CSE 250
Data Structures

Dr. Eric Mikida
epmikida@buffalo.edu

Dr. Oliver Kennedy
okennedy@buffalo.edu

212 Capen Hall

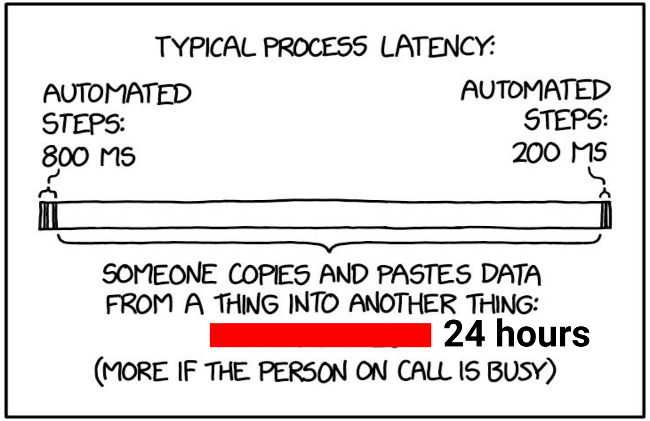
Day 03
When things go wrong (debugging and profiling)

Announcements

- AI Quiz on Autolab
  - Due Wednesday Night
- PA 0 (setup Git) on Autolab
  - Due 1 week from today
  - See Piazza for common problems
- PA 1 (Parsing CSV files in Scala) on Autolab
  - Due 2 weeks from today
  - Submissions open a week from today (or maybe sooner, up to you…)
  - Start Early!

Github + Autolab

- PA 0 allows us to connect your Autolab and Github accounts
- Help us open PA 1 early!
    - PA 1 will open once %90 of you (from both A and B) submit PA 0
    - Once PA 1 opens, allow 24 hours after submitting PA 0 to submit PA 1


TYPICAL PROCESS LATENCY:
AUTOMATED STEPS: 800 MS
AUTOMATED STEPS: 200 MS
SOMEONE COPIES AND PASTES DATA FROM A THING INTO ANOTHER THING:
[redacted] 24 hours
(MORE IF THE PERSON ON CALL IS BUSY)

https://xkcd.com/2565/

Notes on Submissions

- **Github Classroom**
  - You will get an invite link for each individual project
  - Following the link will create a GitHub Git repository for the project with template code
- Edit the repository code according to project specific instructions
  - Make sure to commit and push frequently
  - Create a new submission in Autolab when ready
- Requirements
  - Make sure you are using Scala 2.13.x
  - Don’t add any outside packages

Things WILL go wrong...often

Being a good computer scientist does not mean getting things 100% right all of the time. Things WILL go wrong.

A good computer scientist knows how to solve problems, and how to recover when things go wrong.

Things WILL go wrong...often

Being a good computer scientist does not mean getting things 100% right all of the time. Things WILL go wrong.

A good computer scientist knows how to solve problems, and how to recover when things go wrong.

Let's talk about some useful tools for recovering...

The REPL (read - eval - print loop)

- From IntelliJ: `Ctrl+Shift+D`
  - Highlight a line and press `Ctrl+Shift+X` to execute
  - Copy+past a line and press `Ctrl+Enter` to execute
- From the command line: `scala`
  - Paste or type commands to run them
  - Type `:help` to get a list of additional commands
- From SBT: `console`

Unit Testing

- Break the big problem into smaller problems
  - Test each small solution before combining them
- Useful for debugging
  - Sanity check each step in a large process to make sure it works
  - Separate the UI from the tests
- Useful way to encode your assumptions, constraints, etc
  - Automatic reminder if your assumptions change
  - Also acts as self-documentation

Unit Testing

- Break the big problem into smaller problems

If you're building a boat, you aren't going to build the entire thing then just throw it in the water and hope it floats…you would test throughout the whole process.

The same logic applies to your coding projects!

  - Also acts as self-documentation

Live Demo

Basic Debugging

ScalaTest

```
class HelloWorldTest extends AnyFlatSpec {
    "HelloWorld.doThings()" should "return 5" in {
    assert(HelloWorld.doThings() == 5)
  }
  it should "not return 10" in {
    assert(HelloWorld.doThings() != 10)
  }
  "HelloWorld.x" should "have type Float" in {
    assert(HelloWorld.x.isInstanceOf[Float])
  }
  "Register(0).addToValue" should "return the input value"
  in {
    val reg = Register(0)
    for (i <- 1 to 10000) {  assert(reg.addToValue(i) == i) }
  }
}
```

ScalaTest

```
class HelloWorldTest extends AnyFlatSpec {
    "HelloWorld.doThings()" should "return 5" in {
    assert(HelloWorld.doThings() == 5)
  }
  it should "not return 10" in {
    assert(HelloWorld.doThings() != 10)
  }
  "HelloWorld.x" should "have type Float" in {
    assert(HelloWorld.x.isInstanceOf[Float])
  }
  "Register(0).addToValue" should "return the input value"
  in {
    val reg = Register(0)
    for (i <- 1 to 10000) {  assert(reg.addToValue(i) == i) }
  }
}
```

Describe in “english” what the test should do

“in” defines what the test does

Confirm assumptions with asserts

Call as many asserts that you need

Live Demo

ScalaTest

Profiling

- IntelliJ -> Profilers
  - https://www.jetbrains.com/help/idea/cpu-profiler.html

- SBT -> HProf
  - https://docs.oracle.com/javase/8/docs/technotes/samples/hprof.html

Profiling

- IntelliJ -> Profilers
  - https://www.jetbrains.com/help/idea/cpu-profiler.html

- SBT -> HProf
  - https://docs.oracle.com/javase/8/docs/technotes/samples/hprof.html

```
fork in run := true
javaOptions in run += "-agentlib:hprof=cpu=samples,depth=10"
```

Profiling

- IntelliJ -> Profilers
  - https://www.jetbrains.com/help/idea/cpu-profiler.html

- SBT -> HProf
  - https://docs.oracle.com/javase/8/docs/technotes/samples/hprof.html

```
fork in run := true
javaOptions in run += “-agentlib:hprof=cpu=samples,depth=10”
```

Load HProf **Sample CPU Usage** **Stack Trace Depth**

Live Demo

Profiling with HProf

HProf Traces

```
JAVA PROFILE 1.0.1, created Fri Sep  3 02:24:46 2021

Copyright (c) 2003, 2005, Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without


…

TRACE 300207:
     scala.collection.StrictOptimizedLinearSeqOps.drop(LinearSeq.scala:261)
     scala.collection.StrictOptimizedLinearSeqOps.drop$(LinearSeq.scala:257)
     scala.collection.immutable.List.drop(List.scala:79)
     scala.collection.immutable.List.drop(List.scala:79)


…

CPU SAMPLES BEGIN (total = 185) Fri Sep  3 02:24:48 2021
rank   self  accum   count trace method
   1 44.86% 44.86%      83 300207 scala.collection.StrictOptimizedLinearSeqOps.drop
   2 35.14% 80.00%      65 300210 scala.collection.immutable.$colon$colon.tail
   3  5.95% 85.95%      11 300071 java.lang.ClassLoader.defineClass1
   4  2.16% 88.11%       4 300209 scala.collection.immutable.Range.foreach$mVc$sp
```

HProf Traces

```
JAVA PROFILE 1.0.1, created Fri Sep  3 02:24:46 2021

Copyright (c) 2003, 2005, Oracle and/or its affiliates. All rights reserved.

Redistribution and use in source and binary forms, with or without


…

TRACE 300207
     scala.collection.StrictOptimizedLinearSeqOps.drop(LinearSeq.scala:261)
     scala.collection.StrictOptimizedLinearSeqOps.drop$(LinearSeq.scala:257)
     scala.collection.immutable.List.drop(List.scala:79)
     scala.collection.immutable.List.drop(List.scala:79)


…

CPU SAMPLES BEGIN (total = 185) Fri Sep  3 02:24:48 2021
rank   self  accum   count trace method
   1 44.86% 44.86%      83 300207 scala.collection.StrictOptimizedLinearSeqOps.drop
   2 35.14% 80.00%      65 300210 scala.collection.immutable.$colon$colon.tail
   3  5.95% 85.95%      11 300071 java.lang.ClassLoader.defineClass1
   4  2.16% 88.11%       4 300209 scala.collection.immutable.Range.foreach$mVc$sp
```