

Year in Review

- Historical context: Project on “Extending OpenSHMEM to Support Asynchronous Execution Models” funded by LANL
 - Work done at Rice in 2016 & 2017
 - Habanero group moved from Rice to Georgia Tech in Fall 2017
 - Project will continue at Georgia Tech in 2018
- Graph500 on OpenSHMEM
 - Joint Rice-LANL PAW17 paper “Graph500 on OpenSHMEM: Using A Practical Survey of Past Work to Motivate Novel Algorithmic Developments”
 - Poster at SC17 PGAS Booth
- Exploration of OpenSHMEM Contexts
 - Joint Rice-Intel-LANL OpenSHMEM17 paper “Implementation and Evaluation of OpenSHMEM Contexts Using OFI Libfabric”
- Recently began work supporting development of OSSS OpenSHMEM on UCX, led by SBU
-  HOOVER: Distributed, Dynamic Graph Modeling and Analysis on OpenSHMEM (**focus of today’s talk**)

HOOVER: Distributed, Dynamic Graph Modeling and Analysis on OpenSHMEM

Max Grossman, Vivek Sarkar
Georgia Tech/Rice University



Goal: Build a general framework for creating dynamic graph applications on top of OpenSHMEM.

General – express a wide variety of applications (infectious diseases, insider threat, social networks, etc)

Programmable – user fills in application-specific kernels, no awareness of distribution.

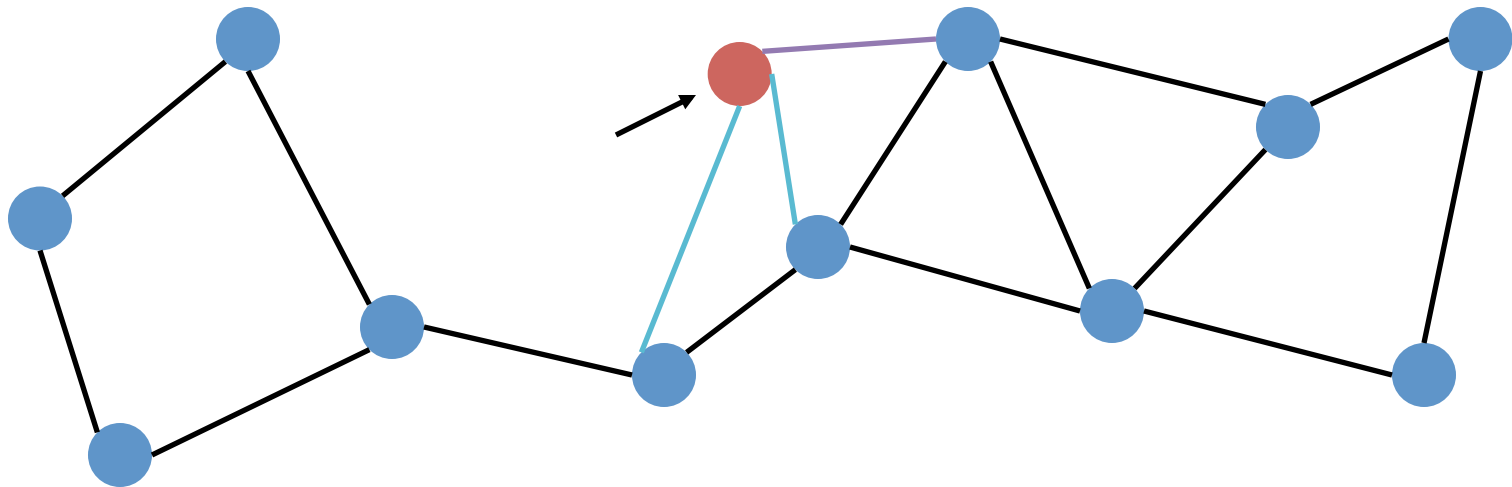
Scalable – fully decoupled execution, no barriers, minimal synchronization, minimal memory overheads.



HOOVER Programming Model

Graph consists of actors with positional and logical attributes; updates to positional attributes define “movement”.

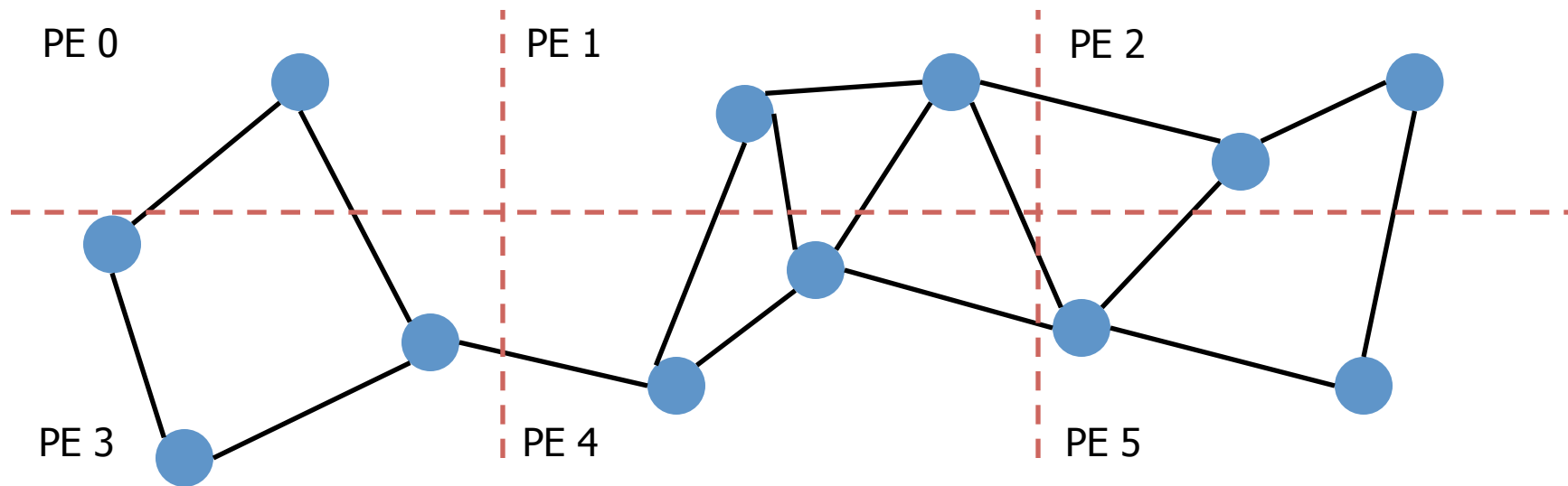
Positional attributes define some idea of location/distance relative to other actors, edges are dynamically updated based on location.



HOOVER Programming Model

Actors are partitioned across OpenSHMEM PEs (user-defined).

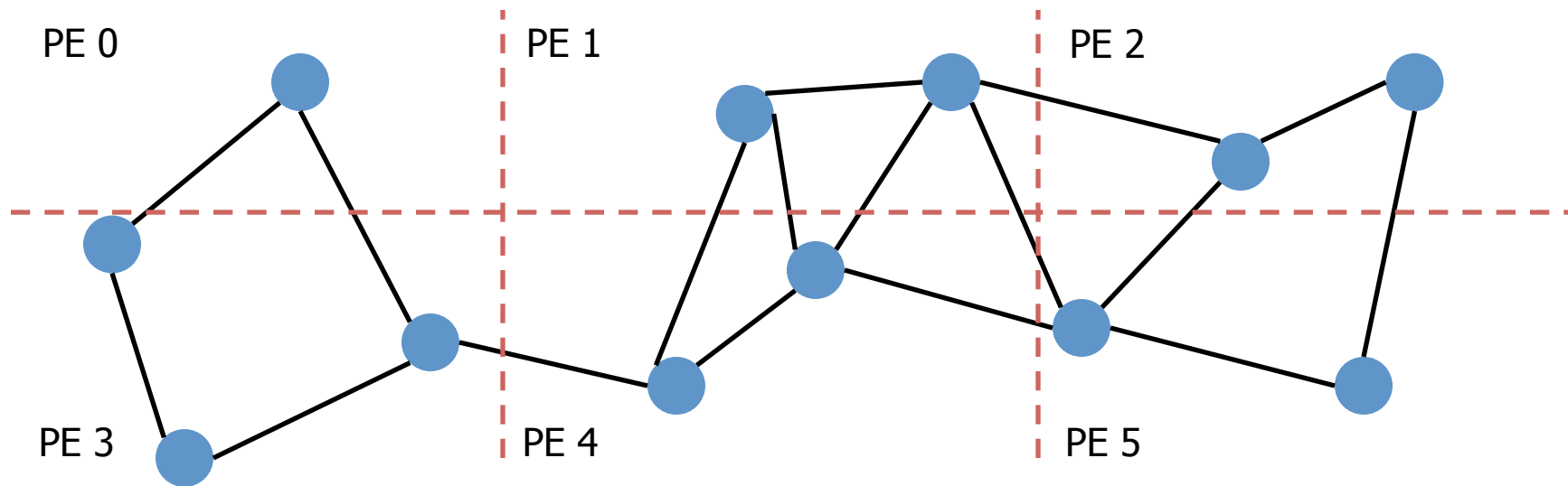
PEs start *entirely de-coupled* (use RDMA to grab data when they need it, no synchronization). PEs become “coupled” at user direction, i.e. lockstep.



HOOVER Programming Model

Actors are partitioned across OpenSHMEM PEs (user-defined).

PEs start *entirely de-coupled* (use RDMA to grab data when they need it, no synchronization). PEs become “coupled” at user direction, i.e. lockstep.



HOOVER Status

Currently in a private Github repo: <https://github.com/agrippa/hover>

- Shoot me an e-mail for access – max.grossman@rice.edu

Core framework and capabilities complete.

Simple visualization support complete.

Preliminary use cases built and being explored, used to motivate optimizations to the framework:

- Static mesh
- Infectious disease modeling
- Insider threat detection (to begin soon)



Quick Demo



Acknowledgements



Stony Brook
University

UNIVERSITY of
HOUSTON