

Teaching Modern Multithreading in CS2 with Actors

Mark C. Lewis - Trinity University

Lisa L. Lacher - University of Houston-Clear Lake

Course History

- Value of parallel in early CS (this audience knows this)
- Trinity has covered multithreading in CS2 since 2002
 - Originally using Java and Java threads
 - Parallel early in semester instead of late
 - Moved to Scala in 2010
 - Coverage broadened to include parallel collections, composable futures, and actors
 - Stress issues with race conditions and deadlock in lectures
 - Class is “flipped”

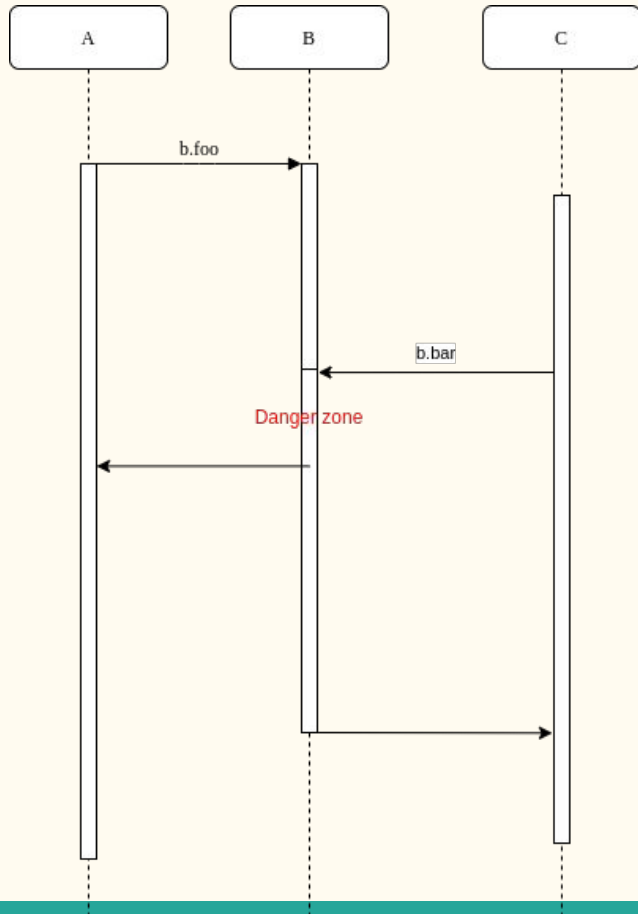
Scala Language

- Scala = Scalable Language
- Primary implementation runs on JVM, calls Java seamlessly
- Most common uses in industry are highly-parallel workloads (web and big data)
- Actors and Futures could be done in Java or C# as well

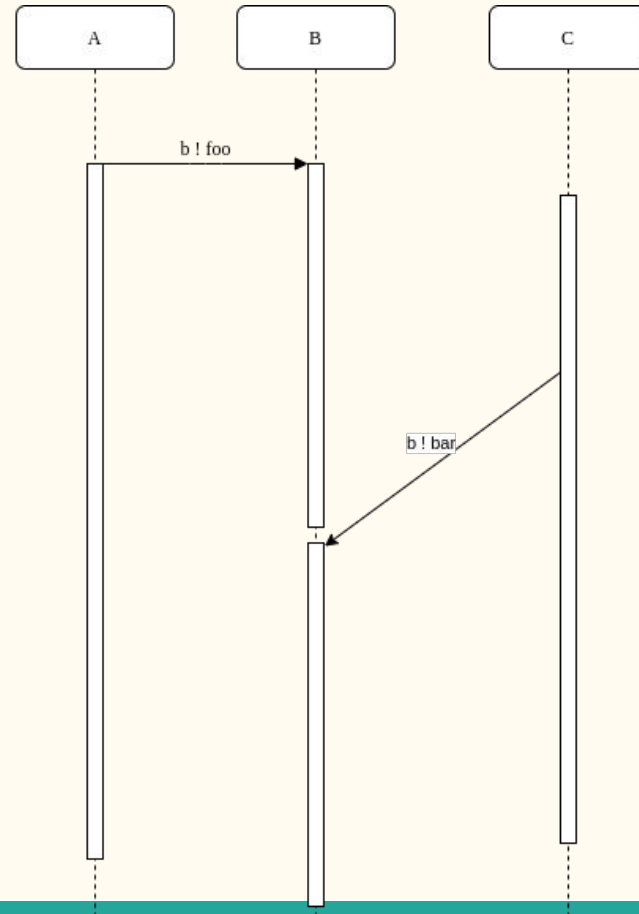
Actor Parallelism and Akka

- Actor model dates back to 1974
- Popularized by Erlang
- Akka is JVM-based actors/reactive streams/http
 - Backend of many large JVM data processing pipelines including Fortnite
- Message passing, asynchronous
- Can be remote
- Uses multiple threads on a single machine
- Enforces strong encapsulation

Objects



Actors



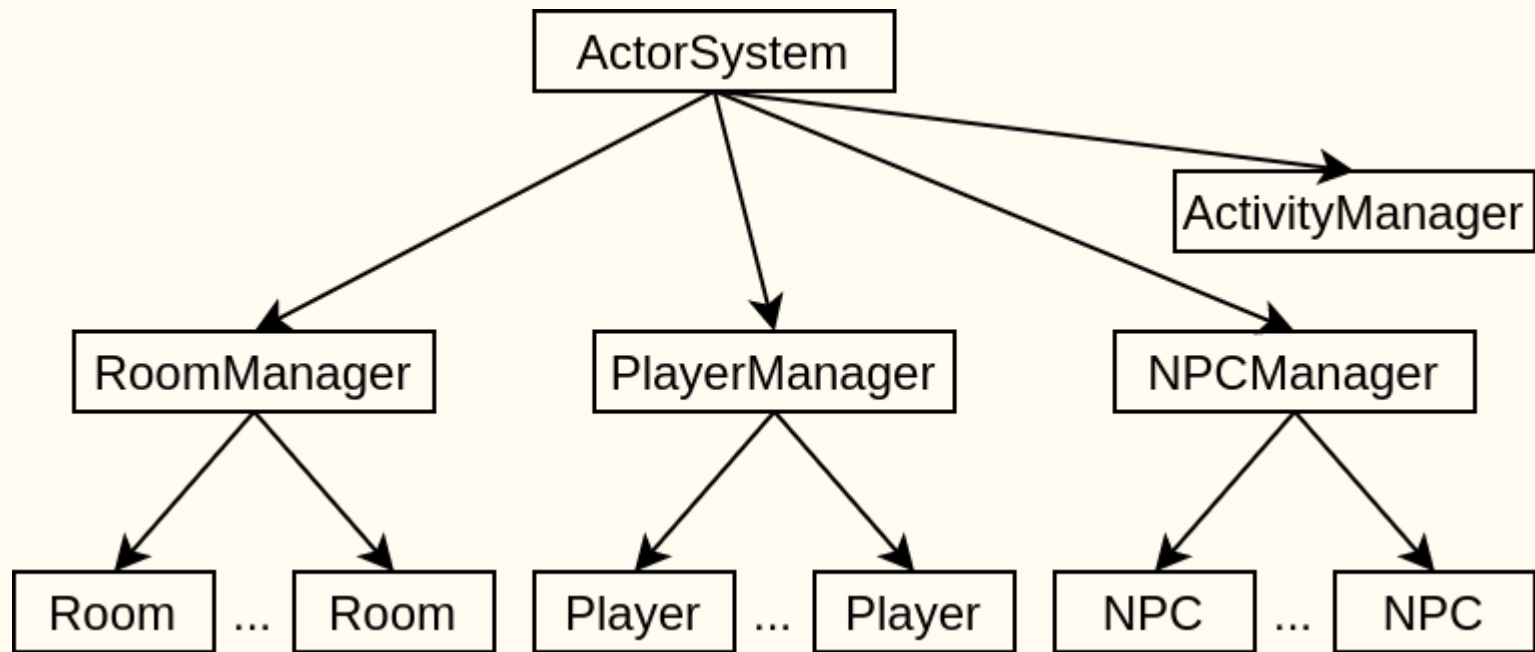
MUD Project

- One of two projects in the class
- Multiplayer text game
 - Students get to pick the theme
- Originally had minimal multithreading (since Spring 2011)
 - Networking forces some because `serverSocket.accept` in a blocking call
- Fall 2016 required actors using Akka
 - Players and Rooms become actors
- Needs to cover many general topics in CS2

Checkpoints

1. Movement, items, help, and exit
2. Store world in a Map and link by keyword
3. Actors
4. Networking
5. Priority Queue and NPCs
6. Combat
7. Shortest path and BST
 - 2 and 3 are refactorings
 - 3 and 6 are the most challenging

Actor Hierarchy



Survey (44 of 99 possible students)

3. How do you feel about the statement, “I enjoyed the MUD project.”? (Strongly Agree, Agree, Disagree, Strongly Disagree)

Strongly Agree	Slightly Agree	Slightly Disagree	Strongly Disagree
68.2%	27.3%	4.5%	0%

4. Comments on how you felt about the MUD project.

More Survey

5. Did you do anything with the MUD project after the course was over?
(Y/N)

Yes	No
20.5%	79.5%

6. If you answered yes to the last question, can you please describe what you did?

More Survey

7. How do you feel about the statement, “The MUD project helped me understand the actor model.”? (Strongly Agree, Agree, Disagree, Strongly Disagree)

Strongly Agree	Slightly Agree	Slightly Disagree	Strongly Disagree
77.3%	22.7%	0%	0%

More Survey

8. How do you feel about the statement, “The actor model helped me understand Object Orientation.”? (Strongly Agree, Agree, Disagree, Strongly Disagree)

Strongly Agree	Slightly Agree	Slightly Disagree	Strongly Disagree
52.3%	36.4%	9.1%	2.3%

More Survey

9. Have you done any parallel work since taking CS2 (later courses, side-projects, internships, professional, etc.)?(Y/N)

Yes	No
29.5%	70.5%

Comments

- 32 answers to question #4
- Challenging/Satisfying (8)
 - The MUD project challenged me in ways I didn't think I could be challenged. Despite the frustration of solving these difficult problems, the project is very rewarding. After all of it I really feel like I did something cool
 - It was difficult especially once we got into the later projects, but building it up from nothing made it worth the effort. Also the creative aspect was enough to have fun, but not too much to make it stressful for “non-creative” types.
 - I thought it was challenging but it was also fun. It is a great feeling to interact with something you made and to improve it. I also felt like actors made a lot of sense to me
 - I thought the project was a really fun and interesting way to learn how to organize code. Once I figured out the “messaging” scheme of the actor model it was actually relatively straightforward to add all the required functionality. It was very satisfying in the end to have a playable game.

Comments

- Creative/Personalized (6)
 - It was a lot of fun! I enjoyed personalizing the game and learned a lot over the course of the project
 - I liked how much creativity was allowed so that we could make the MUD our own.
- Helped with Actors/Parallelism (5)
 - I wrote the code for pretty much the whole project and didn't even really need too much outside help. I felt more responsible for the end product and the actor system implemented felt like the most natural implementation of parallelism
 - I felt the project gave me a good understanding of actors as well as the pitfalls of running operations simultaneously.

Comments

- Favorite Project (4)
 - Probably my single favorite project so far in the degree.
 - I would say it's one of my favorite projects in our CS program. I think I still remember many details of it. Very instructive.
- Topic Coverage (3)
 - I enjoyed the project overall as it utilizes most of the concepts covered in CS2 in an actual program.
 - It was a lot, but I loved how everything we were learning tied into the game somehow.

Comments

- Continued Working On It (3)
 - I loved the project. I have continued to build on it in my own time and couldn't speak more highly about how much I learned from it.
 - I enjoyed it and thought it was a great project for class. I've also included it on my resume and discussed it with interviewers. Specifically the head of engineering at Sledgehammer Games, where I'll work this summer, was very interested in the project and had a lot of questions about it.
 - It was one of my favorite projects I did while at Trinity. I continued to work on it even after I graduated.

Instructor Perspective

- Strongly enforces multithreading on project
- Opportunities to revisit multithreading
- Grading is time consuming

Conclusions and Future Work

- We feel that this project generally works well and accomplishes its goals
- Would like to demonstrate multithreading and performance advantages
- Consider moving to Akka Typed

Questions?