Java Barrier Synchronizers:
Overview

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

Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Lesson

- Understand how different barrier synchronizers allow threads to wait for operations performed in other threads to complete

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDownLatch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Learning Objectives in this Lesson

• Understand how different barrier synchronizers allow threads to wait for operations performed in other threads to complete

• Recognize a human known use of barrier synchronization
Overview of Barrier Synchronization
Overview of Barrier Synchronization

- A barrier is a synchronization mechanism that halts the progress of one or more threads at a particular point

See [en.wikipedia.org/wiki/Barrier_(computer_science)](http://en.wikipedia.org/wiki/Barrier_(computer_science))
Overview of Barrier Synchronization

- Barriers can be used several ways
Overview of Barrier Synchronization

- Barriers can be used several ways

**A. Entry barrier** - e.g., concurrent computations wait until object is initialized
Overview of Barrier Synchronization

- Barriers can be used several ways

A. **Entry barrier** – e.g., concurrent computations wait until object is initialized

Main thread spawns a number of worker threads & then performs some time-consuming initialization of data structures
Overview of Barrier Synchronization

- Barriers can be used several ways

**A. Entry barrier** – e.g., concurrent computations wait until object is initialized

The worker threads wait on the barrier until the main thread completes its initializations
Overview of Barrier Synchronization

• Barriers can be used several ways

A. **Entry barrier** – e.g., concurrent computations wait until object is initialized

The main thread signals worker threads that they can begin by decrementing the barrier to 0
Overview of Barrier Synchronization

- Barriers can be used several ways

A. **Entry barrier** – e.g., concurrent computations wait until object is initialized

B. **Exit barrier** – e.g., block until all concurrent threads have done their processing before continuing
Overview of Barrier Synchronization

• Barriers can be used several ways

A. **Entry barrier** – e.g., concurrent computations wait until object is initialized

B. **Exit barrier** – e.g., block until all concurrent threads have done their processing before continuing

*The main thread waits for all worker threads to finish*
Overview of Barrier Synchronization

- Barriers can be used several ways

A. **Entry barrier** – e.g., concurrent computations wait until object is initialized

B. **Exit barrier** – e.g., block until all concurrent threads have done their processing before continuing

The main thread waits for all worker threads to finish
Barriers can be used several ways:

A. **Entry barrier** – e.g., concurrent computations wait until object is initialized

B. **Exit barrier** – e.g., block until all concurrent threads have done their processing before continuing

The main thread waits for all worker threads to finish.
Overview of Barrier Synchronization

- Barriers can be used several ways
  
  A. **Entry barrier** – e.g., concurrent computations wait until object is initialized
  
  B. **Exit barrier** – e.g., block until all concurrent threads have done their processing before continuing

The main thread waits for all worker threads to finish
Overview of Barrier Synchronization

- Barriers can be used several ways

  A. **Entry barrier** – e.g., concurrent computations wait until object is initialized

  B. **Exit barrier** – e.g., block until all concurrent threads have done their processing before continuing

*The main thread can now continue*
Overview of Barrier Synchronization

- Barriers can be used several ways

  A. **Entry barrier** – e.g., concurrent computations wait until object is initialized

  B. **Exit barrier** – e.g., block until all concurrent threads have done their processing before continuing

  C. **Cyclic barrier** – e.g., a group of threads all wait for each other to reach a barrier before advancing to the next cycle
Overview of Barrier Synchronization

• Barriers can be used several ways

  A. **Entry barrier** – e.g., concurrent computations wait until object is initialized

  B. **Exit barrier** – e.g., block until all concurrent threads have done their processing before continuing

  C. **Cyclic barrier** – e.g., a group of threads all wait for each other to reach a barrier before advancing to the next cycle

A fixed- or variable-size pool of threads can run concurrently
Overview of Barrier Synchronization

• Barriers can be used several ways

A. **Entry barrier** – e.g., concurrent computations wait until object is initialized

B. **Exit barrier** – e.g., block until all concurrent threads have done their processing before continuing

C. **Cyclic barrier** – e.g., a group of threads all wait for each other to reach a barrier before advancing to the next cycle

At the end of each cycle a decision is made about whether to continue or not
Barriers can be categorized in several ways:

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDown Latch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Overview of Barrier Synchronization

- Barriers can be categorized in several ways

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDown Latch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Overview of Barrier Synchronization

- Barriers can be categorized in several ways

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDown Latch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Overview of Barrier Synchronization

- Barriers can be categorized in several ways

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDown Latch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Overview of Barrier Synchronization

- Barriers can be categorized in several ways

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDown Latch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>
Overview of Barrier Synchronization

- Barriers can be categorized in several ways

<table>
<thead>
<tr>
<th># of Iterations</th>
<th>Fixed # of Parties</th>
<th>Variable # of Parties</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-Shot</td>
<td>CountDown Latch</td>
<td>Phaser</td>
</tr>
<tr>
<td>Cyclic</td>
<td>CyclicBarrier</td>
<td>Phaser</td>
</tr>
</tbody>
</table>

These categories are not mutually exclusive
Human Known Uses of Barrier Synchronization
Human Known Uses of Barrier Synchronization

• A human known use is protocol used by a museum tour guide

See [en.wikipedia.org/wiki/Tour_guide](en.wikipedia.org/wiki/Tour_guide)
Human Known Uses of Barrier Synchronization

• A human known use is protocol used by a museum tour guide

A. **Entry barrier** – Group of tourists wait outside museum until it opens
Human Known Uses of Barrier Synchronization

- A human known use is protocol used by a museum tour guide
  
  **A. Entry barrier** – Group of tourists wait outside museum until it opens
  
  **B. Exit barrier** – Museum closes after last group of tourists leave
Human Known Uses of Barrier Synchronization

• A human known use is protocol used by a museum tour guide
  
  A. Entry barrier – Group of tourists wait outside museum until it opens
  
  B. Exit barrier – Museum closes after last group of tourists leave
  
  C. Cyclic barrier – Tour guide waits for all tourists to finish exploring a room before continuing tour in next room

Barriers can be used for both fixed- & variable-sized number of tourists
End of Java Barrier
Synchronizers: Overview