Summary of Java (Common) Fork-Join Pool Benefits

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

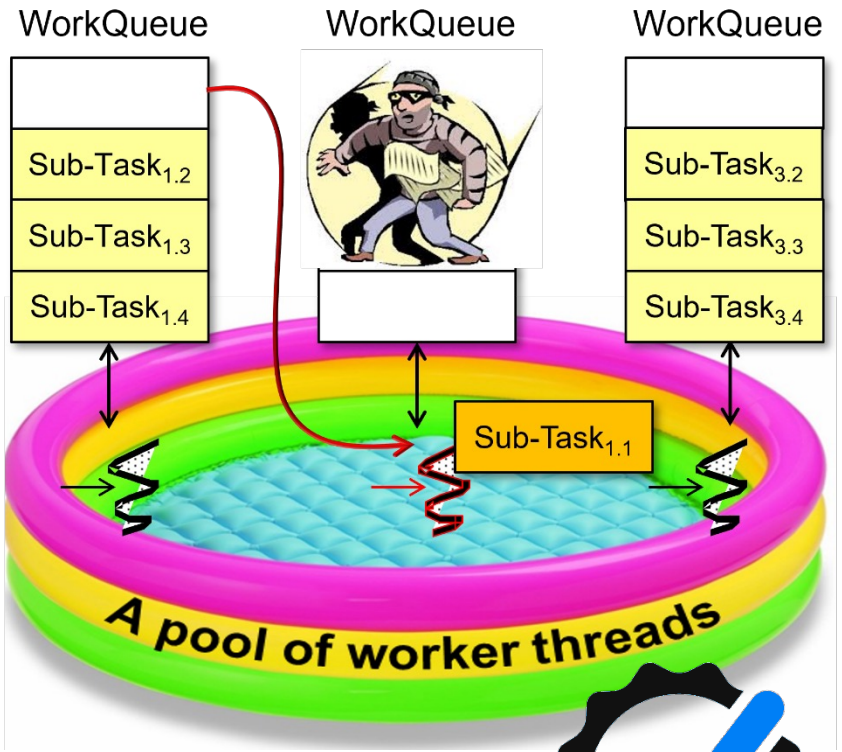
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

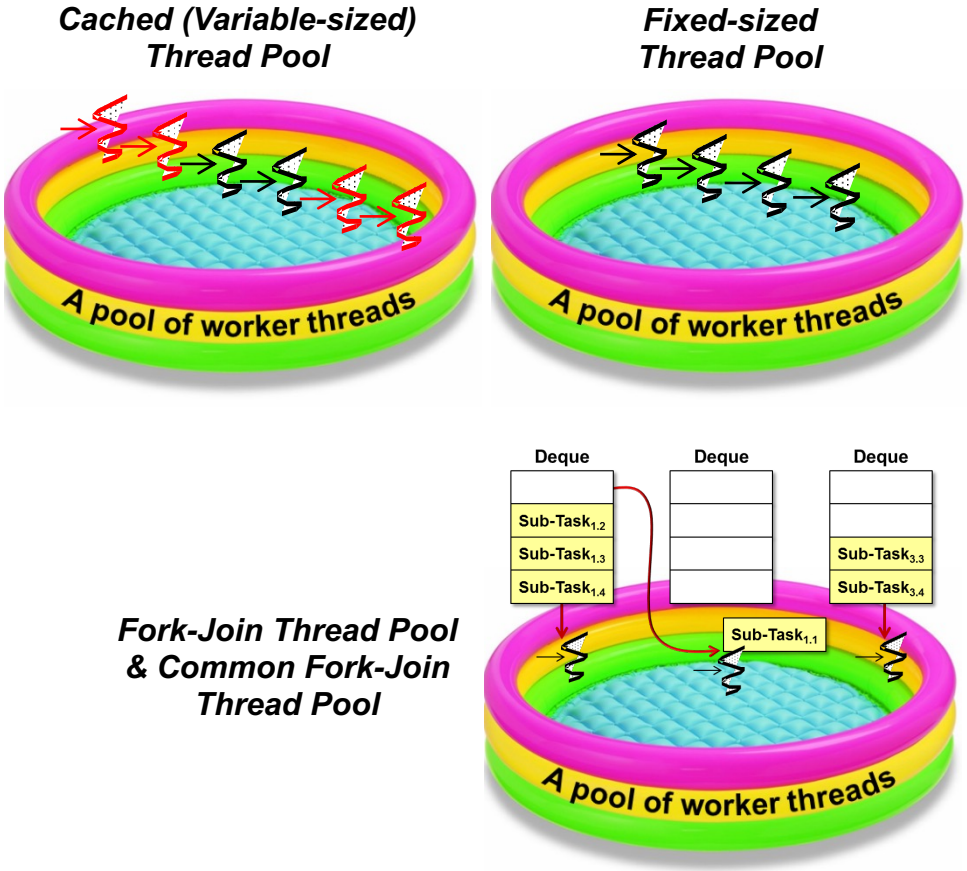
- Understand how the common fork-join pool helps to maximize processor core utilization
- Recognize how the ManagedBlocker interface helps avoid starvation & improve performance
- Be able to apply the ManagedBlocker interface on blocking synchronizers & queues
- Know how to encapsulate ManageBlocker & apply it on blocking I/O operations
- Be aware of the benefits of the Java (common) fork-join pool



Benefits of the Java Fork-Join Pool

Benefits of the Java Common Fork-Join Pool

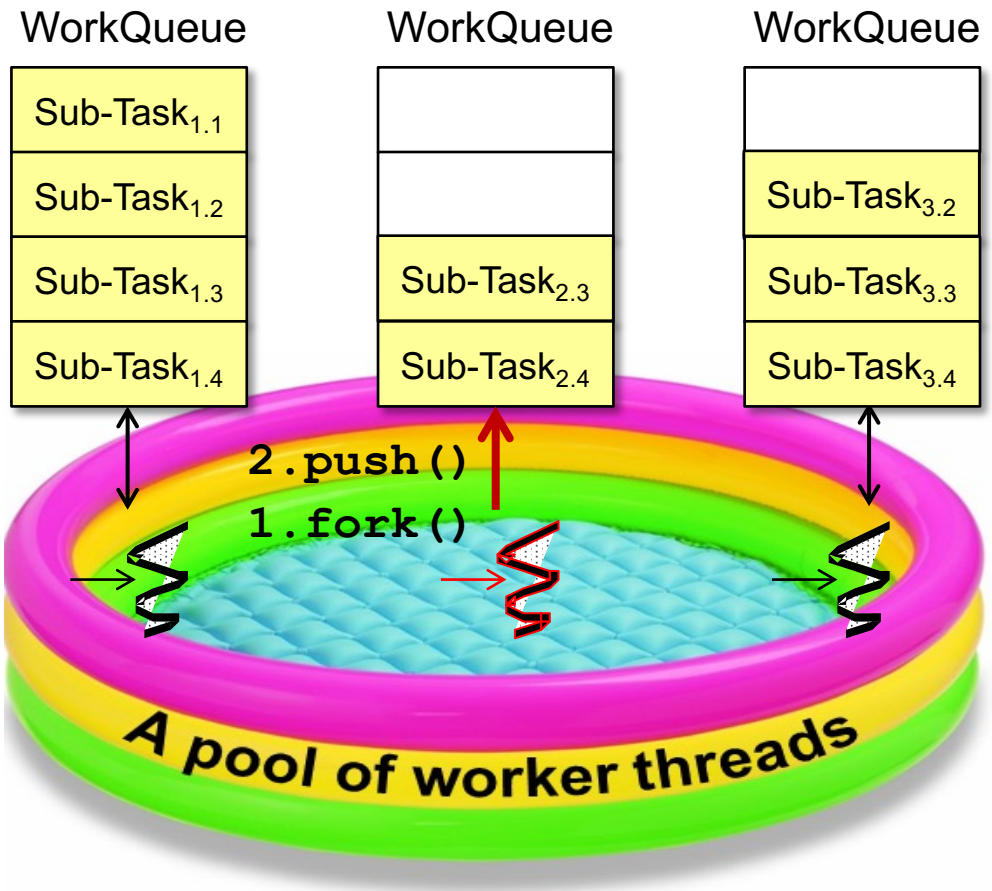
- There are several benefits of the Java fork-join pool vs. other Java thread pools



See docs.oracle.com/javase/tutorial/essential/concurrency/pools.html

Benefits of the Java Common Fork-Join Pool

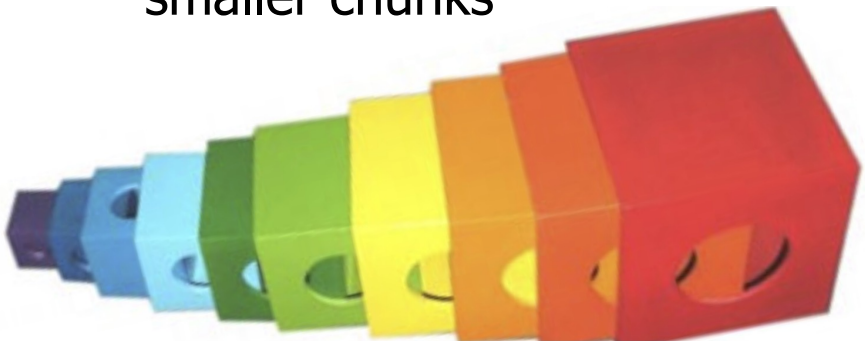
- There are several benefits of the Java fork-join pool vs. other Java thread pools
  - Locality of reference
    - Improves cache performance

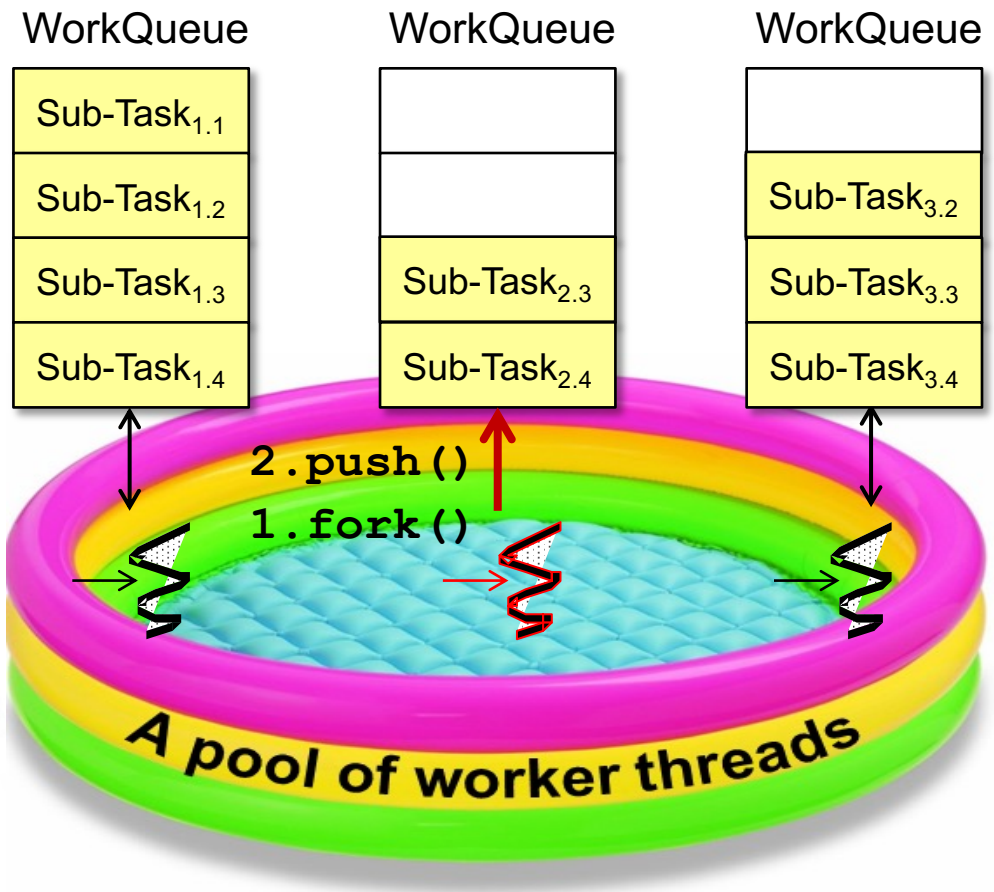


See en.wikipedia.org/wiki/Locality_of_reference

Benefits of the Java Common Fork-Join Pool

- There are several benefits of the Java fork-join pool vs. other Java thread pools
  - Locality of reference
  - Recursive decomposition
    - Larger chunks are pushed onto the deque before smaller chunks

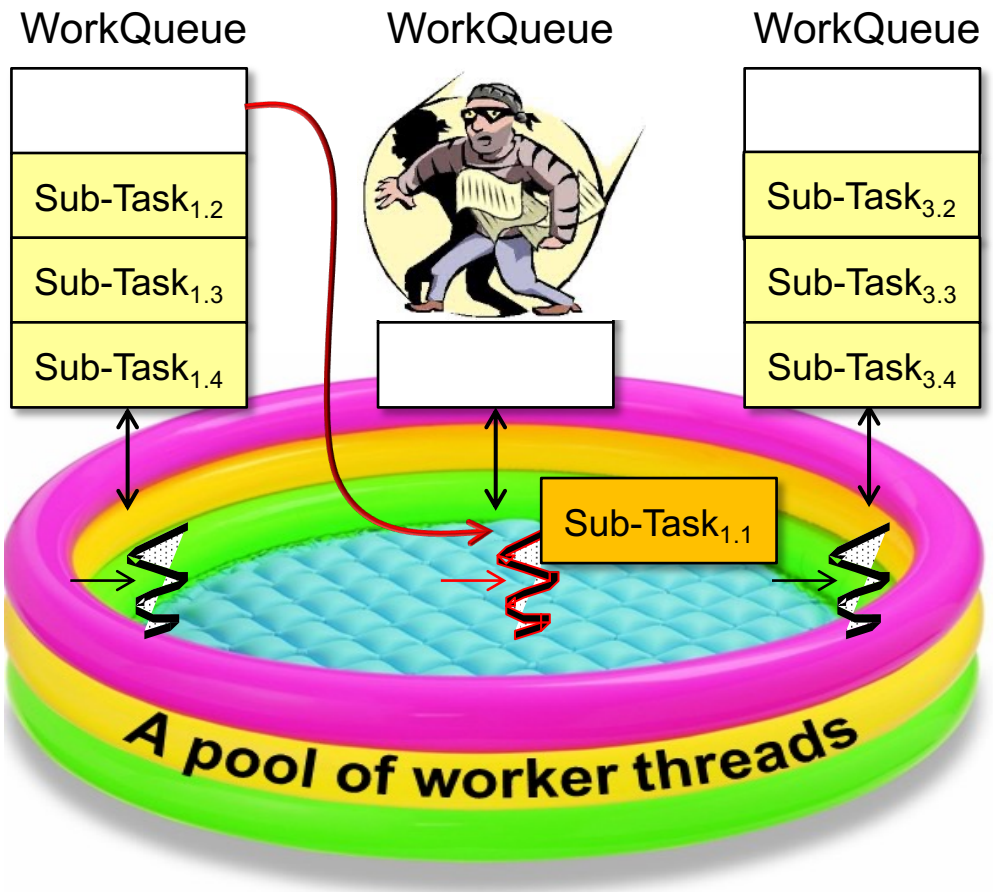




See developer.ibm.com/articles/j-java-streams-5-brian-goetz

Benefits of the Java Common Fork-Join Pool

- There are several benefits of the Java fork-join pool vs. other Java thread pools
  - Locality of reference
  - Recursive decomposition
  - Work-stealing
    - To maximize core utilization, idle worker threads "steal" work from the tail of busy threads' deques



See gee.cs.oswego.edu/dl/papers/fj.pdf

Benefits of the Java Common Fork-Join Pool

- There are several benefits of the Java fork-join pool vs. other Java thread pools
  - Locality of reference
  - Recursive decomposition
  - Work-stealing
  - Auto-scaling via the Managed Blocker interface
    - Temporarily add worker threads to a Java fork-join pool

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/ForkJoinPool.ManagedBlocker.html

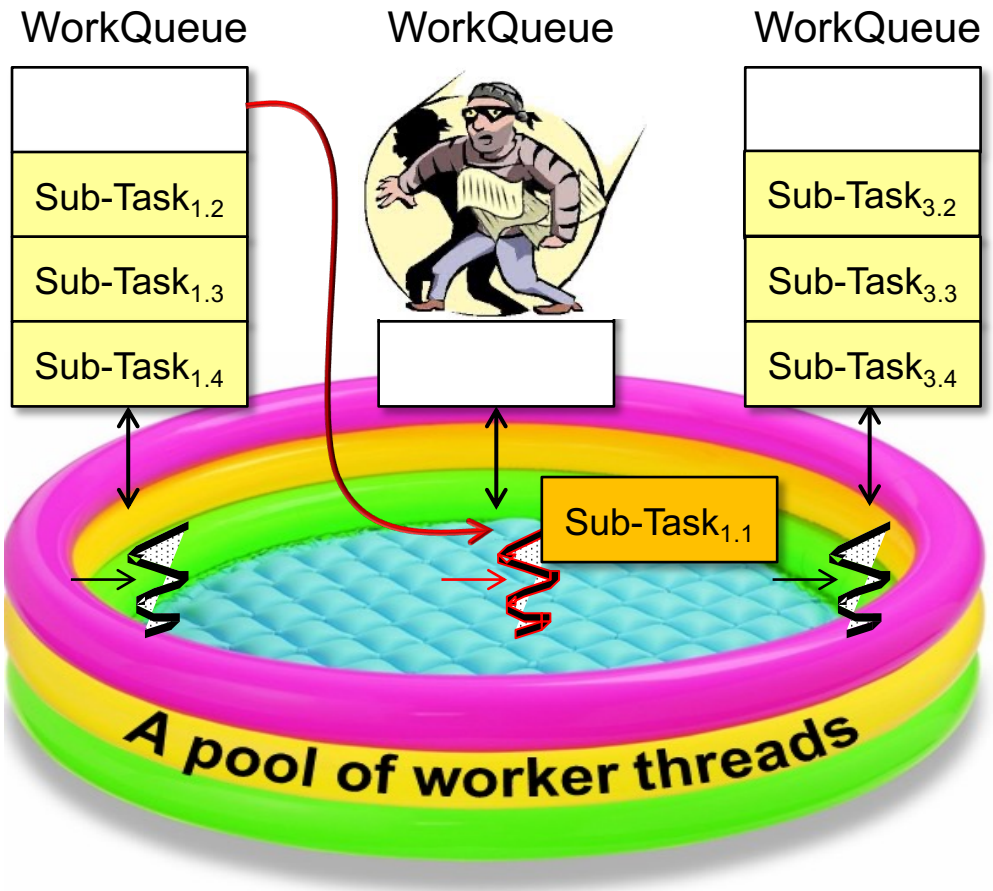
Benefits of the Java Common Fork-Join Pool

Benefits of the Java Common Fork-Join Pool

- There are also several benefits of the Java common fork-join pool vs. other Java thread pools
  - Optimized resource utilization
    - It's aware of which cores are used globally within a process

Benefits of the Java Common Fork-Join Pool

- There are also several benefits of the Java common fork-join pool vs. other Java thread pools
  - Optimized resource utilization
    - It's aware of which cores are used globally within a process
      - Enables efficient work-stealing



Benefits of the Java Common Fork-Join Pool

- There are also several benefits of the Java common fork-join pool vs. other Java thread pools
  - Optimized resource utilization
  - A common ForkJoinPool can be used without explicitly creating a new instance

See en.wikipedia.org/wiki/Out_of_the_box_(feature)

Benefits of the Java Common Fork-Join Pool

- There are also several benefits of the Java common fork-join pool vs. other Java thread pools
  - Optimized resource utilization
  - A common ForkJoinPool can be used without explicitly creating a new instance
    - Reduced need for manual configuration, initialization, & cleanup

End of Summary of Java (Common) Fork-Join Pool Benefits