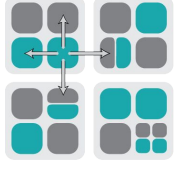
PDES in an Adaptive Runtime System

UIUC: Eric Mikida, Nikhil Jain, Joshua Lew, Laxmikant Kale
RPI: Chris Carothers, Elsa Gonsiorowski
LLNL: David Jefferson, Peter Barnes

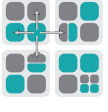
PPL UIUC PARALLEL PROGRAMMING LABORATORY



Charm++

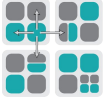
What is PDES?

- Simulation of events at discrete points in virtual time
  - Contrary to time-stepped, work is not uniform/dense in time
  - Events must be executed in increasing order of virtual time
  - Traffic, Supercomputer Interconnects, Circuits, Battle Sims
- Events executed by Logical Processes (LPs)
- Synchronization required to maintain event order
- Focus is on ROSS and the Time Warp protocol

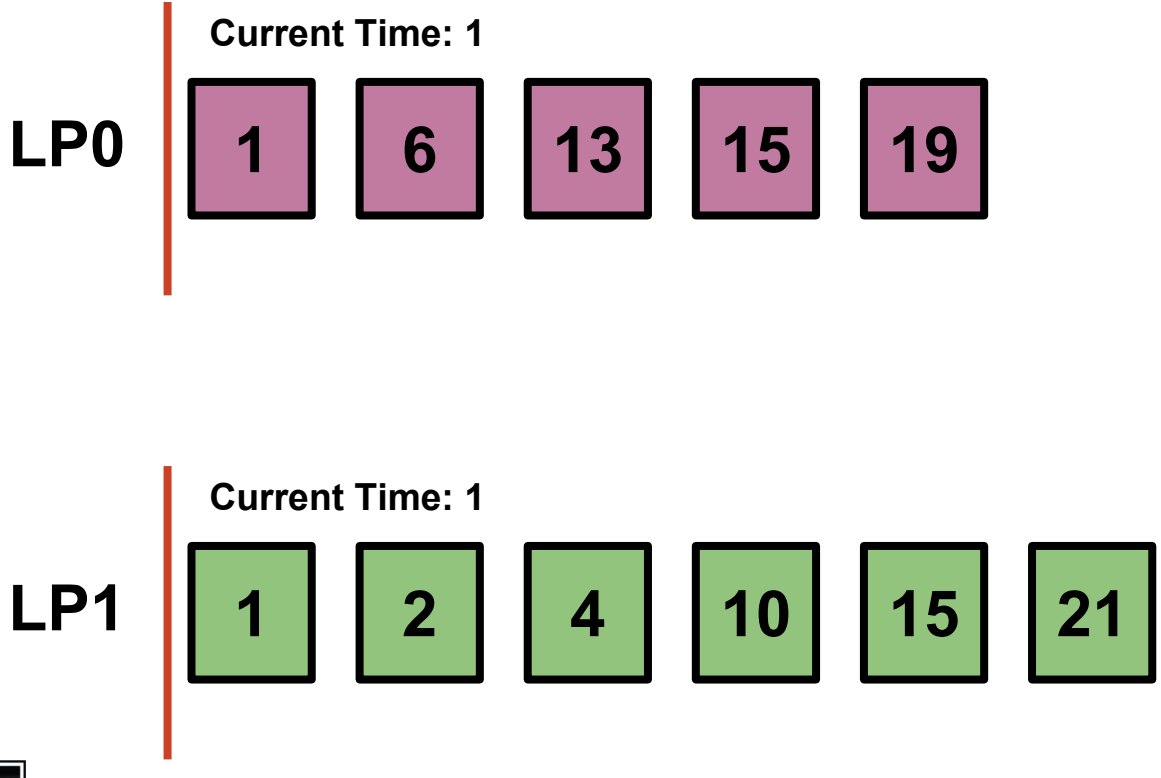


Basic Terminology

- Logical Process (LP)
- Total Executed Events
- Committed Events
- Rolled-back Events
- Global Virtual Time (GVT)
- Fossil Collection
- Event Efficiency
- Event Rate

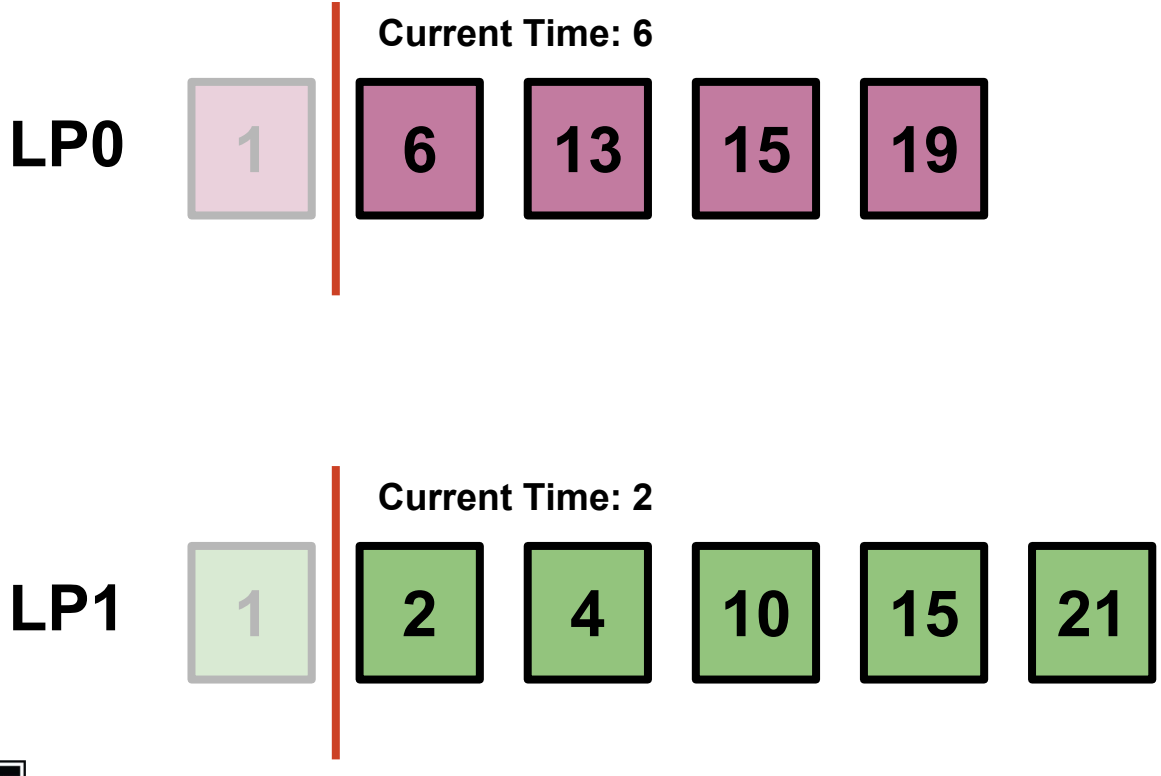


A (Small) Example



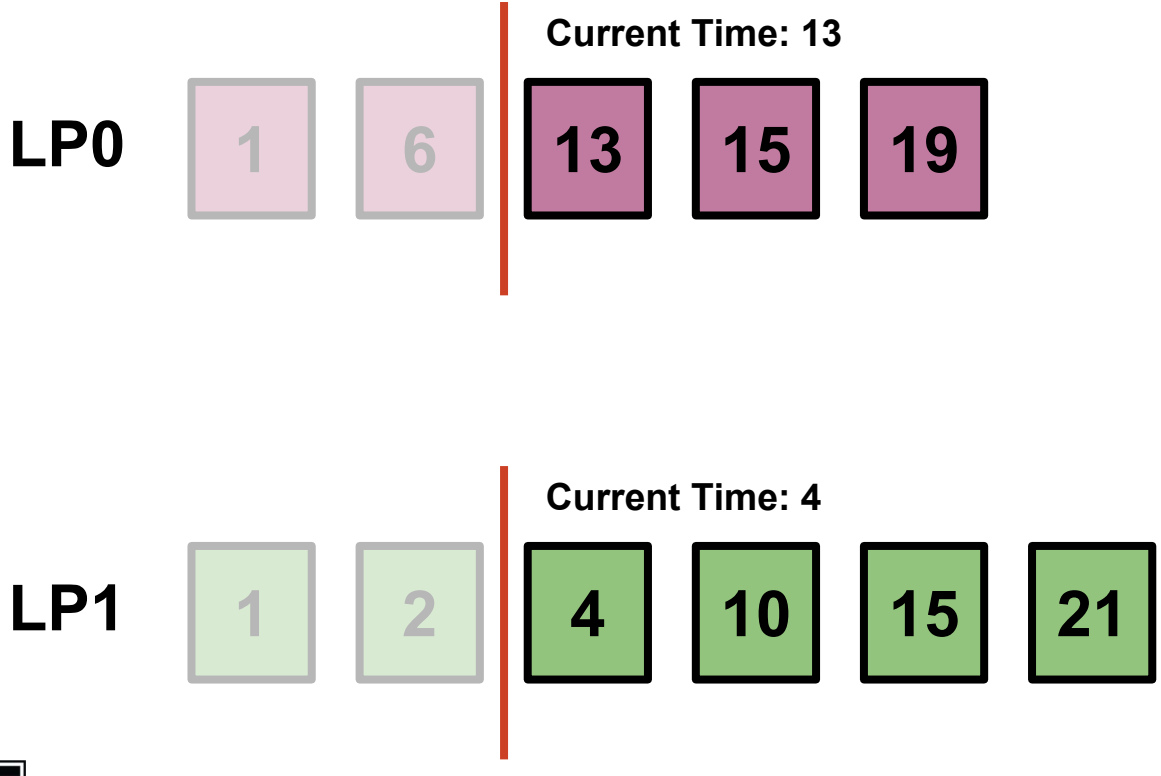
Total Events: 0
Total Rollbacks: 0
Committed Events: 0

A (Small) Example



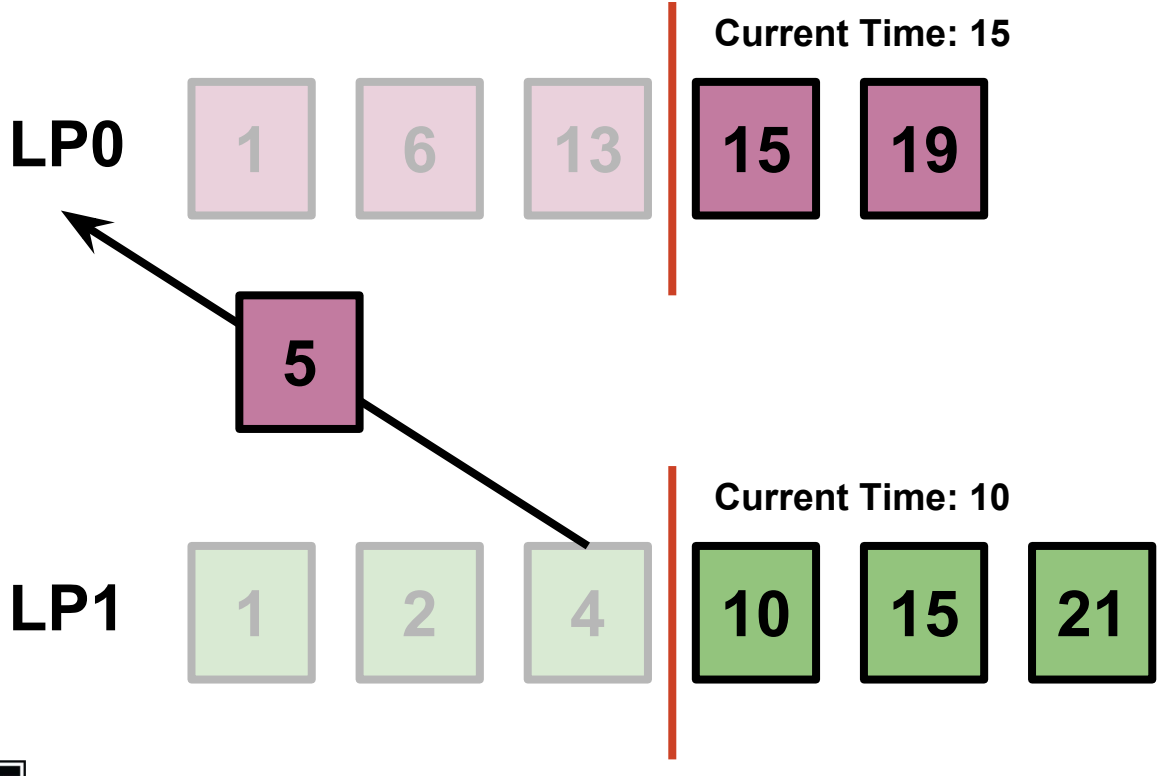
Total Events: 2
Total Rollbacks: 0
Committed Events: 0

A (Small) Example



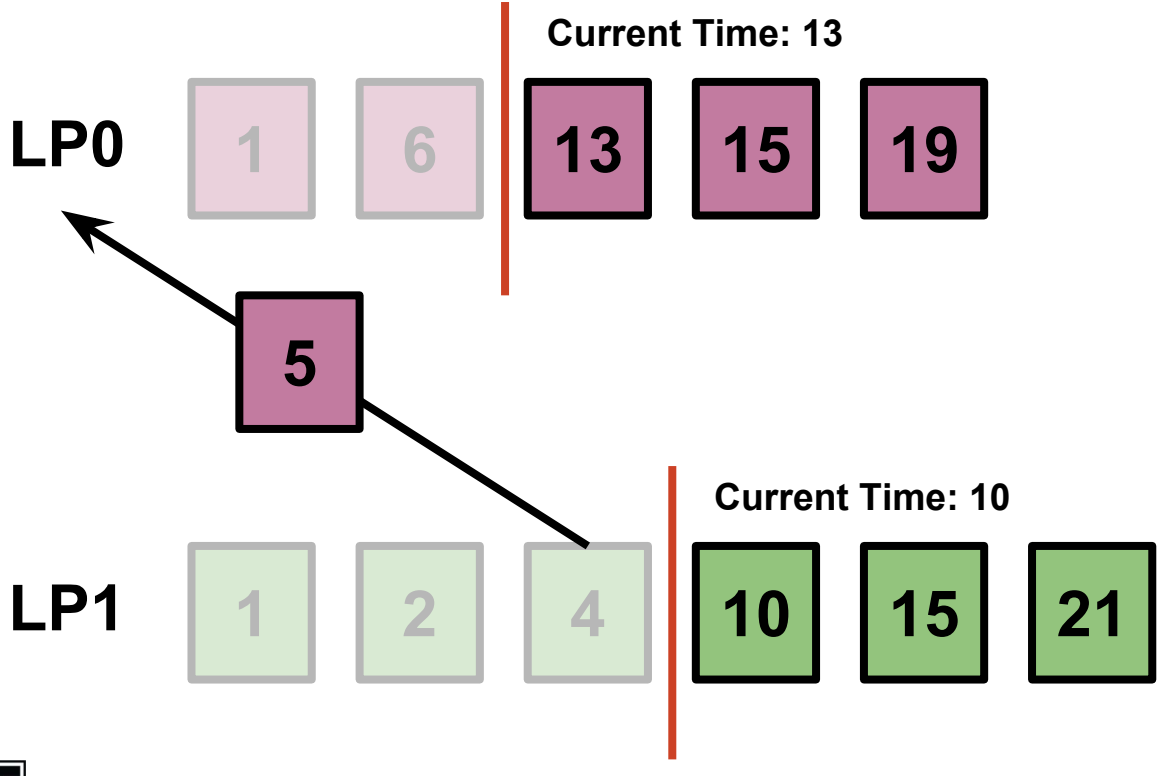
Total Events: 4
Total Rollbacks: 0
Committed Events: 0

A (Small) Example



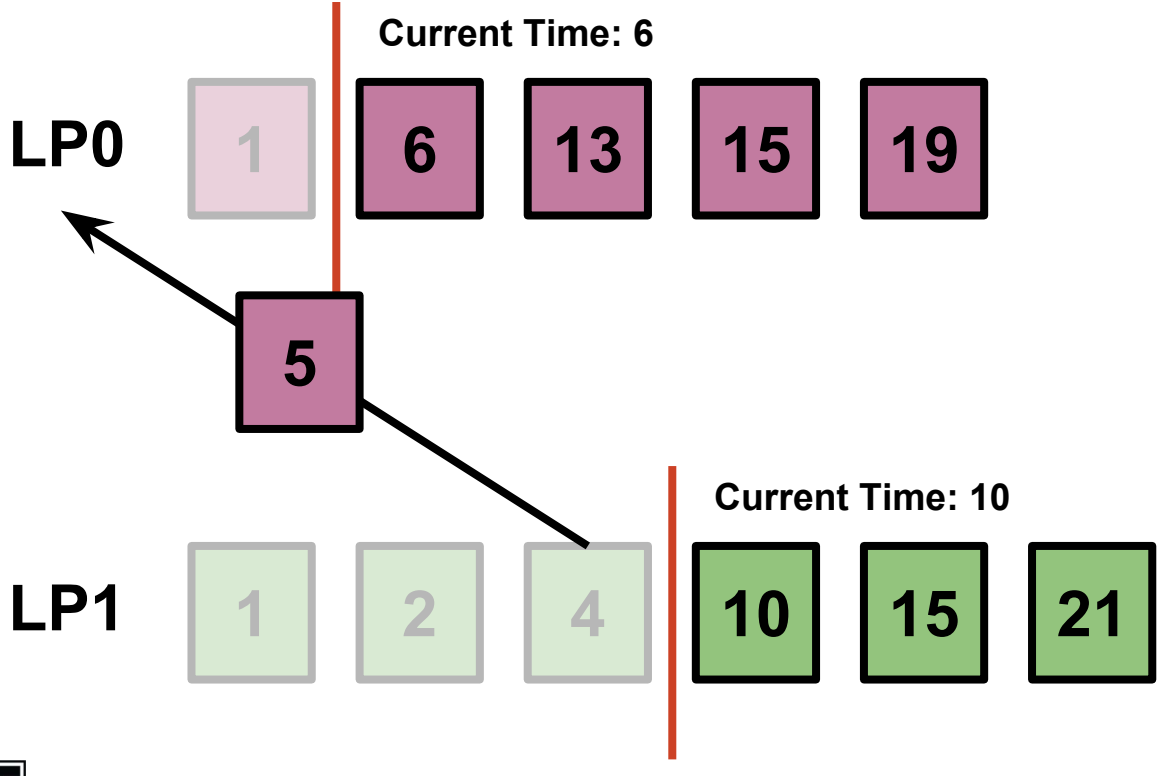
Total Events: 6
Total Rollbacks: 0
Committed Events: 0

A (Small) Example



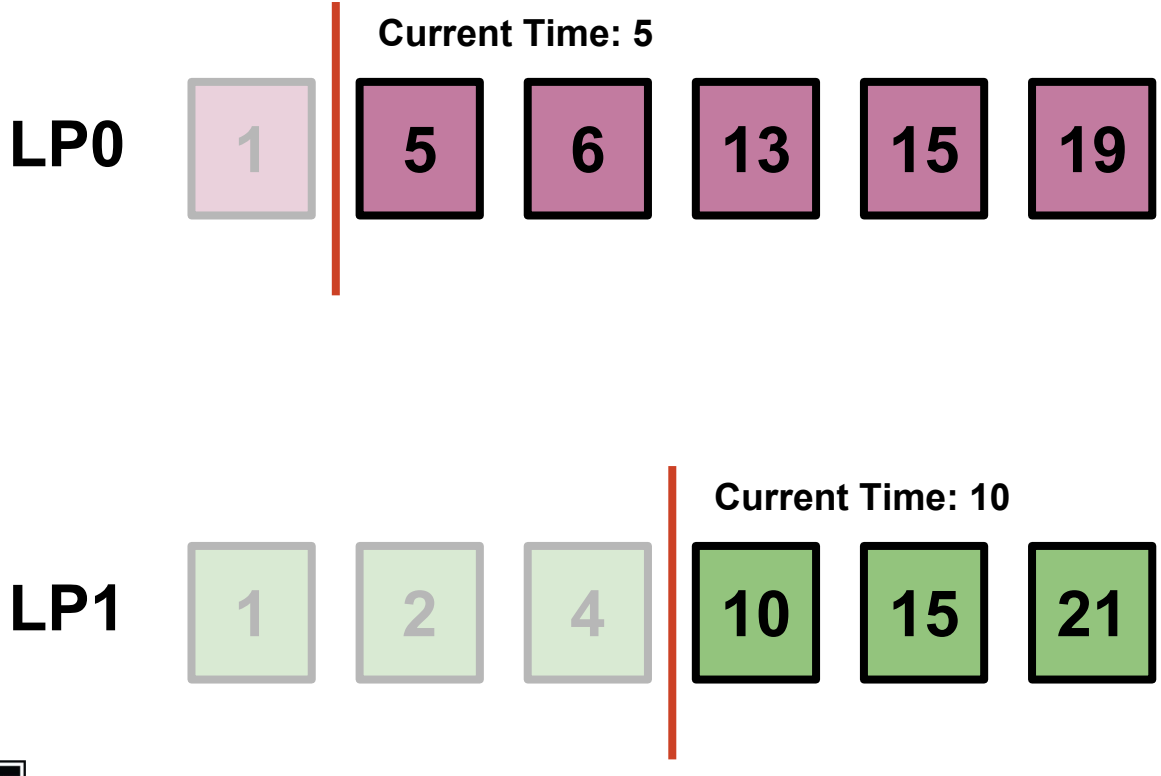
Total Events: 6
Total Rollbacks: 1
Committed Events: 0

A (Small) Example



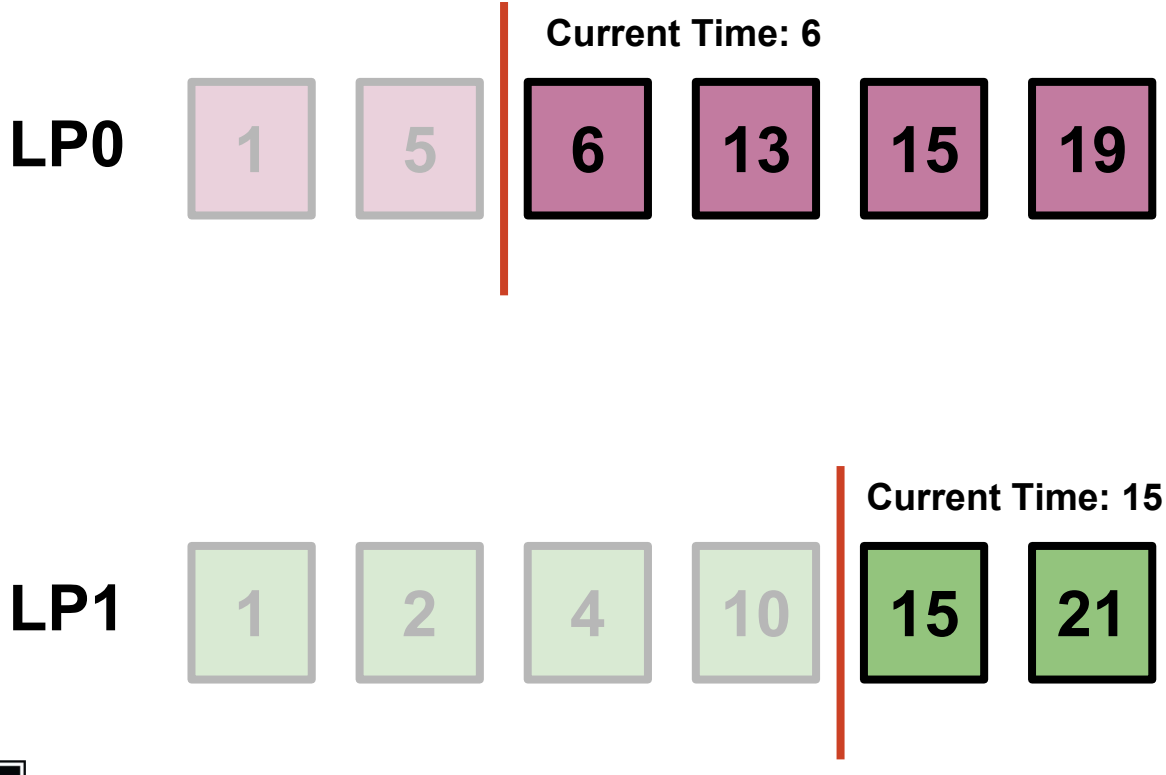
Total Events: 6
Total Rollbacks: 2
Committed Events: 0

A (Small) Example



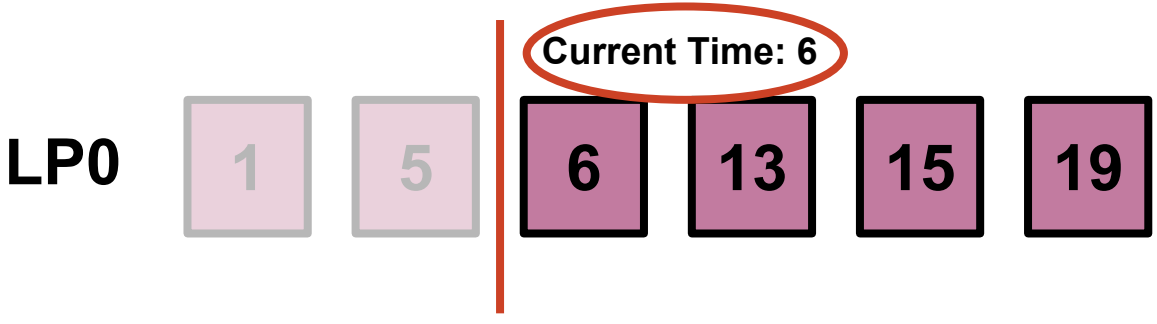
Total Events: 6
Total Rollbacks: 2
Committed Events: 0

A (Small) Example



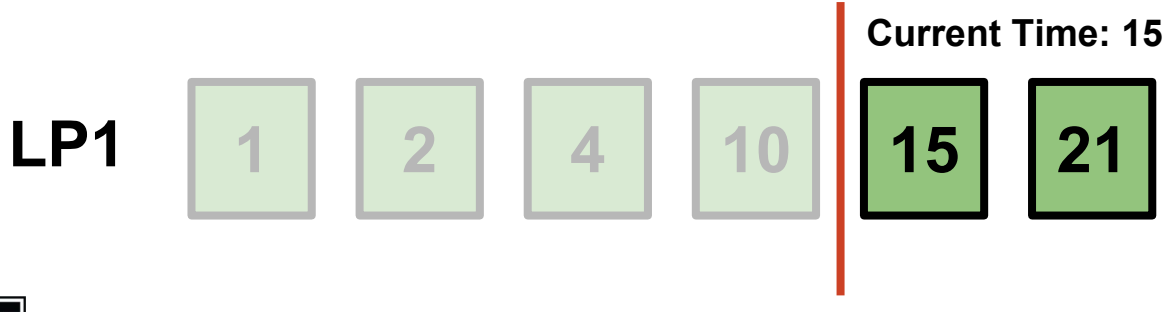
Total Events: 8
Total Rollbacks: 2
Committed Events: 0

A (Small) Example

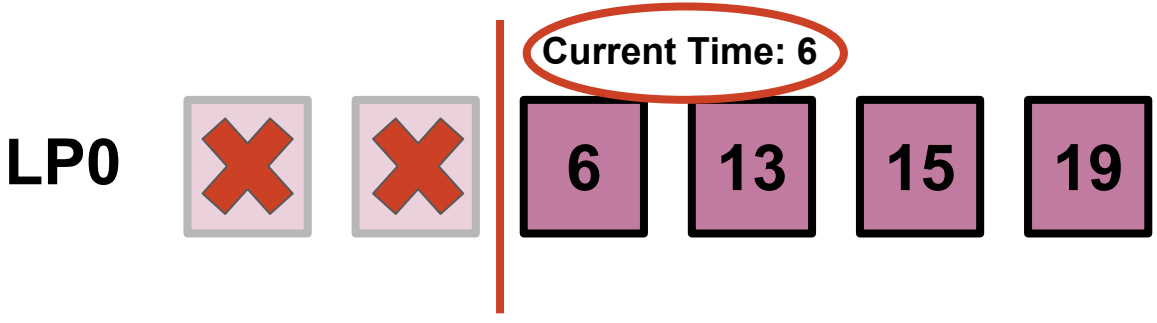


Total Events: 8
Total Rollbacks: 2
Committed Events: 0

GVT COMPUTATION: Find the minimum time

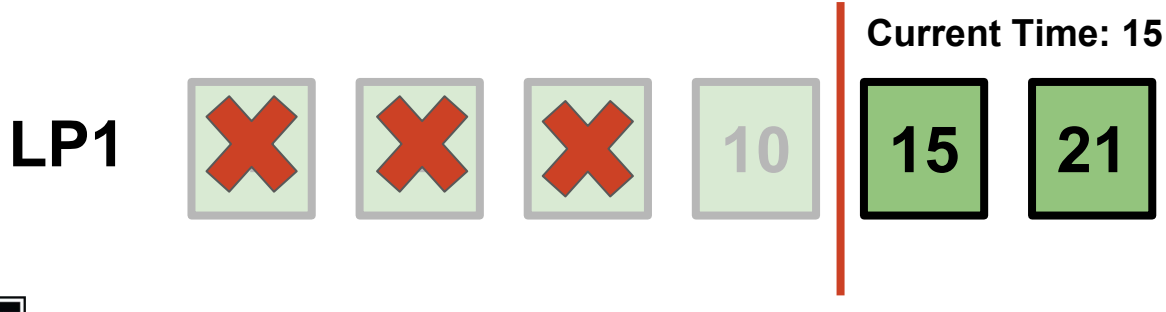


A (Small) Example

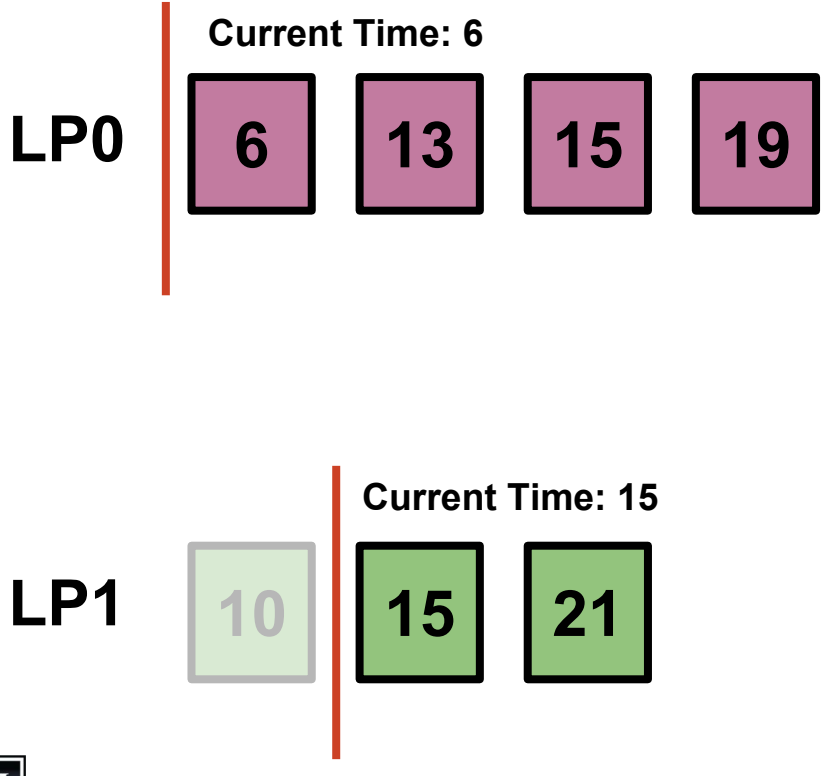


Total Events: 8
Total Rollbacks: 2
Committed Events: 5

FOSSIL COLLECTION: Commit and free events



A (Small) Example

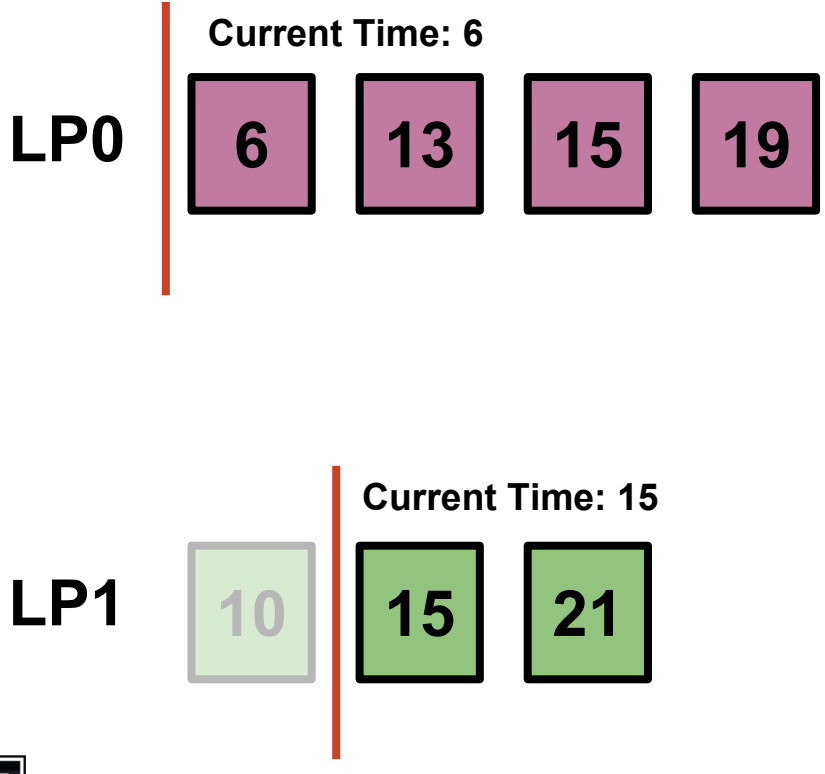


Total Events: 8
Total Rollbacks: 2
Committed Events: 5

Event Efficiency: $(E_C - E_R)/E_C$

(5-2)/5 = 60%

A (Small) Example



Total Events: 8
Total Rollbacks: 2
Committed Events: 5

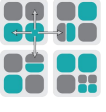
Event Efficiency: $(E_C - E_R)/E_C$

(5-2)/5 = 60%

Event Rate: E_C / t

Previous Work

- Implemented a version of ROSS on top of Charm++
  - LPs are chares: adaptive overlap, migration, location mgmt.
  - Good match to programming model: 7,277 ⇒ 3,991 SLOC
- Improved performance on PHOLD and Dragonfly models
  - Up to 40% higher event rate for PHOLD
  - Up to 5x higher event rate for Dragonfly model



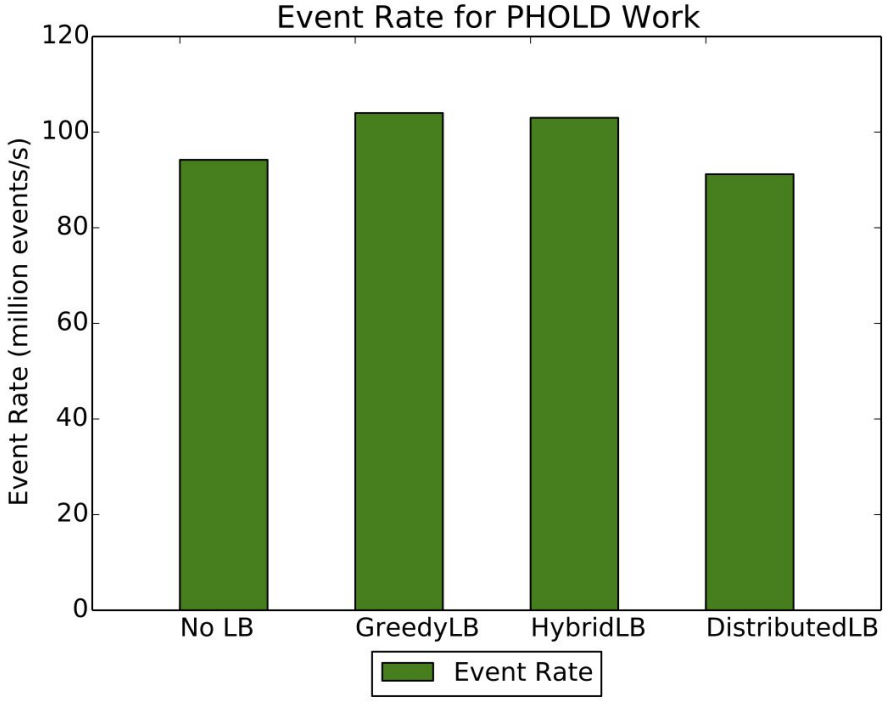
New Work - Load Balancing

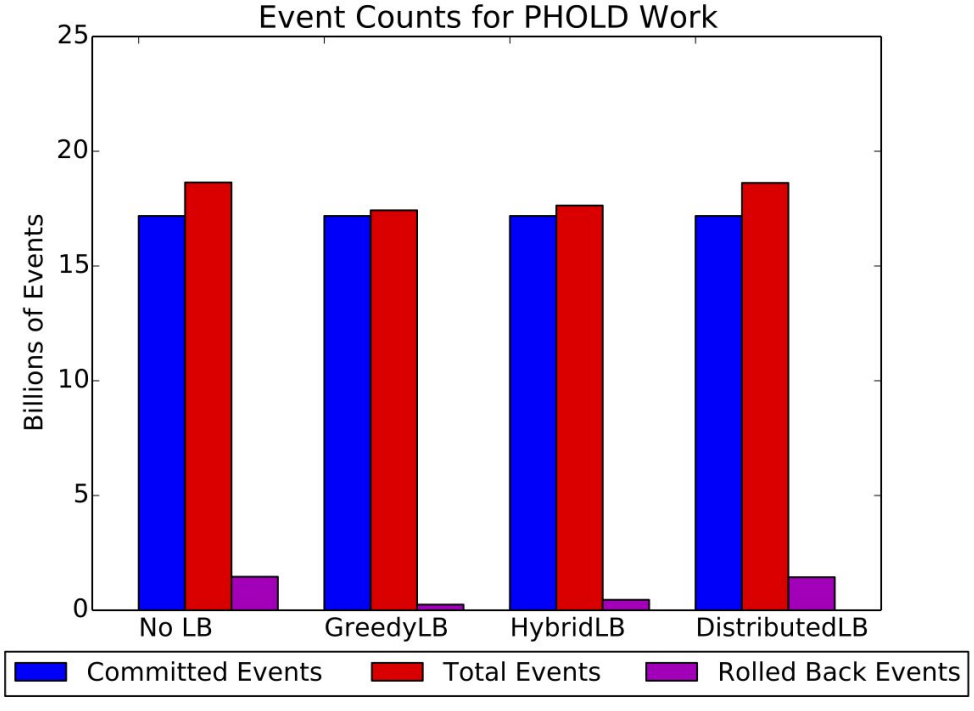
- LPs are now migratable by the runtime system
- Load of each chare measured by RTS
- Three different load balancing strategies:
  - GreedyLB
  - DistributedLB
  - HybridLB

Load Balancing - PHOLD

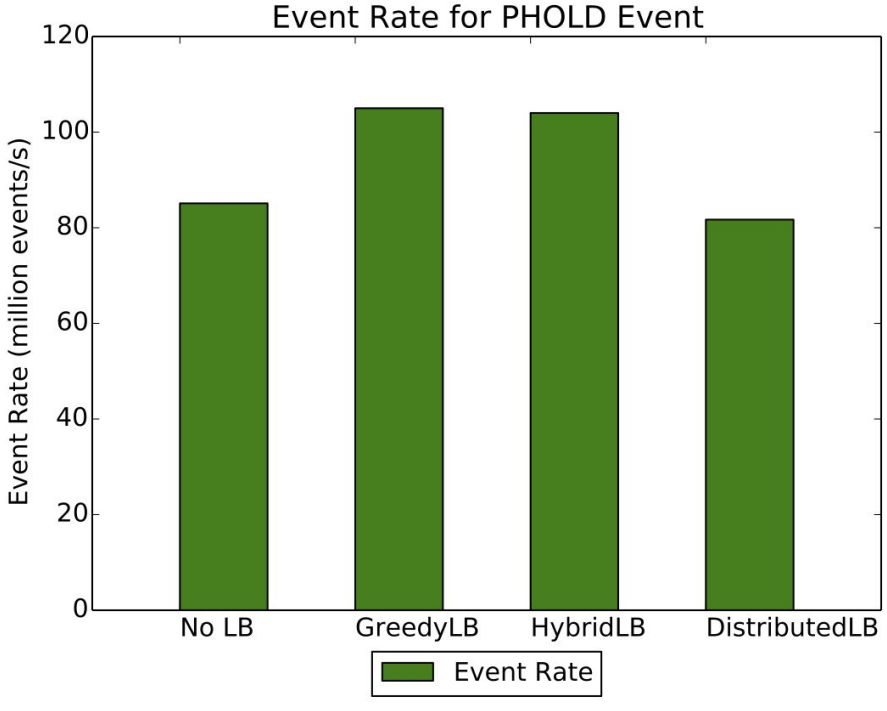
- Basic PHOLD micro-benchmark to test comm loads
  - Each event causes a new event to be created/sent
  - New events are remote sends with probability *P*
  - Results in a uniform and balanced execution
- Added two sources of imbalance
  - Work imbalance: some LPs take longer per event
  - Event imbalance: some LPs receive more events
- All runs on 64 nodes of Vesta (BG/Q machine at ANL)

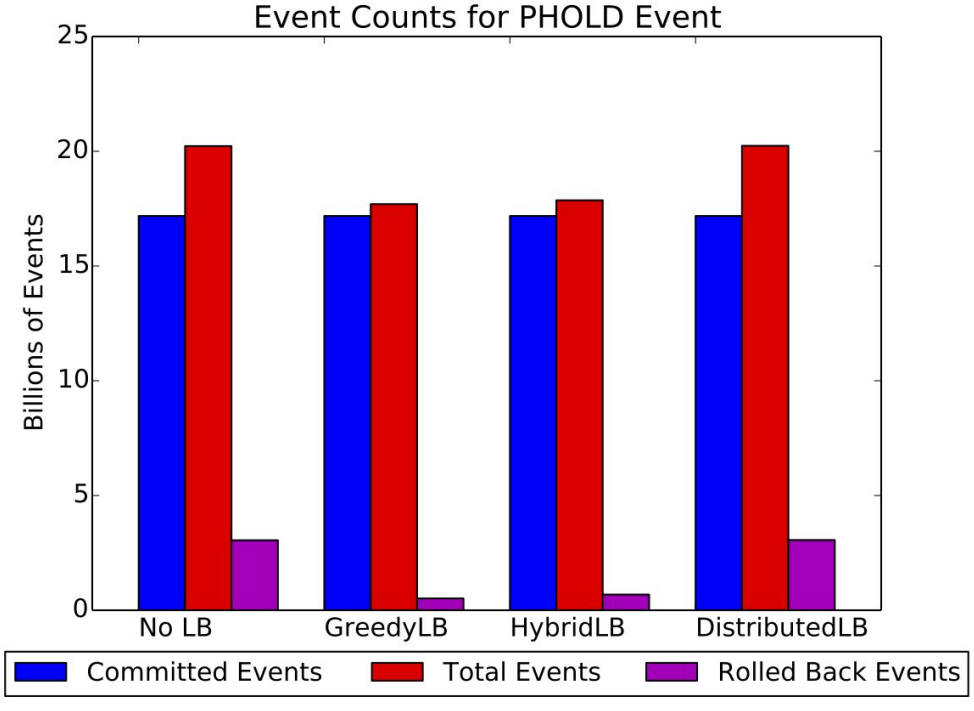
Load Balancing - PHOLD



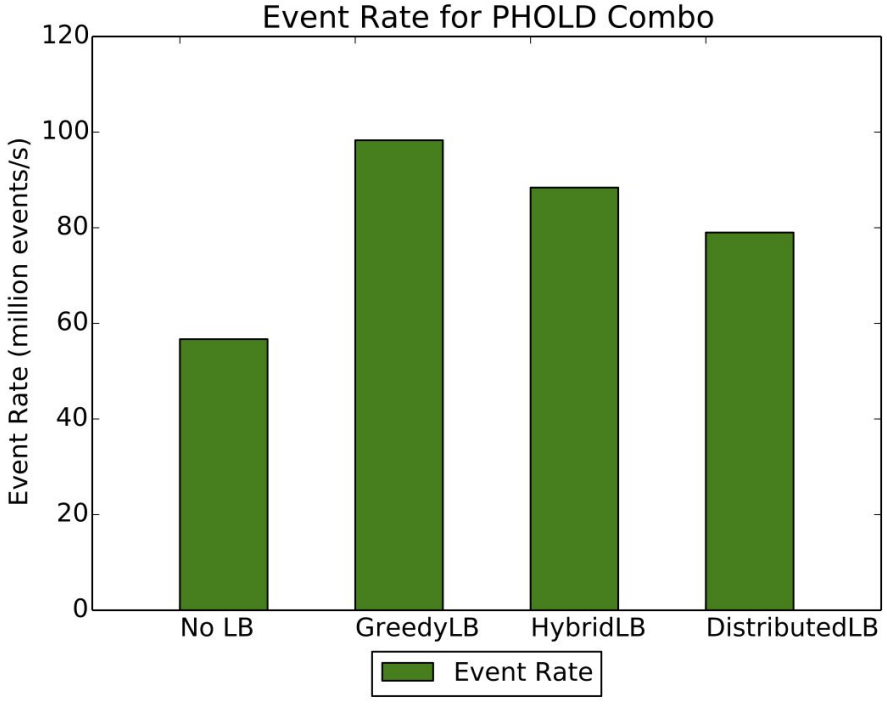


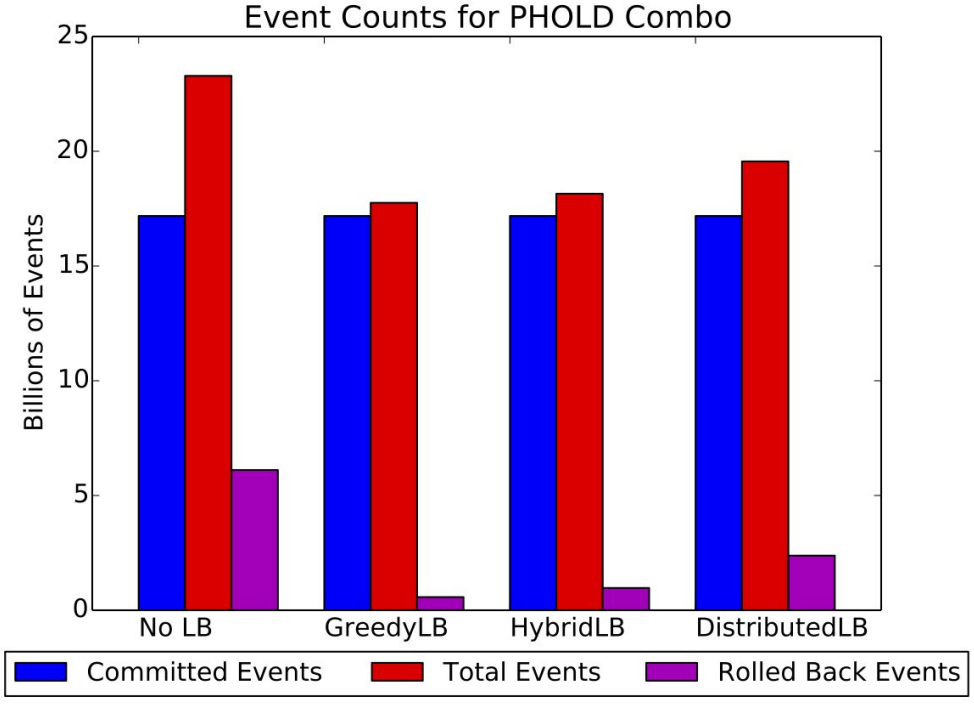
Load Balancing - PHOLD





Load Balancing - PHOLD





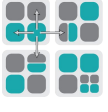
Load Balancing - Overhead

- Data collection, decision cost, migration cost vary
- Reasons to use lower overhead strategies
  - Shorter duration simulations
  - Simulations that require frequent load balancing
  - Simulations with high memory consumption

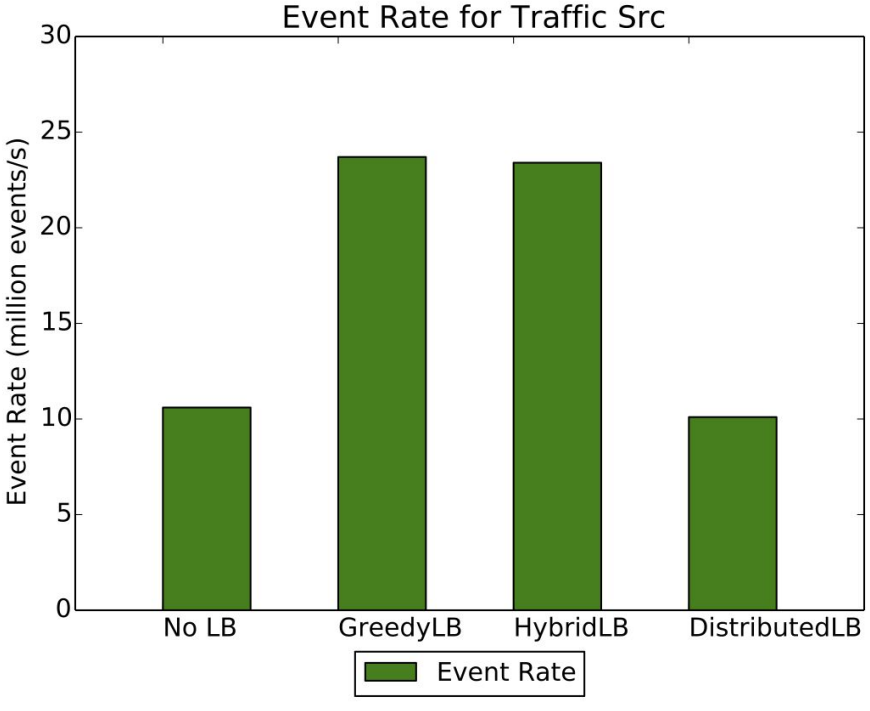
	No LB	GreedyLB	HybridLB	DistributedLB
Total Runtime	311 seconds	179 seconds	247 seconds	321 seconds
LB Time	N/A	8 seconds	0.9 seconds	0.02 seconds

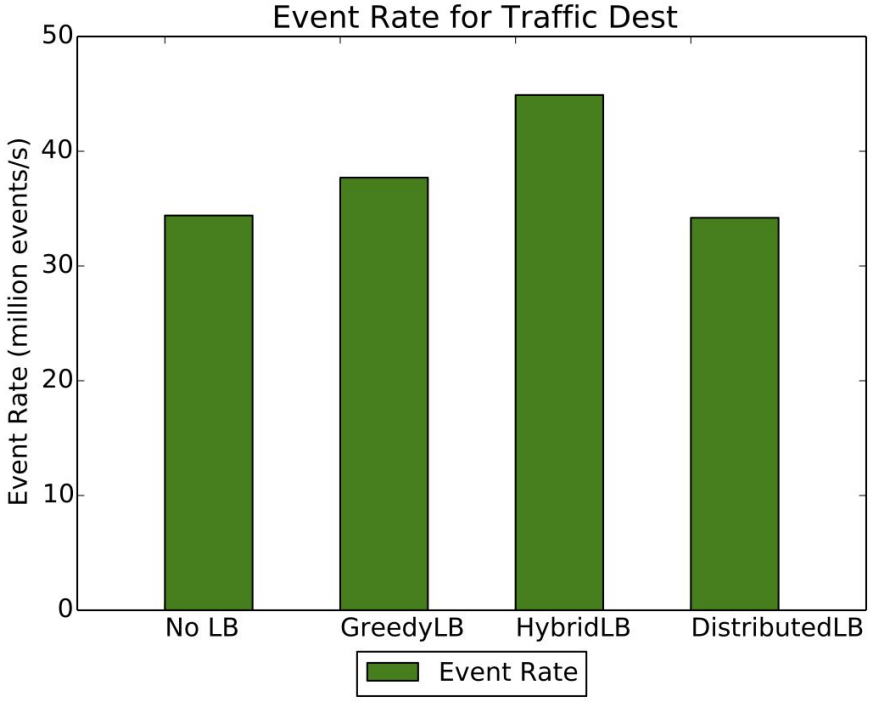
Load Balancing - Traffic

- LPs are intersections arranged in a grid
  - Events are cars arriving, departing, changing lane
  - Cars travel from source to destination
  - Roads have a capacity, cars wait until a road is free
- Two imbalanced configurations
  - Source congestion - many cars start from one place
  - Destination congestion - many cars going to one place



Load Balancing - Traffic





Future Work

- Look deeper into DistributedLB
- Look into other strategies (communication aware)
- More interesting load metrics focused on event efficiency
- Joining GVT improvements with load balancing