Understand Java Parallel Streams Internals: Order of Results Overview

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 order of results in a parallel stream is more deterministic than processing order.

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

- The order of *results* in a parallel stream is more deterministic than processing order.
- 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 order of *results* in a parallel stream is more deterministic than processing order.
- 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

See [developer.ibm.com/languages/java/articles/j-java-streams-3-brian-goetz/#eo](developer.ibm.com/languages/java/articles/j-java-streams-3-brian-goetz/#eo)
The order of results in a parallel stream is more deterministic than processing order. 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.
The order of *results* in a parallel stream is more deterministic than processing order.

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