Java Parallel Streams Internals:
Order of Results (Part 1)

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 parallel stream internals, e.g.
- Know what can change & what can’t
  - Splitting, combining, & pooling mechanisms
  - Order of processing
- Order of results
  - Overview
Java Parallel Stream
Results Order
The processing results in a parallel stream are more deterministic than ordering.

See en.wikipedia.org/wiki/Deterministic_algorithm
Java Parallel Stream Results Order

- The processing *results* in a parallel stream are more deterministic than ordering.
- Programmers can control if results are presented in "encounter order" (EO).

**EO is the order in which the stream source makes its elements available**

Java Parallel Stream Results Order

- The processing *results* in a parallel stream are more deterministic than ordering.
- Programmers can control if results are presented in “encounter order” (EO).
- EO is maintained if source is ordered & the aggregate operations used are obliged to maintain order.

The processing *results* in a parallel stream are more deterministic than ordering.

- Programmers can control if results are presented in “encounter order” (EO).
  - EO is maintained if source is ordered & the aggregate operations used are obliged to maintain order.
  - The semantics are the same whether the stream is parallel or sequential.

---

Java Parallel Stream Results Order

<table>
<thead>
<tr>
<th>Intermediate operation (behavior f)</th>
<th>Intermediate operation (behavior g)</th>
<th>Terminal operation (reducer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input x</td>
<td>f(x)</td>
<td>Output g(f(x))</td>
</tr>
</tbody>
</table>

**Stream factory operation ()**
The processing results in a parallel stream are more deterministic than ordering.
- Programmers can control if results are presented in “encounter order” (EO).
  - EO is maintained if source is ordered & the aggregate operations used are obliged to maintain order.
  - The semantics are the same whether the stream is parallel or sequential.
- Performance may differ, however.
End of Java Parallel Stream Internals: Order of Results (Part 1)