Decoupling Application and Runtime Data in Graphical Simulations
Chinmayee Shah, Omid Mashayekhi, Hang Qu, Philip Levis

**Nimbus**
Graphical simulations are limited to supercomputers or a HPC cluster. Nimbus is a distributed system for running graphical simulations in the computing cloud.

**Distributed Simulations Today**
- Simulations *statically partition* spatial domain and map each partition to a fixed worker.
- All steps use *same partitioning* strategy.
- Workers run in *lock-step*, and keep CPUs idle or busy with wasteful computation.
- Simulations assume that *resources are uniform* & always available, which may not hold in the cloud.

**Why not use current cloud systems?**
- Graphical simulations operate over *geometric data*, making data and task placement important.
- They use *complex and coupled data structures*.
- Computation intensity varies across space and time.
- Simulations are iterative, with *dynamic job and data dependencies* that are not known in advance.

**System Design an Programming Model**
- Central controller
  - assigns jobs
  - manages data exchange and versions
- Workers
  - manage thread pools
  - execute jobs
- Every job has explicit read, write and before set dependencies to minimize scheduling overhead.

**Application and Data Partitions**
- Nimbus runtime versions, copies and exchanges disjoint logical data objects.

**Application Data Manager**
- Translates between logical data objects to contiguous app objects.
- Constructs an app object that matches a job’s app partition, read and write set, and caches across jobs.

![Diagram](image)

- Ensures that app and copy jobs access data with the correct version from the right app object.
- Ensures consistency with controller view of data.
- Copies old data from app objects if old versions must persist for jobs yet to be executed or fault tolerance.
- Reduced time by 50%, but doubled memory usage.

**Status**
- Ported a Physbam water simulation to Nimbus.
- Run simulations up to $2000^3$ on Amazon EC2.
- Future work includes eliminating double copies, adding runtime support for trees and chimera grids.