Enabling Simulation and Information Technologies
Solutions Schemes and
Challenges for Very large Models

Boris Jeremić

Department of Civil and Environmental Engineering
University of California, Davis

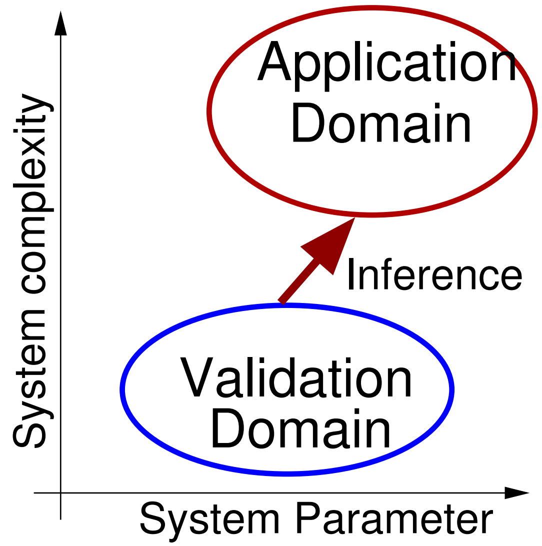
Motivation

- Create high fidelity models of constructed facilities (bridges, buildings, port structures, dams...).
- Models will live concurrently with the physical system they represent.
- Models to provide owners and operators with the capabilities to assess operations and future performance.
- Use observed performance to update and validate models through simulations.

- Niklaus Wirth's Law (NK of Pascal and Modula fame): *Software is slowing faster than the hardware is accelerating*
- in other words:

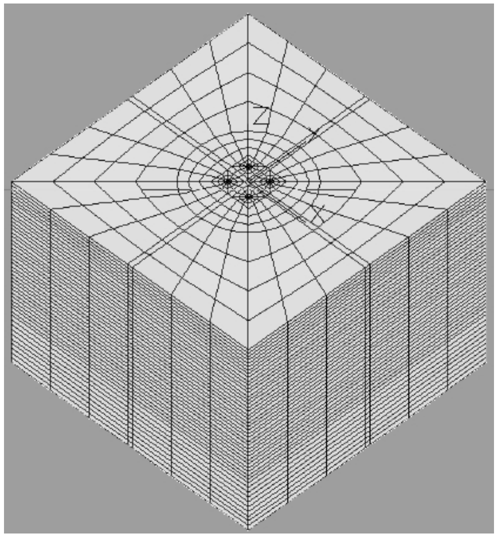
Moore giveth, and OpenSees taketh away.

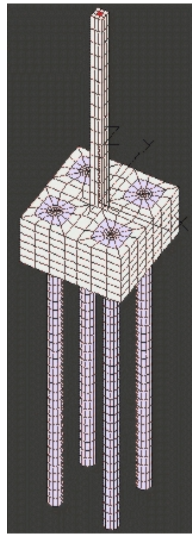
Application Domain

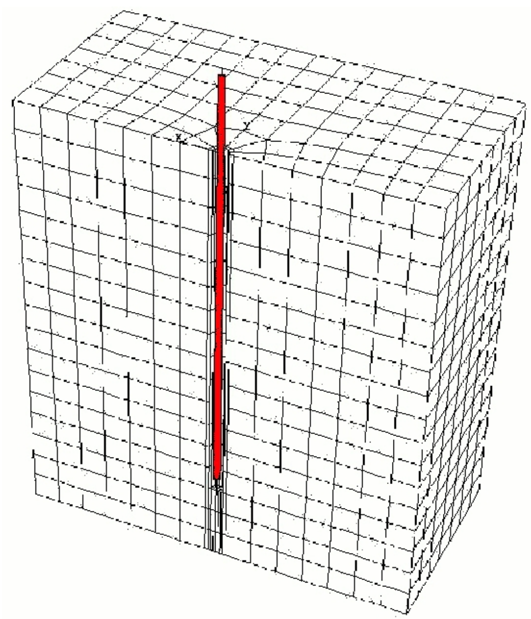


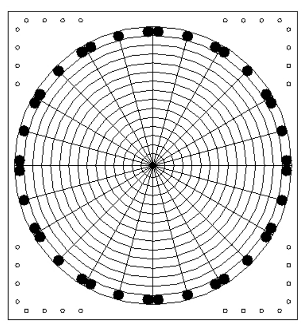
- Inference $\Rightarrow$ Based on **physics** or **statistics**
- Validation domain $\longrightarrow$ non–convex aggregation of tests
- Physical experiments provide for non–overlapping validation domain

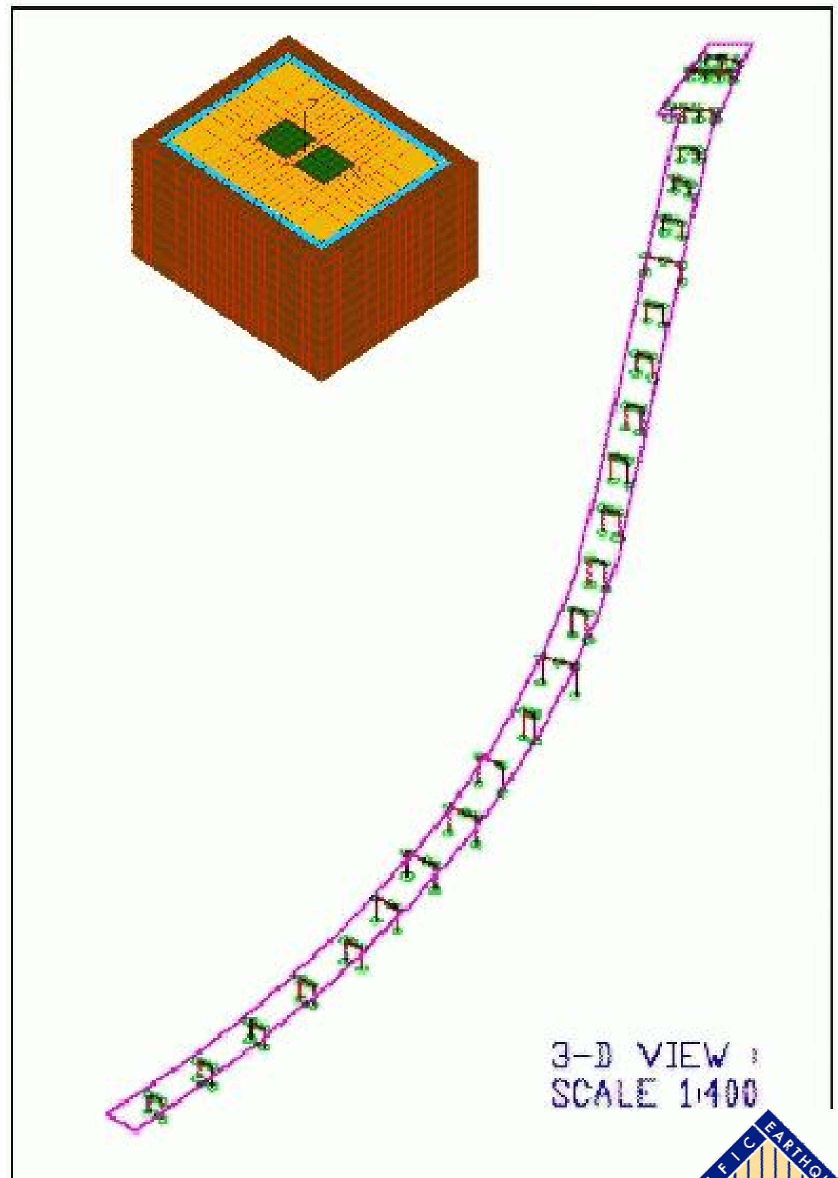
Somewhat Detailed Models









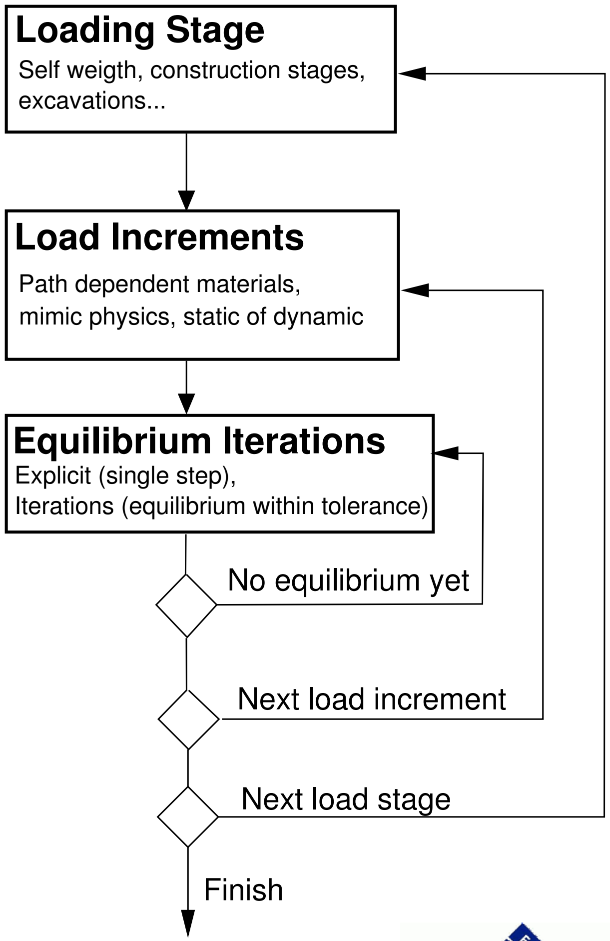


Large Scale Numerical Simulation Goal

- Scalable parallel finite element method for inelastic computations (solid and structural elements)
- Available for a range of sequential and parallel machines, including clusters, grids of machines and clusters (DMPs), and also multiprocessors (SMPs)
- Public domain, portable platform.
- Extensible (directly usable!) to a wide area grids or resources

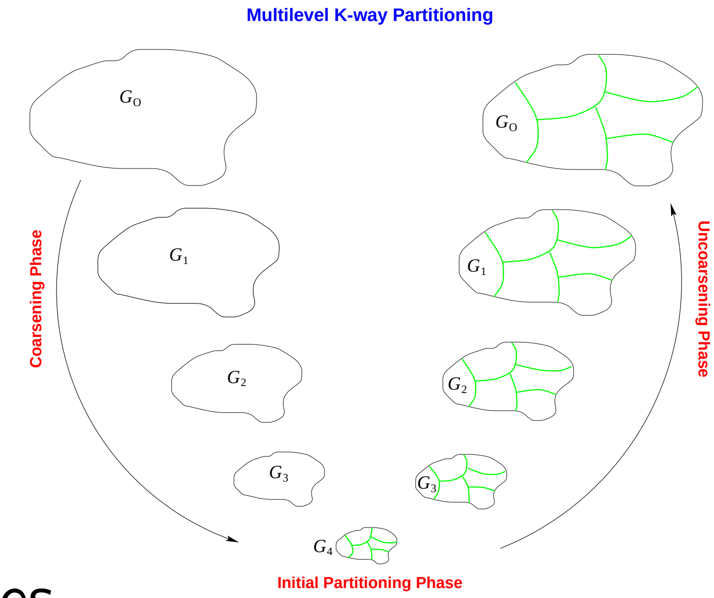
Inelastic Parallel Problem

- Presence of elastic and inelastic computational domains.
- Difference in computational load for elastic and inelastic state determination leads to computational load imbalance
- This leads to imbalanced computations, very inefficient, not much gain from using parallelization
- Internal state determination can take as much as 80% of CL



Development of the Plastic Domain Decomposition

- Based on work of Karypis et al.
- Multilevel Graph Partitioning
  - graph coarsening
  - initial partitioning
  - multilevel refinement
- Par–METIS system
- Weighted graphs edges and graph nodes
- Zoltan framework used as an interface layer for extensibility

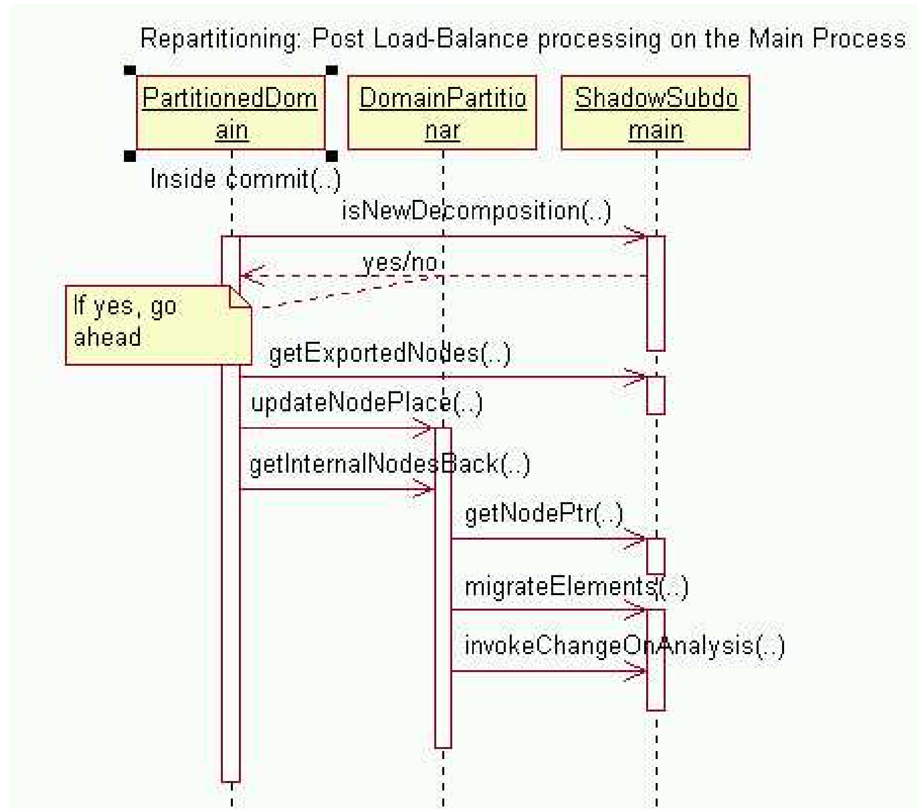


Plastic Domain Decomposition

- Graph partitioning $\longrightarrow$ balance multiple phases simultaneously, while also minimizing the inter-processor communications costs
- It is a multi-objective optimization problem (minimize both the inter-processor communications, the data redistribution costs and create balanced partitions)
- Take into the account (deterministic or probabilistic):
  - heterogeneous element loads that change in each iteration
  - heterogeneous processor performance (multiple generations nodes)
  - inter-processor communications (LAN or WAN)
  - data redistribution costs
- Follows standard interface (more or less)
- Works on SMPs, local DMPs, grids of computers

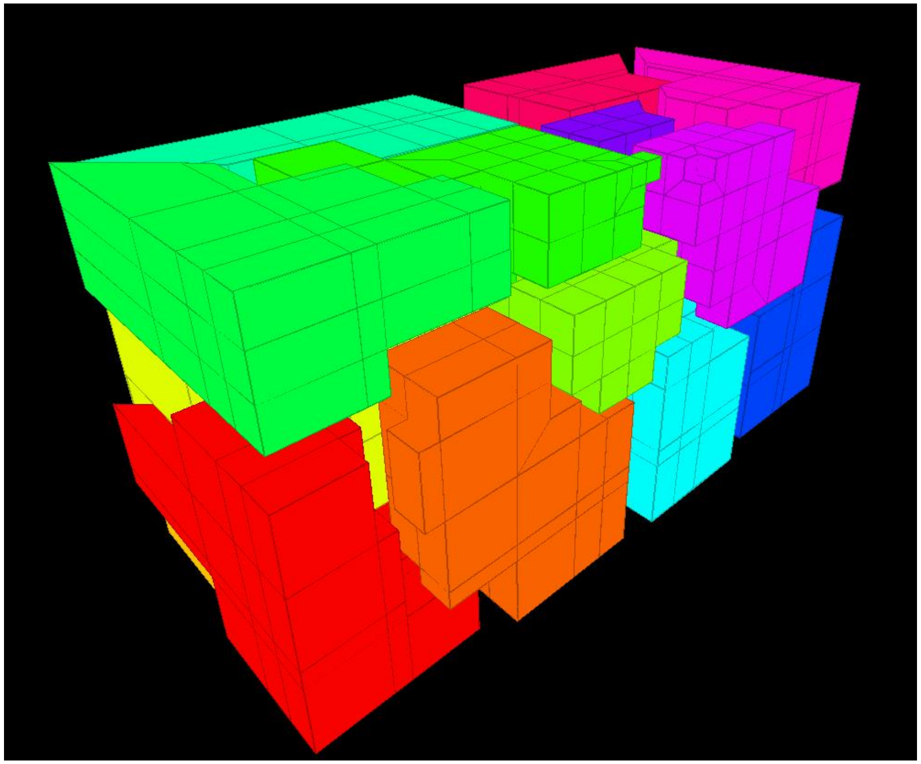
PDD Design

- Work flow sequencing (Rational Rose)
- Designed and tested on a low budget local DMP (LAN and WAN)
- Extension to tasks other than internal state determination and SES (remote data bases, model discovery, visualization)



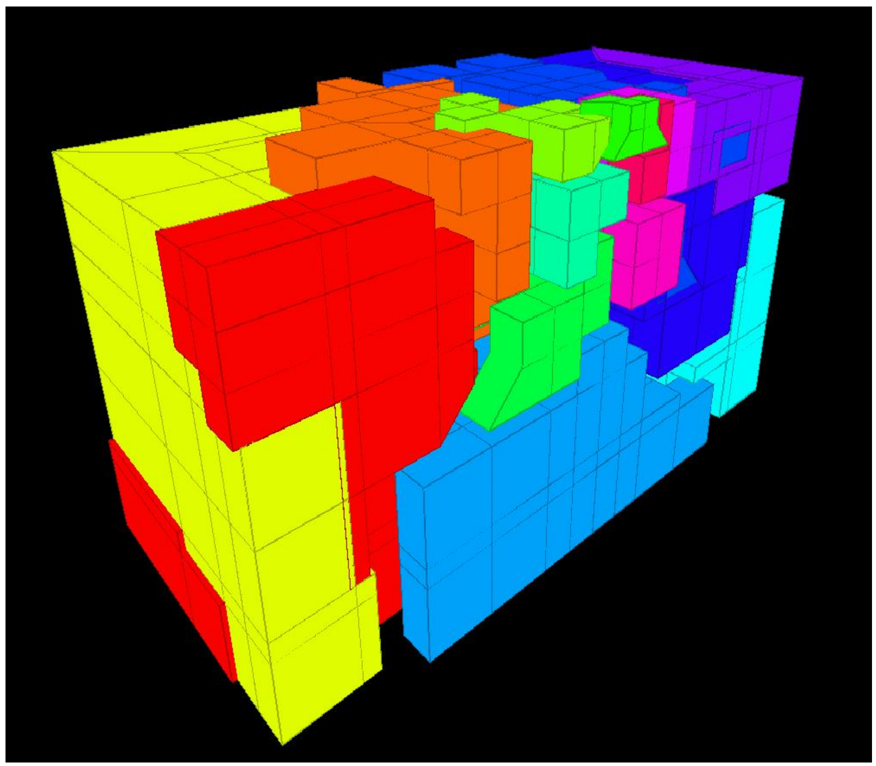
PDD: Elastic Decomposition

- 16 subdomains,
- approx. equal number of elastic elements per domain,
- minimized inter–domain boundary
- OpenSees $\Rightarrow$ Zoltan $\Rightarrow$ Par–METIS



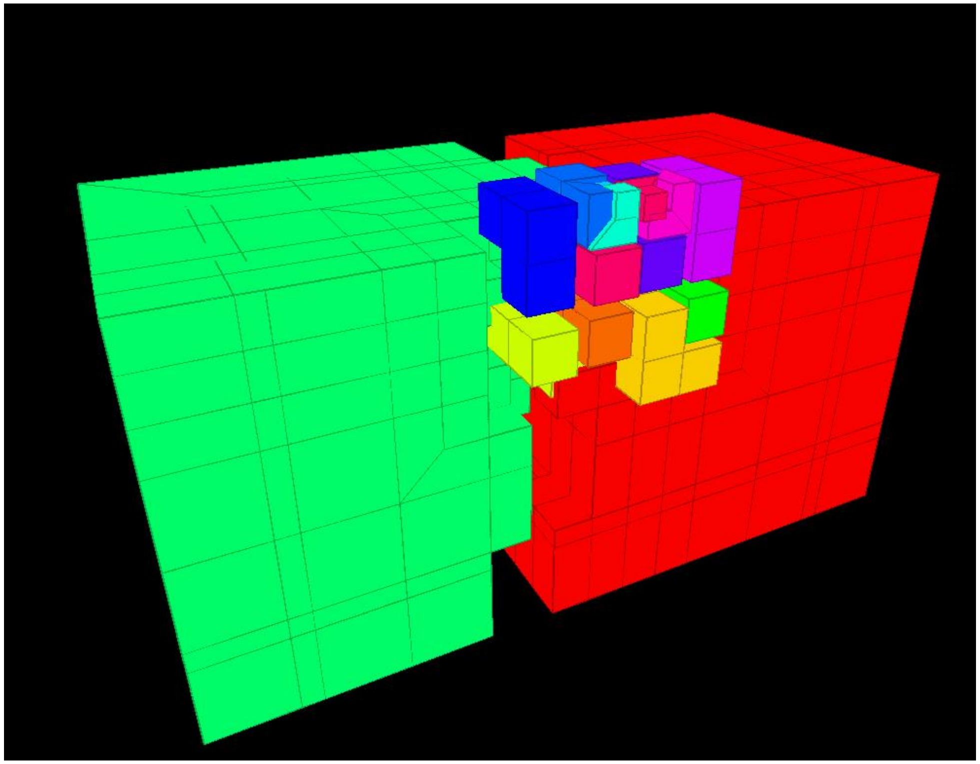
PDD: Mild Plasticity

- Small amount of plastic elements close to the pile
- Note smaller domain close to the pile
- Note also somewhat increased inter–domain boundary



PDD: Severe Plasticity

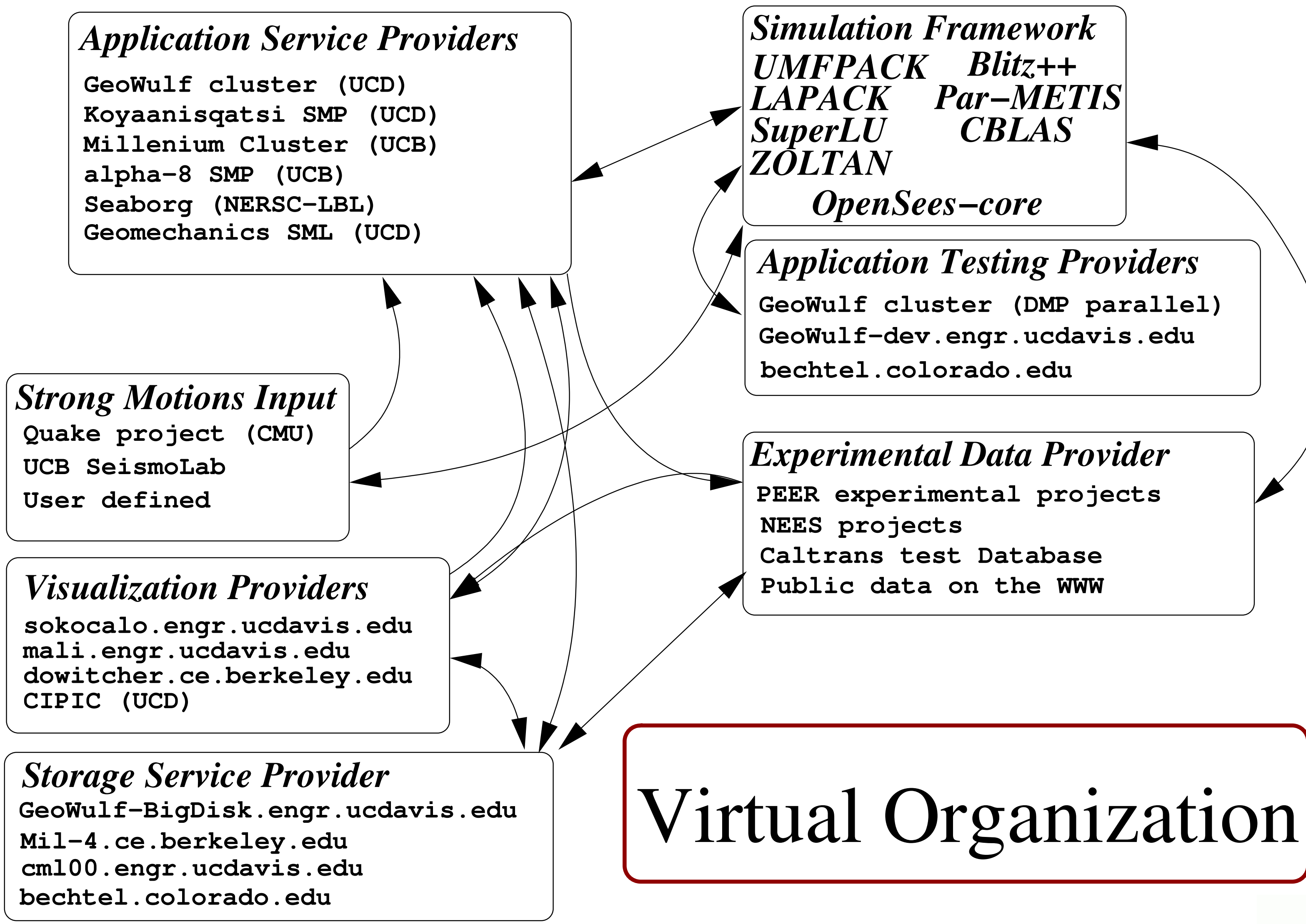
- Note small number of elements per CL heavy domains
- Note also two large, mostly elastic, subdomains
- Good DD for internal state determination, but bad for SES
- Use graph–partitioner to optimize SES phase using info on imbalance and network speed



Grid Services

- In addition to internal state determination and solving system of equations, other computational jobs can be distributed
- Data bases of
  - data bases of models
  - repositories of input motions
  - results storage providers
  - visualization (post–processing, rendering)
- A multi–objective (slack) optimization problem on larger scale
  - minimize network (inter–processor) communication
  - minimize data and job redistribution cost
  - balance computational work

Current Grid Resources



Conclusions