

Key Methods in the Java ForkJoinPool

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

- Recognize some of the key methods in the ForkJoinPool class

ForkJoinPool		
m	ForkJoinPool()	
m	ForkJoinPool(int)	
m	commonPool()	ForkJoinPool
m	execute(Runnable)	void
m	execute(ForkJoinTask<?>)	void
m	invoke(ForkJoinTask<T>)	T
m	invokeAll(Collection<Callable<T>>)	List<Future<T>>
m	invokeAll(Collection<Callable<T>>, long, TimeUnit)	List<Future<T>>
m	invokeAny(Collection<Callable<T>>, long, TimeUnit)	T
m	invokeAny(Collection<Callable<T>>)	T
m	submit(Runnable, T)	ForkJoinTask<T>
m	submit(ForkJoinTask<T>)	ForkJoinTask<T>

There are *many* other methods in this class, a few of which we discuss here!

Key Methods in Java ForkJoinPool

Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService

```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(Runnable cmd) {...}

    <T> Future<T> submit
        (Callable<T> task) {...}

    <T> List<Future<T>> invokeAll
        (Collection<? extends
         Callable<T>> tasks) {...}

    <T> T invokeAny
        (Collection<? extends
         Callable<T>> tasks) {...}
```

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

Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService
- It therefore implements the ExecutorService methods

```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(Runnable cmd){...}

    <T> Future<T> submit
        (Callable<T> task){...}

    <T> List<Future<T>> invokeAll
        (Collection<? extends
            Callable<T>> tasks){...}

    <T> T invokeAny
        (Collection<? extends
            Callable<T>> tasks){...}
```

Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService
- It therefore implements the ExecutorService methods
 - Arrange async execution of a one-way task



```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(Runnable cmd) {...}

    <T> Future<T> submit
        (Callable<T> task){...}

    <T> List<Future<T>> invokeAll
        (Collection<? extends
            Callable<T>> tasks){...}

    <T> T invokeAny
        (Collection<? extends
            Callable<T>> tasks){...}
```

Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService
- It therefore implements the ExecutorService methods
 - Arrange async execution of a one-way task
 - Submit a two-way task for execution, return a future



```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(Runnable cmd){...}

    <T> Future<T> submit
        (Callable<T> task){...}

    <T> List<Future<T>> invokeAll
        (Collection<? extends
            Callable<T>> tasks){...}

    <T> T invokeAny
        (Collection<? extends
            Callable<T>> tasks){...}
```

Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService
- It therefore implements the ExecutorService methods
 - Arrange async execution of a one-way task
 - Submit a two-way task for execution, return a future
 - Run all tasks in the collection & wait for them all to finish

```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(Runnable cmd){...}

    <T> Future<T> submit
        (Callable<T> task){...}

    <T> List<Future<T>> invokeAll
        (Collection<? extends
            Callable<T>> tasks){...}

    <T> T invokeAny
        (Collection<? extends
            Callable<T>> tasks){...}
```


Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService
- It therefore implements the ExecutorService methods
 - Arrange async execution of a one-way task
 - Submit a two-way task for execution, return a future
 - Run all tasks in the collection & wait for them all to finish
 - Run all tasks in the collection & wait for the first to finish

```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(Runnable cmd){...}

    <T> Future<T> submit
        (Callable<T> task){...}

    <T> List<Future<T>> invokeAll
        (Collection<? extends
            Callable<T>> tasks){...}

    <T> T invokeAny
        (Collection<? extends
            Callable<T>> tasks){...}
```

Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService
- It therefore implements the ExecutorService methods
 - Arrange async execution of a one-way task
 - Submit a two-way task for execution, return a future
 - Run all tasks in the collection & wait for them all to finish
 - Run all tasks in the collection & wait for the first to finish

```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(Runnable cmd){...}

    <T> Future<T> submit
        (Callable<T> task){...}

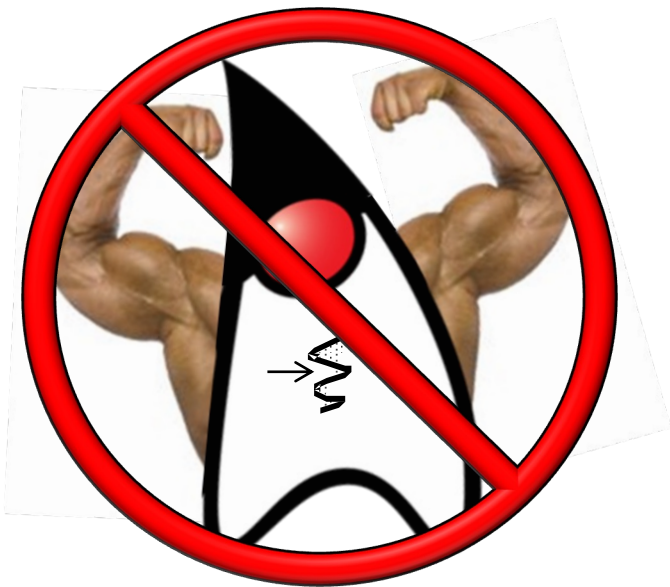
    <T> List<Future<T>> invokeAll
        (Collection<? extends
            Callable<T>> tasks){...}

    <T> T invokeAny
        (Collection<? extends
            Callable<T>> tasks){...}
```

There are also versions of these methods that are time-bounded

Key Methods in Java ForkJoinPool

- ForkJoinPool extends Abstract ExecutorService
- It therefore implements the ExecutorService methods



```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(Runnable cmd){...}

    <T> Future<T> submit
        (Callable<T> task){...}

    <T> List<Future<T>> invokeAll
        (Collection<? extends
            Callable<T>> tasks){...}

    <T> T invokeAny
        (Collection<? extends
            Callable<T>> tasks){...}
```

However, these methods don't directly leverage powerful fork-join features!

Key Methods in Java ForkJoinPool

- ForkJoinPool extends Abstract ExecutorService
 - It therefore implements the ExecutorService methods
 - It also implements key methods for non-ForkJoinTask clients

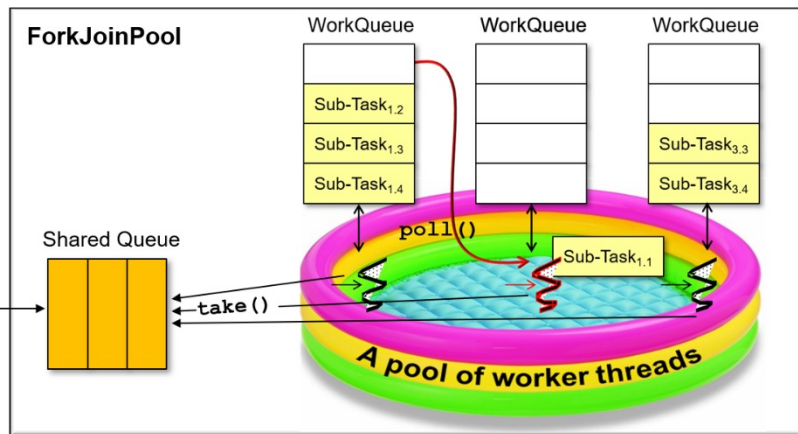
```
class ForkJoinPool extends  
    AbstractExecutorService {  
    ...  
    void execute (ForkJoinTask<T>  
        task)
```

```
{ ... }
```

```
T invoke (ForkJoinTask<T> task)  
{ ... }
```

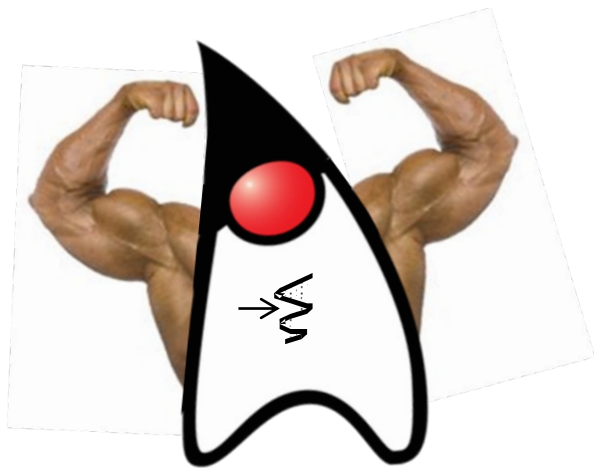
```
ForkJoinTask<T> submit  
    (ForkJoinTask<T> task)
```

```
{ ... }
```



Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService
 - It therefore implements the ExecutorService methods
 - It also implements key methods for non-ForkJoinTask clients



```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(ForkJoinTask<T>
        task)
    { ... }

    T invoke(ForkJoinTask<T> task)
    { ... }

    ForkJoinTask<T> submit
        (ForkJoinTask<T> task)
    { ... }
```

These methods *can* directly leverage powerful fork-join features

Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService
 - It therefore implements the ExecutorService methods
 - It also implements key methods for non-ForkJoinTask clients
 - Arrange async execution of one-way task



```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(ForkJoinTask<T>
                task)
    { ... }

    T invoke(ForkJoinTask<T> task)
    { ... }

    ForkJoinTask<T> submit
        (ForkJoinTask<T> task)
    { ... }
```

Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService
 - It therefore implements the ExecutorService methods
 - It also implements key methods for non-ForkJoinTask clients
 - Arrange async execution of one-way task
 - Perform the task, blocking until it completes



```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(ForkJoinTask<T>
                task)
    { ... }

    T invoke(ForkJoinTask<T> task)
    { ... }

    ForkJoinTask<T> submit
        (ForkJoinTask<T> task)
    { ... }
```

Key Methods in Java ForkJoinPool

- ForkJoinPool extends AbstractExecutorService
 - It therefore implements the ExecutorService methods
 - It also implements key methods for non-ForkJoinTask clients
 - Arrange async execution of one-way task
 - Perform the task, blocking until it completes
 - Submit a ForkJoinTask for async execution, return a future

```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    void execute(ForkJoinTask<T>
                task)
    { ... }

    T invoke(ForkJoinTask<T> task)
    { ... }

    ForkJoinTask<T> submit
        (ForkJoinTask<T> task)
    { ... }
```



Key Methods in Java ForkJoinPool

- The ForkJoinPool size defaults to # of cores available to Java runtime

```
class ForkJoinPool extends
    AbstractExecutorService {
    public ForkJoinPool() {
        this(Math.min(MAX_CAP,
            Runtime.getRuntime()
                .availableProcessors()),
            ...);
    }

    public ForkJoinPool
        (int parallelism) {
        this(parallelism, ...);
    }
    ...
}
```

Key Methods in Java ForkJoinPool

- The ForkJoinPool size defaults to # of cores available to Java runtime

Returns # of processor cores available to the Java execution environment



```
class ForkJoinPool extends
    AbstractExecutorService {
    public ForkJoinPool() {
        this(Math.min(MAX_CAP,
            Runtime.getRuntime()
            .availableProcessors()),
            ...);
    }

    public ForkJoinPool
        (int parallelism) {
        this(parallelism, ...);
    }
    ...
}
```

Key Methods in Java ForkJoinPool

- The ForkJoinPool size defaults to # of cores available to Java runtime
- This size can also be controlled programmatically via the ForkJoinPool constructor

```
class ForkJoinPool extends
    AbstractExecutorService {
    public ForkJoinPool() {
        this(Math.min(MAX_CAP,
            Runtime.getRuntime()
                .availableProcessors()),
            ...);
    }

    public ForkJoinPool
        (int parallelism) {
        this(parallelism, ...);
    }
    ...
}
```

Key Methods in Java ForkJoinPool

- The common fork-join pool can be accessed via a static method

```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    static final ForkJoinPool
        common;

    public static ForkJoinPool
        commonPool () {
        return common;
    }
}
```

Key Methods in Java ForkJoinPool

- The common fork-join pool can be accessed via a static method

```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    static final ForkJoinPool
        common;

    public static ForkJoinPool
        commonPool() {
        return common;
    }
}
```

This method accesses a static field that can be accessed via all threads in a process

Key Methods in Java ForkJoinPool

- The common fork-join pool can be accessed via a static method
- The common pool is used by any ForkJoinTask that is not explicitly submitted to a specified pool



```
class ForkJoinPool extends
    AbstractExecutorService {
    ...
    static final ForkJoinPool
        common;

    public static ForkJoinPool
        commonPool() {
        return common;
    }
}
```

Other Methods in Java ForkJoinPool

Other Methods in Java ForkJoinPool

- ForkJoinPool also provides various management & monitoring operations

int	getParallelism() – Returns the targeted parallelism level of this pool
int	getPoolSize() – Returns the number of worker threads that have started but not yet terminated
int	getQueuedSubmissionCount() – Returns an estimate of the number of tasks submitted to this pool that have not yet begun executing
long	getStealCount() – Returns an estimate of the total number of tasks stolen from one thread's work queue by another

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

Other Methods in Java ForkJoinPool

- ForkJoinPool also provides various management & monitoring operations

int	<u>getParallelism()</u> – Returns the targeted parallelism level of this pool
int	<u>getPoolSize()</u> – Returns the number of worker threads that have started but not yet terminated
int	<u>getQueuedSubmissionCount()</u> – Returns an estimate of the number of tasks submitted to this pool that have not yet begun executing
long	<u>getStealCount()</u> – Returns an estimate of the total number of tasks stolen from one thread's work queue by another

Other Methods in Java ForkJoinPool

- ForkJoinPool also provides various management & monitoring operations

int	<u>getParallelism()</u> – Returns the targeted parallelism level of this pool
int	<u>getPoolSize()</u> – Returns the number of worker threads that have started but not yet terminated
int	<u>getQueuedSubmissionCount()</u> – Returns an estimate of the number of tasks submitted to this pool that have not yet begun executing
long	<u>getStealCount()</u> – Returns an estimate of the total number of tasks stolen from one thread's work queue by another

Other Methods in Java ForkJoinPool

- ForkJoinPool also provides various management & monitoring operations

int	<u>getParallelism()</u> – Returns the targeted parallelism level of this pool
int	<u>getPoolSize()</u> – Returns the number of worker threads that have started but not yet terminated
int	<u>getQueuedSubmissionCount()</u> – Returns an estimate of the number of tasks submitted to this pool that have not yet begun executing
long	<u>getStealCount()</u> – Returns an estimate of the total number of tasks stolen from one thread's work queue by another

Other Methods in Java ForkJoinPool

- ForkJoinPool also provides various management & monitoring operations

int	getParallelism() – Returns the targeted parallelism level of this pool
int	getPoolSize() – Returns the number of worker threads that have started but not yet terminated
int	getQueuedSubmissionCount() – Returns an estimate of the number of tasks submitted to this pool that have not yet begun executing
long	getStealCount() – Returns an estimate of the total number of tasks stolen from one thread's work queue by another

End of Key Methods in Java ForkJoinPool