

Throwing Exceptions for Concurrency Errors

Author: Michaela Tremblay

Advisor: Phil Hatcher and Michel Charpentier

Department: Computer Science

Background

- ❖ As multiprocessor systems become widely used, multithreaded programs become more important.
- ❖ Threads work best with multiple CPUs so that a task can be handled in parallel.
- ❖ **Threads are hard.** They are hard to understand and hard to debug.
- ❖ Since threads are so widely used today, many solutions are being researched to make using them easier.

Multithreaded programming



Problems

Bugs	How it occurs
❖ Deadlock	❖ When a process gets into a state in which all threads are no longer running, such as if all threads are waiting for another thread to do something
❖ Automicity Violations	❖ When the desired serializability among multiple memory accesses is violated
❖ Order Violations	❖ When the desired order between two groups of memory accesses is flipped.

What others are doing

Solutions

- ❖ Discourage memory sharing (Example: Google Go)
- ❖ Deterministic scheduling
- ❖ Warn users of race conditions (Example: Helgrind)

Our Solution

- ❖ The programmer's intent will be checked in order to identify any data races. This will be done by checking whether or not an object was marked for sharing and whether or not the thread trying to access the object has permission to do so. If either of these checks fail, then a run-time intention exception is thrown.

Goals

1. Determine if this proposed solution is feasible.
2. Determine if **concurrency bugs would be caught more easily** using this proposed solution.

Approach

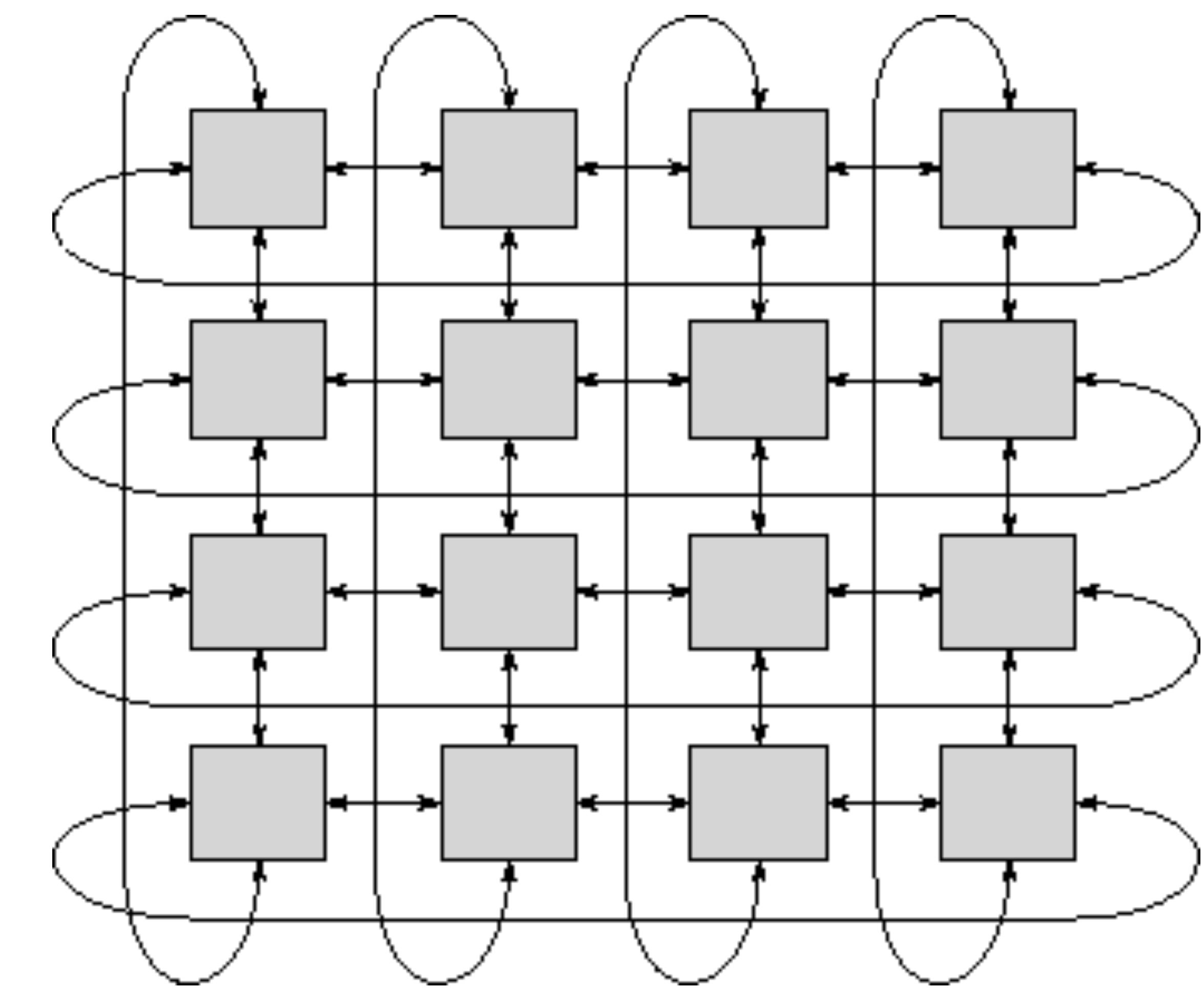
- ❖ Develop parts of the concurrency library.
 - ❖ Based off of java.util.concurrent
- ❖ Write two applications using TScript.
 - ❖ Are errors easier to catch?

Concurrency Library

- ❖ cyclic barrier
- ❖ countdown latch
- ❖ executor
- ❖ atomic number
- ❖ future task

Applications

- ❖ 20 longest words in N files
- ❖ Network on a Chip



Observations

- ❖ Spent a lot of time getting the code to run without any Intention Errors.
- ❖ Once all the Intention Errors were gone the program ran without any bugs.
- ❖ I did not have to fight with the threads to figure out where bugs were occurring because the Intention Errors told me exactly where they were.

Conclusion

- ❖ Creating the TScript language as proposed is possible
- ❖ Tscript does help catch concurrency bugs.
- ❖ The Intention Errors catch multiple bugs that may not have been caught otherwise.
- ❖ The programmer needs to focus on solving the Intention Errors instead of searching for where a bug is located.

Next Steps

Although TScript works well for eliminating race conditions, there are other concurrency bugs that are not yet caught by TScript. TScript should be extended to catch bugs such as deadlock and order violations.