The Java Fork-Join Pool:

Key Methods in ForkJoinPool

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

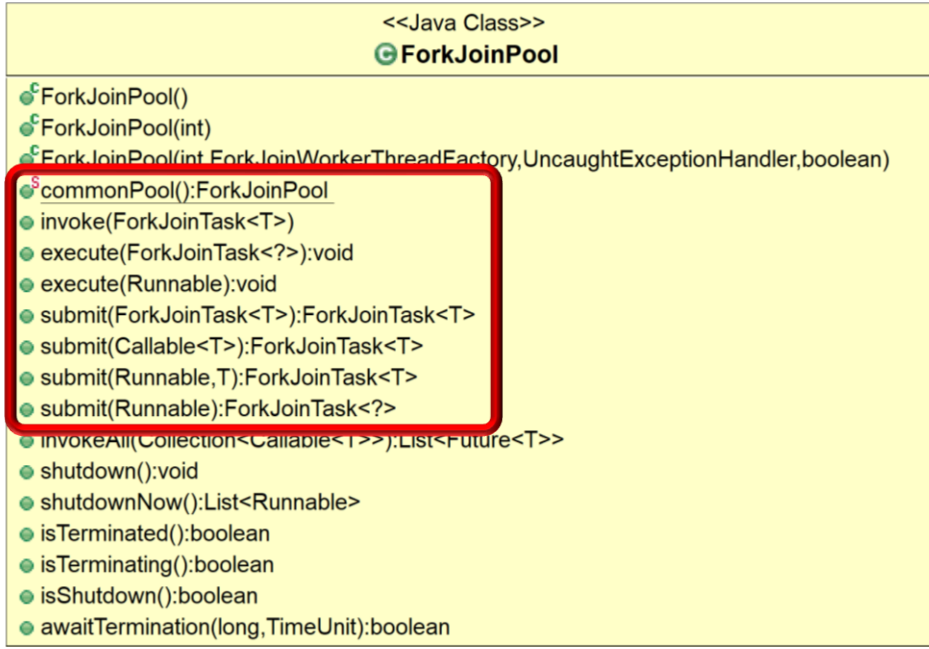
Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Recognize the key methods in the ForkJoinPool class



<<Java Class>>

ForkJoinPool

ForkJoinPool()

ForkJoinPool(int)

ForkJoinPool(int,ForkJoinWorkerThreadFactory,UncaughtExceptionHandler,boolean)

commonPool():ForkJoinPool

invoke(ForkJoinTask<T>)

execute(ForkJoinTask<?>):void

execute(Runnable):void

submit(ForkJoinTask<T>):ForkJoinTask<T>

submit(Callable<T>):ForkJoinTask<T>

submit(Runnable,T):ForkJoinTask<T>

submit(Runnable):ForkJoinTask<?>

invokeAll(Collection<Callable<T>>):List<Future<T>>

shutdown():void

shutdownNow():List<Runnable>

isTerminated():boolean

isTerminating():boolean

isShutdown():boolean

awaitTermination(long,TimeUnit):boolean

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){...}
```

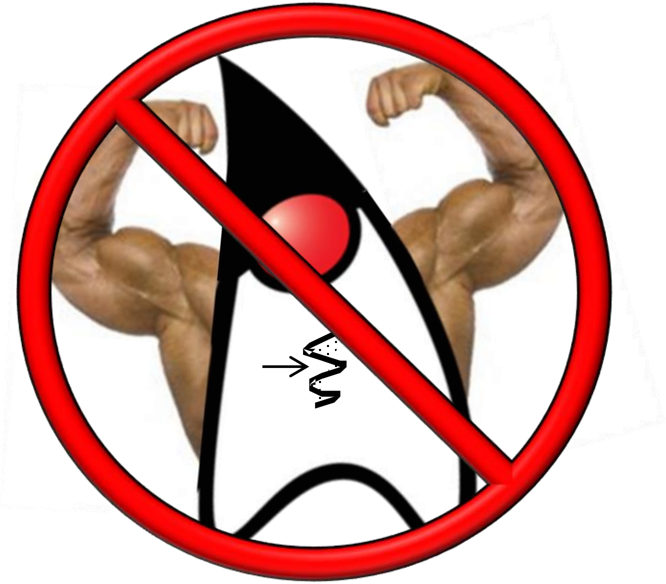
Key Methods in Java ForkJoinPool

- 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

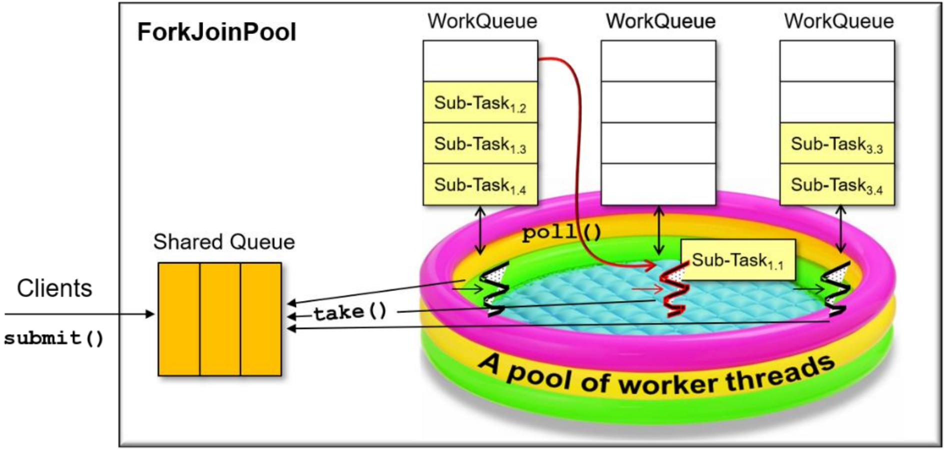


```
class ForkJoinPool extends
      AbstractExecutorService {
  ...
  void execute(Runnable cmd){...}

  <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 leverage the powerful fork-join pool features

Key Methods in Java ForkJoinPool

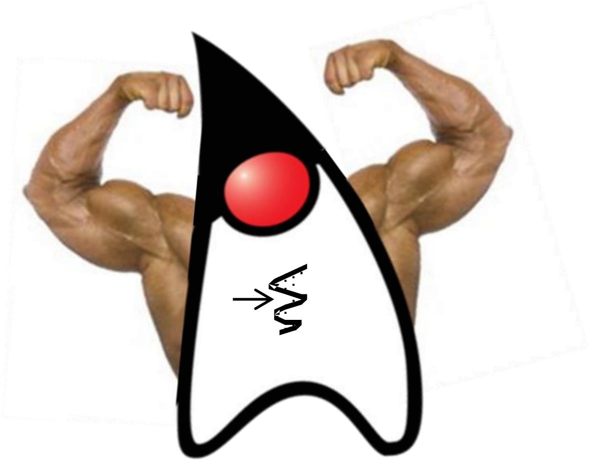
- ForkJoinPool extends AbstractExecutorService
  - It therefore implements the ExecutorService methods
  - It also implements key methods for non-ForkJoinTask clients



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

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

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

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* leverage the powerful properties of the fork-join pool

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)
  { ... }
```

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

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)
  { ... }
```

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

Key Methods in Java ForkJoinPool

- ForkJoinPool extends Abstract ExecutorService
  - 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 execution, return a future

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

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

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

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

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, ...);
  }
  ...
```

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

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, ...);
  }
  ...
```

See docs.oracle.com/javase/8/docs/api/java/lang/Runtime.html#availableProcessors

Key Methods in Java ForkJoinPool

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

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

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

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

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;
  }
```

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

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;
  }
```

Key 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

Key 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

Key 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

Key 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

Key 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 the Java Fork-Join Pool: Key Methods in ForkJoinPool