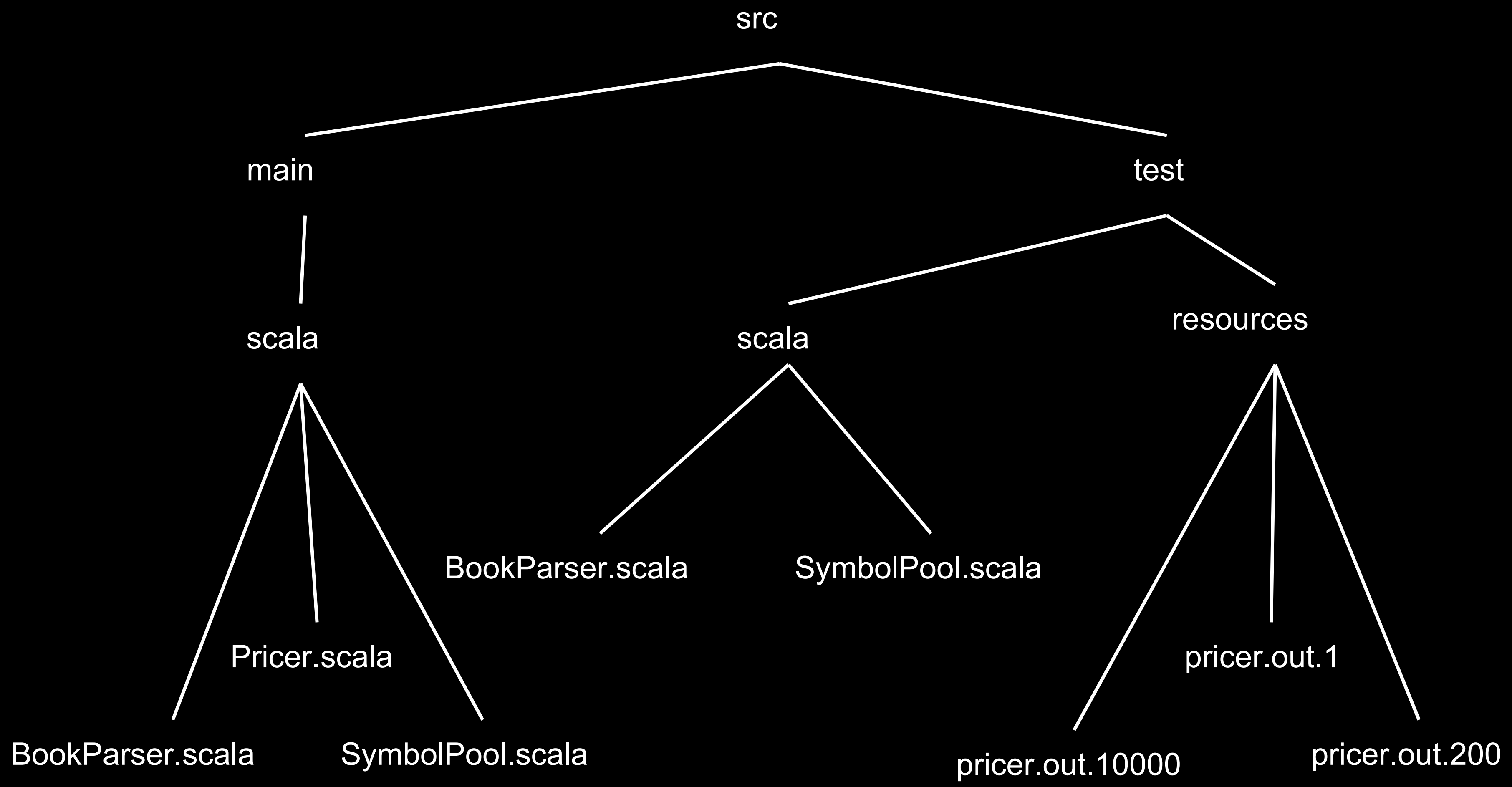
- Path
  - Wraps NIO.2 Path
- FileSystem
  - Wraps NIO.2 File System
  - Allows customization
  - Each NIO.2 Path is created from a FileSystem factory



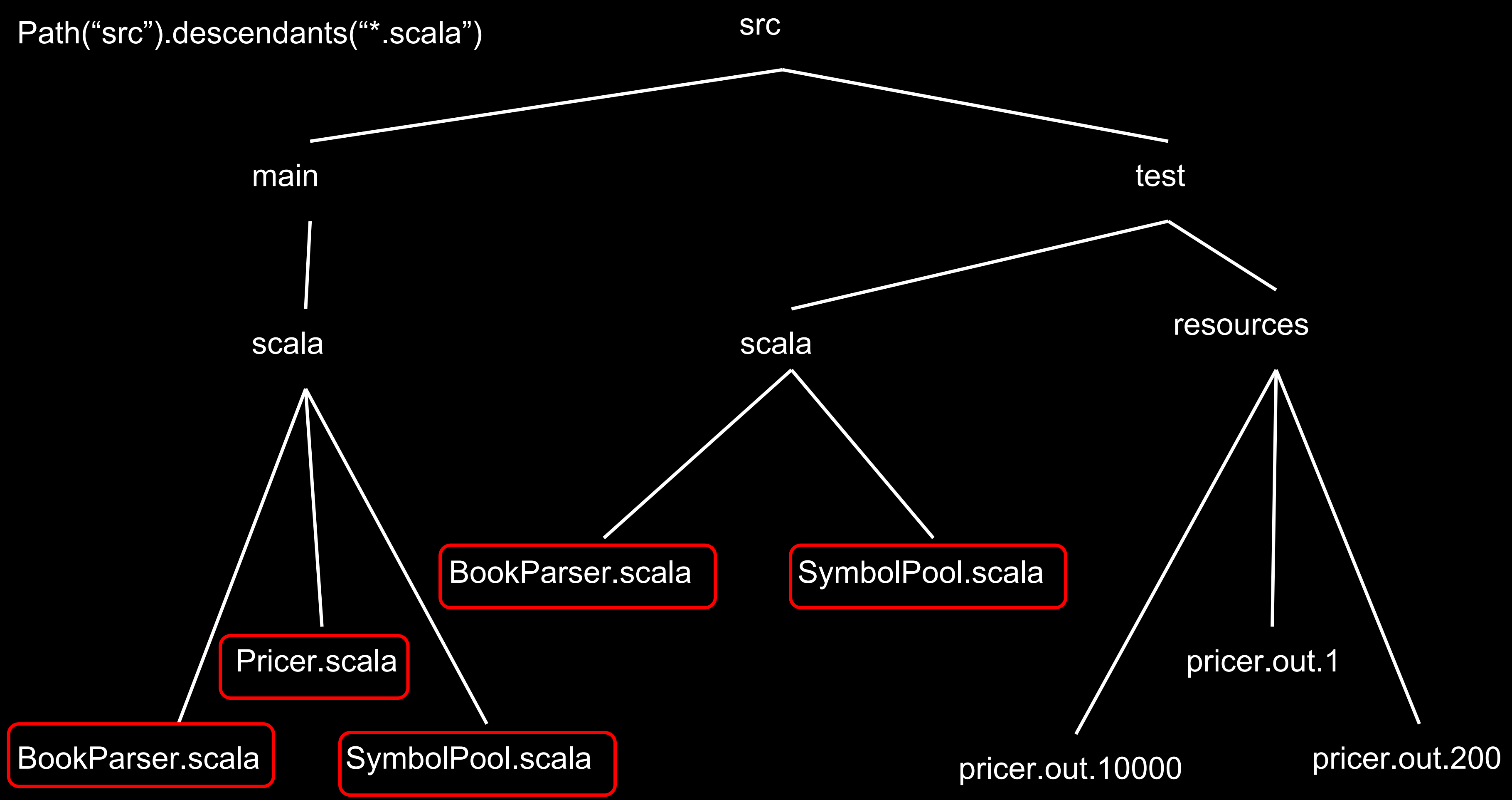

FileSystem
UnixFileSystem
ZipFileSystem
WindowsFileSystem
BsdFileSystem
SolarisFileSystem
MacOSXFileSystem

Top Level Classes (cont’d)

- PathSpec
  - Specifies a group of paths, usually in reference to a “root” path
  - PathSpecs are lazy

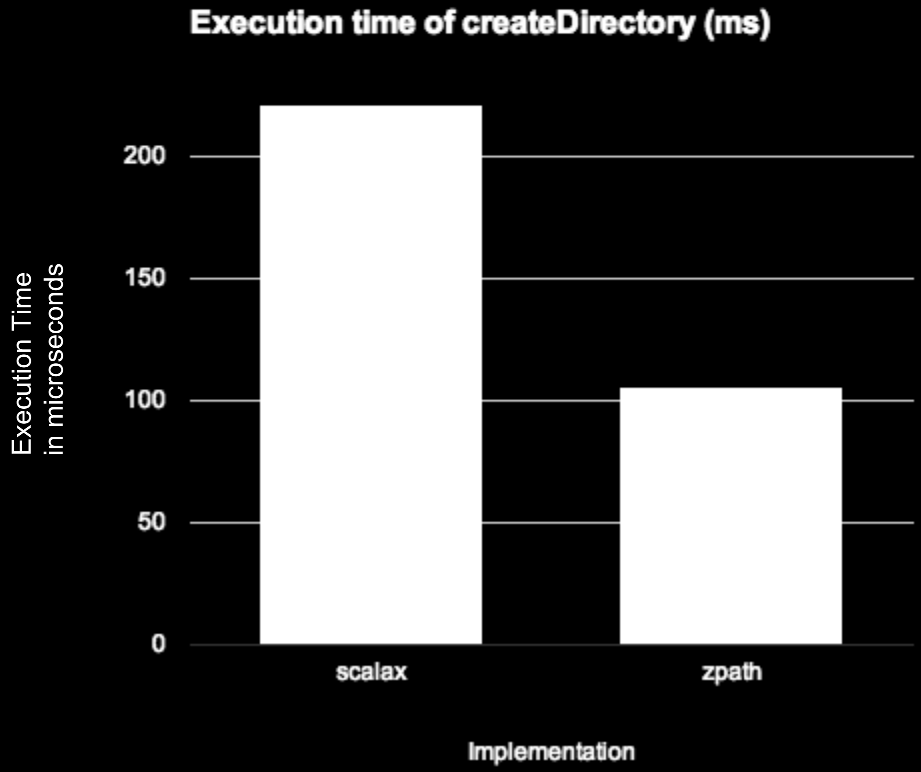

src
main
test
scala
scala
resources
BookParser.scala
SymbolPool.scala
Pricer.scala
BookParser.scala
SymbolPool.scala
pricer.out.1
pricer.out.10000
pricer.out.200

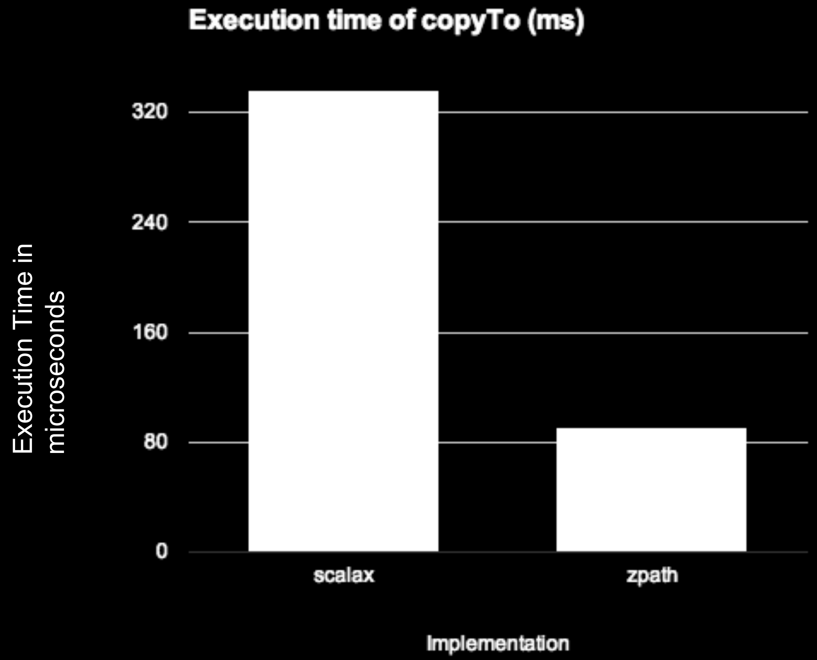
```
Path("src").descendants("*.scala")
```

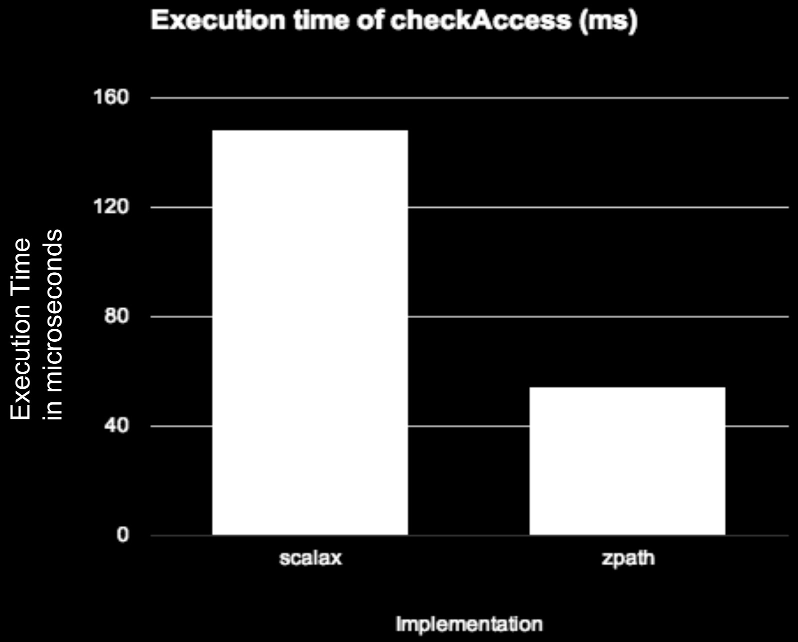


Performance Benefits

- scalax was not built with performance in mind
- We devised methods to remove redundant disk access


Execution time of createDirectory (ms)
Execution Time in microseconds
200
150
100
50
0
scalax
zpath
Implementation


Execution time of copyTo (ms)
Execution Time in microseconds
320
240
160
80
0
scalax
zpath
Implementation

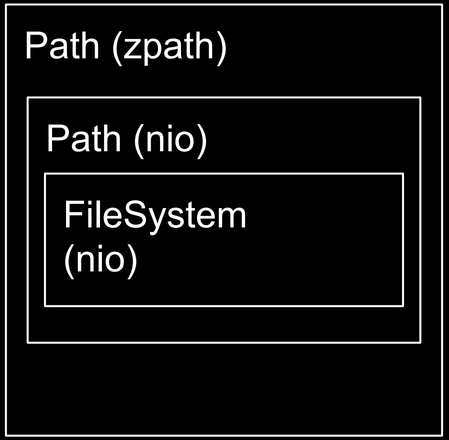

Execution time of checkAccess (ms)
Execution Time in microseconds
160
120
80
40
0
scalax
zpath
Implementation

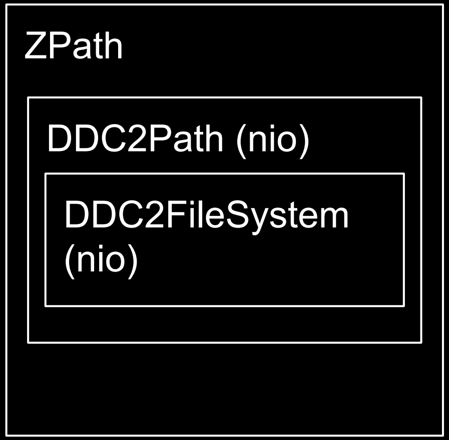
Application

- DDC2: Dynamic Data Cache 2
  - Distributed file system
  - Robust and fault tolerant
- Plug-and-play installation
  - Replaced mix of io/nio/scalax
  - Tested serialization of objects
  - Not done: testing at scale in the lab

Moving forward

- Build an nio FileSystem using DDC2
- Allows us to treat DDC2 Paths agnostically





Demo

Questions?