4.2 Lecture Summary

4 Dataflow Synchronization and Pipelining

4.2 Point-to-Point Synchronization with Phasers

Lecture Summary: In this lecture, we looked at a parallel program example in which the span (critical path length) would be 6 units of time if we used a barrier, but is reduced to 5 units of time if we use individual phasers as shown in the following table:

Task 0	Task 1	Task 2
$1a : X = A(); //cost = 1$	$1b : Y = B(); //cost = 2$	$1c : Z = C(); //cost = 3$
$2a : ph0.arrive();$	$2b : ph1.arrive();$	$2c : ph2.arrive();$
$3a : ph1.awaitAdvance(0);$	$3b : ph0.awaitAdvance(0);$	$3c : ph1.awaitAdvance(0);$
$4a : D(X, Y); //cost = 3$	$4b : ph2.awaitAdvance(0);$	$4c : F(Y, Z); //cost = 1$
	$5b : E(X, Y, Z); //cost = 2$	

Each column in the table represents execution of a separate task, and the calls to **arrive**() and **awaitAdvance**(0) represent synchronization across different tasks via phaser objects, **ph0**, **ph1**, and **ph2**, each of which is initialized with a party count of 1 (only one signalling task). (The parameter 0 in **awaitAdvance**(0) represents a transition from phase 0 to phase 1.)

Optional Reading:

1. Documentation on Java Phaser class.

Mark as completed