

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

**Douglas C. Schmidt**

**[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)**

**[www.dre.vanderbilt.edu/~schmidt](http://www.dre.vanderbilt.edu/~schmidt)**

**Professor of Computer Science**

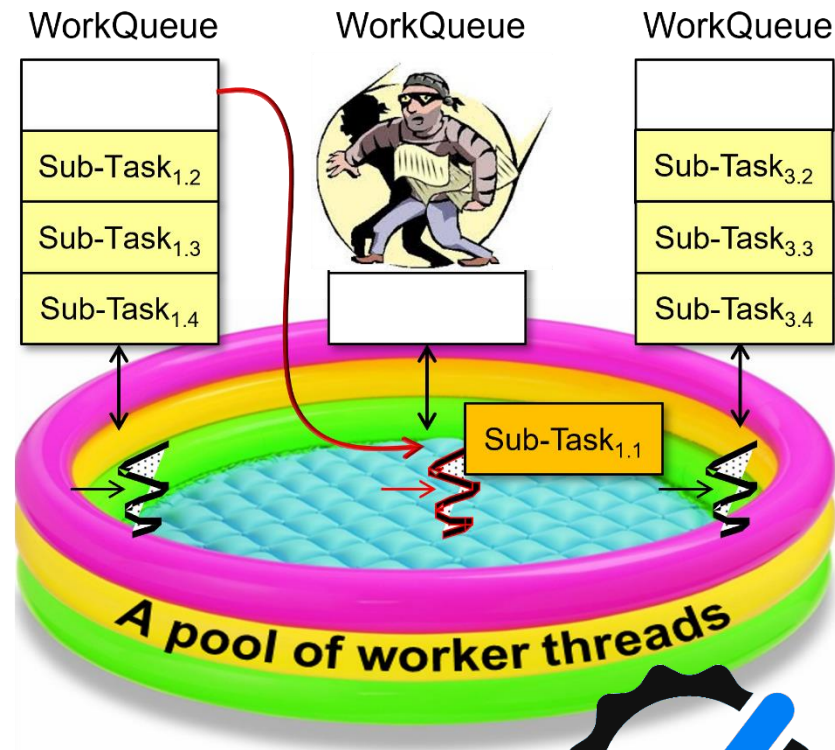
**Institute for Software  
Integrated Systems**

**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 ManagedBlocker & 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

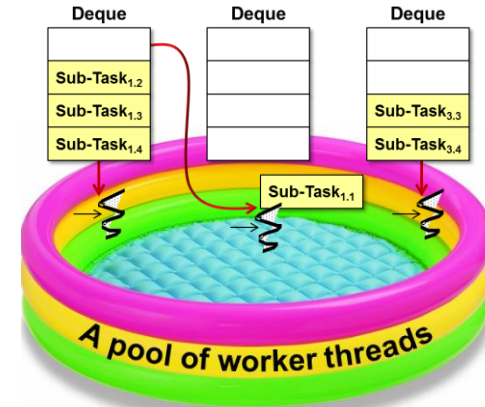
*Cached (Variable-sized)  
Thread Pool*



*Fixed-sized  
Thread Pool*

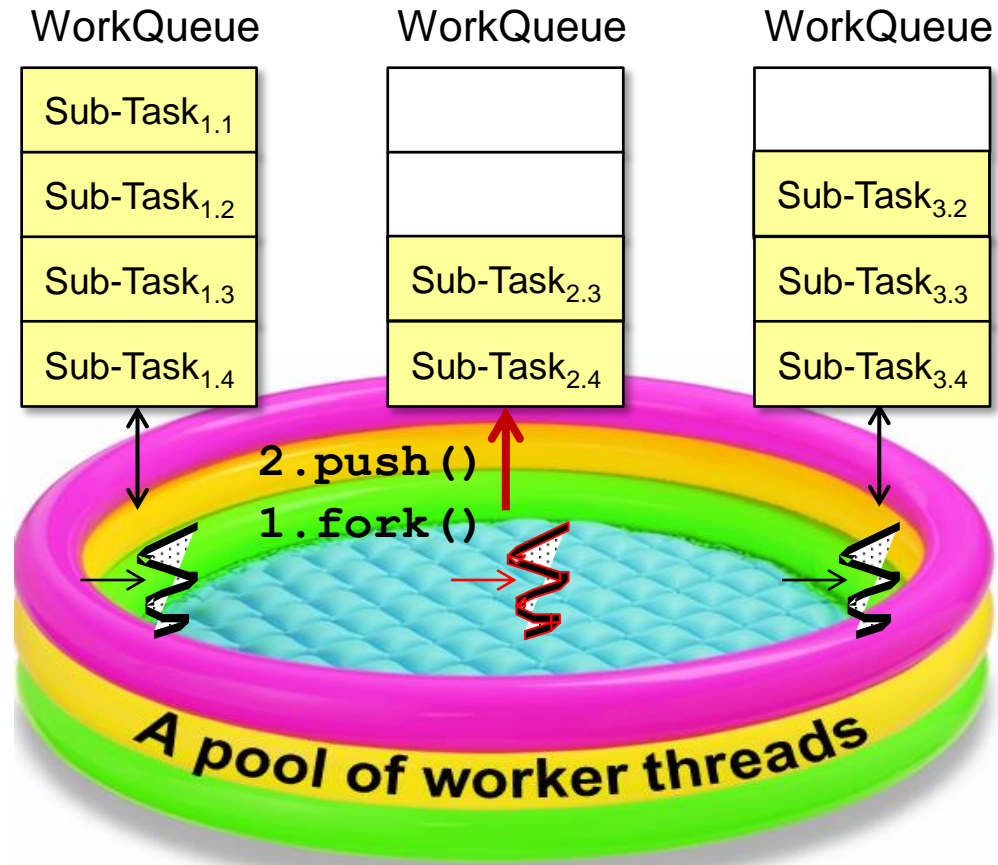


*Fork-Join Thread Pool  
& Common Fork-Join  
Thread Pool*



# Benefits of the Java Common Fork-Join Pool

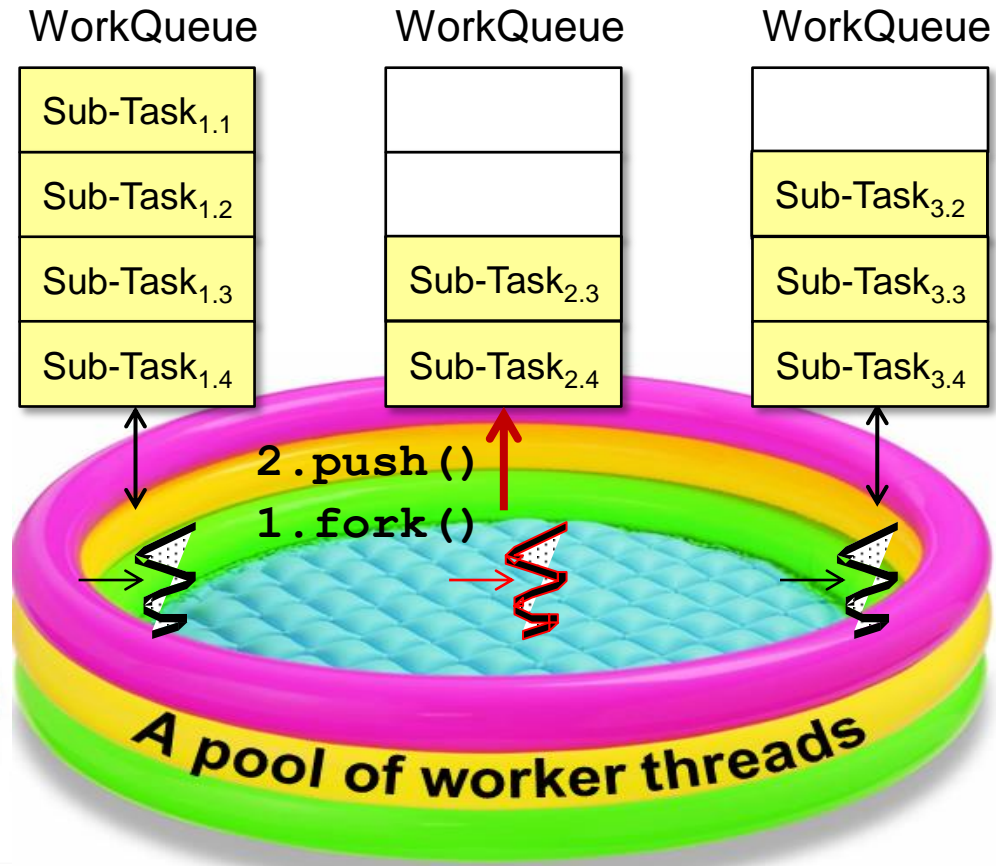
- There are several benefits of the Java fork-join pool vs. other Java thread pools
  - Locality of reference
    - Improves cache performance



See [en.wikipedia.org/wiki/Locality\\_of\\_reference](https://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

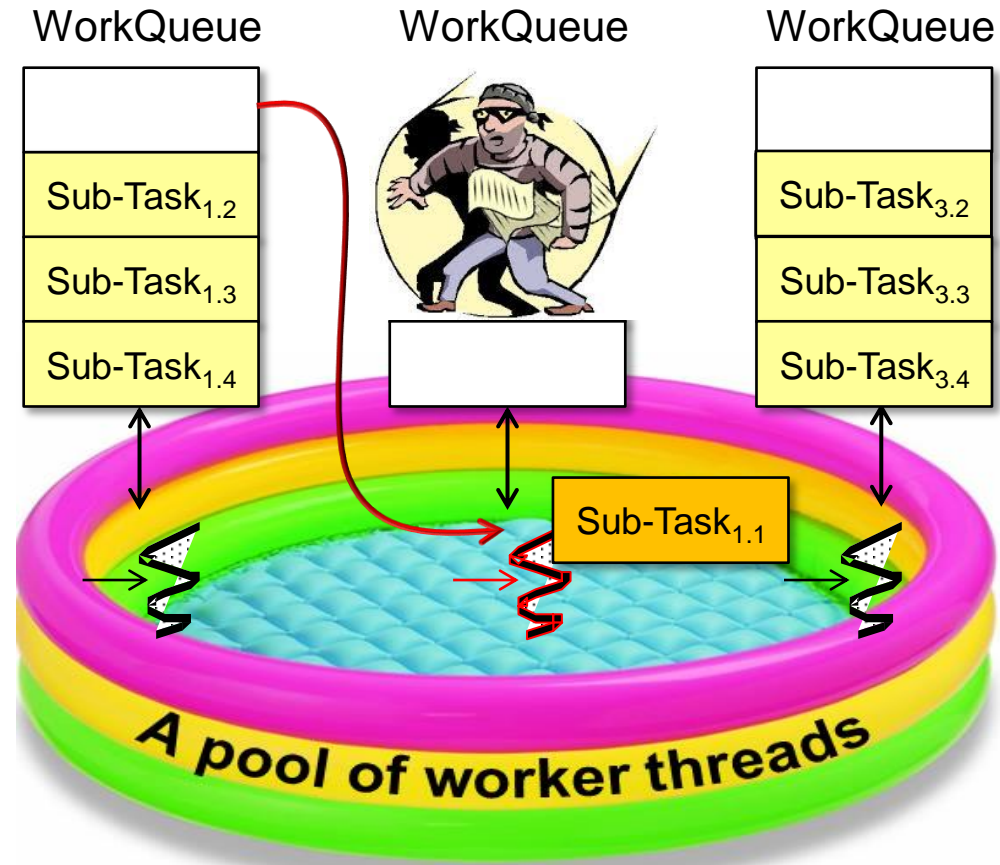


See [developer.ibm.com/articles/j-java-streams-5-brian-goetz](https://developer.ibm.com/articles/j-java-streams-5-brian-goetz)



# 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’ dequeues



---

# 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
  - Auto-scaling via the Managed Blocker interface
  - Temporarily add worker threads to the common fork-join pool



---

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