Phaser Beams: Combining Task and Stream Parallelism

What we used
Phasers for point to point or collective synchronization primitive
Phaser accumulators – reduction construct based on phasers

What we did
Phaser accumulators extensions:
• Bounded phasers limit the phase difference between a consumer and a producer
• Bounded buffer for accumulators • with eager/lazy policies
• Dynamic cycle detection to allow batching

What can we use it for
Expressing streaming programs in an imperative task level programming language (Habanero Java)...
Pipeline and Split Join Streaming Patterns:

...or combining task and stream parallelism
In the declarative, high level Concurrent Collections model, we replace the implementation of the CnC basic blocks (item collections, tag collections) with the phaser beams based model to get the Streaming CnC model.

Results
We accept restrictions on what streams we can express:
• only one input tag collection • no feedback loops
But in return we get:
• more flexibility (dynamic graph structure) • higher performance • no need to write low level detailed streaming code

Phasers: a unified deadlock-free construct for collective and point-to-point synchronization, Jun Shirako at al, ICS 08, 2008
StreamIt: A language for streaming applications, W. Thies et al., CCA, 2002