IBM Research

OptimalGrid: Middleware for Automatic Deployment of Distributed FEM Problems on an Internet-Based Computing Grid

Toby Lehman (toby@almaden.ibm.com)
Glenn Deen (glenn@almaden.ibm.com)
James Kaufman (kaufman@almaden.ibm.com)
John Thomas

December 3, 2003

Talk Outline

- Value Proposition
- FEM Characteristics
- OptimalGrid Design
- Solving a Connected Problem with OptimalGrid
- The Eden Model Example
- OptimalGrid Screen Shot
- Conclusion

OptimalGrid: Combining Grid and Autonomic

- Purpose: Create a self-healing grid for connected problems
- Original Problem Domain: Finite Element Modeling
- Value Proposition:

  (1) Less Pain: No more . . .

    (1) Manual partitioning of parallel problems

    (2) Deploying problem pieces over heterogeneous machines

    (3) Managing and monitoring the distributed runtime job

  (2) More Pleasure:

    (1) Use “free” or “available” cycles in one’s organization

    (2) Concentrate on the problem algorithm, not its distribution

    (3) Run much MUCH larger programs given the increased resources

    (4) Bring high-performance computing to the average man.

Objectives: to automate the parallel process

Automatically …

(1) Partition the problem into parallelizable pieces
- User provides problem description
- User provides "atomic cell" methods

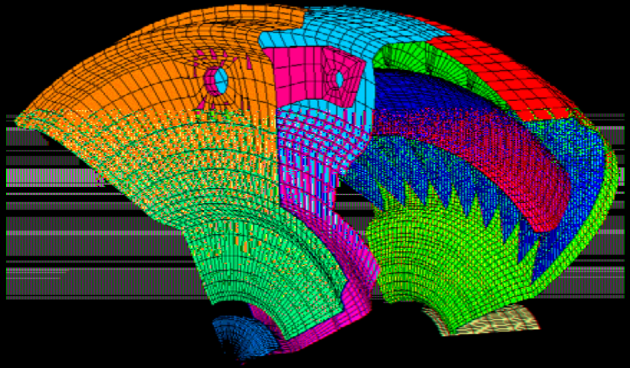
(2) Deploy the pieces to available machine, assign roles

(3) Manage, monitor and orchestrate the running problem

(4) Augment with utilities (grid services) for aggregation, visualization and report

Finite Element Modeling

torque converter



Broad Class of Problems

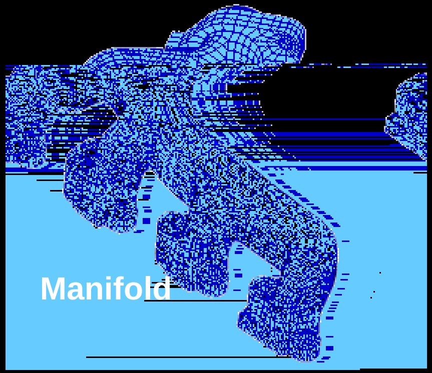
Anything involving flow!!

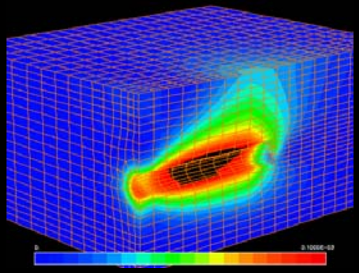
- Heat
- Electron tunneling
- Stress
- Process flow
- Oil refinery
- Finance

Highly Parallelizable

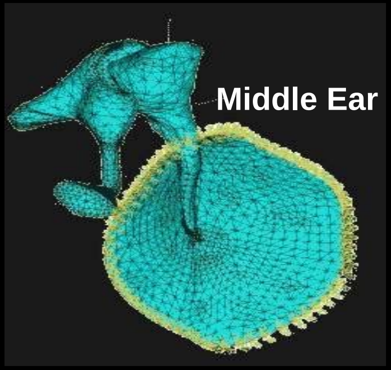
Abstraction to more general class of computing problems.

Refinery process



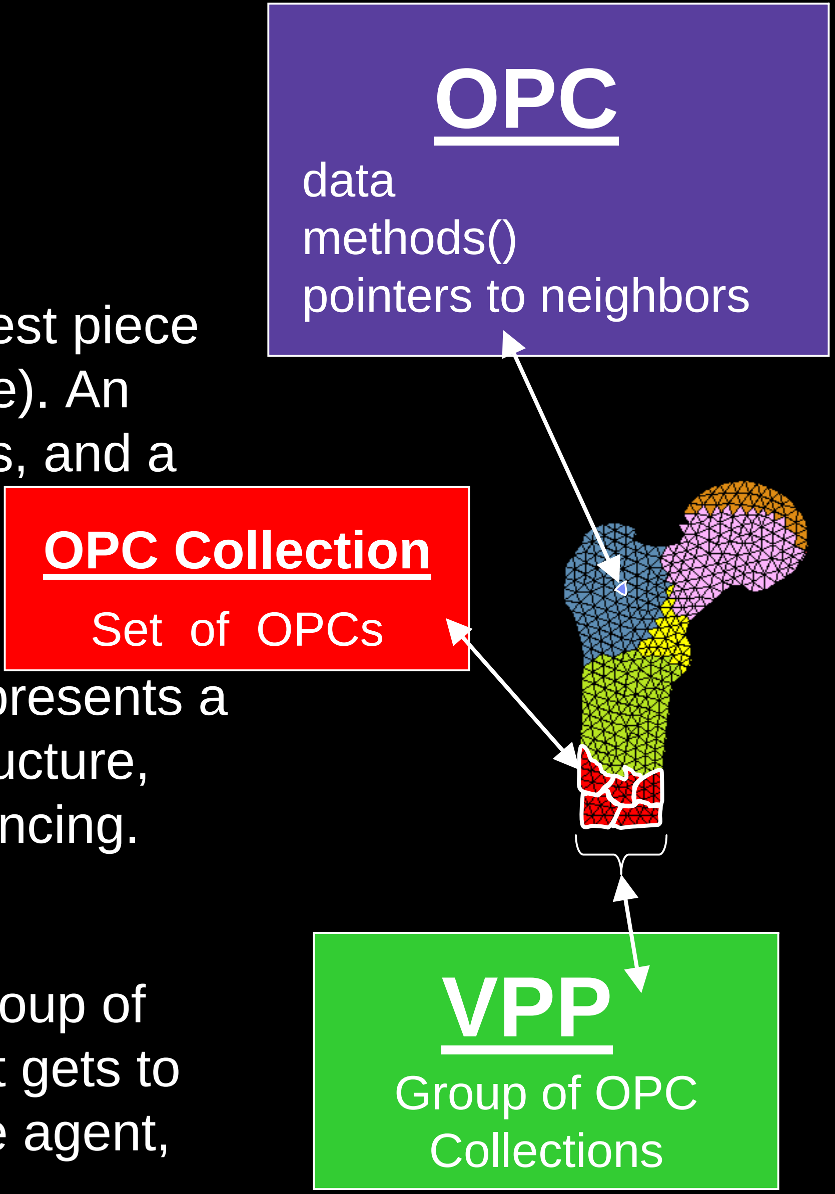


Electron tunneling

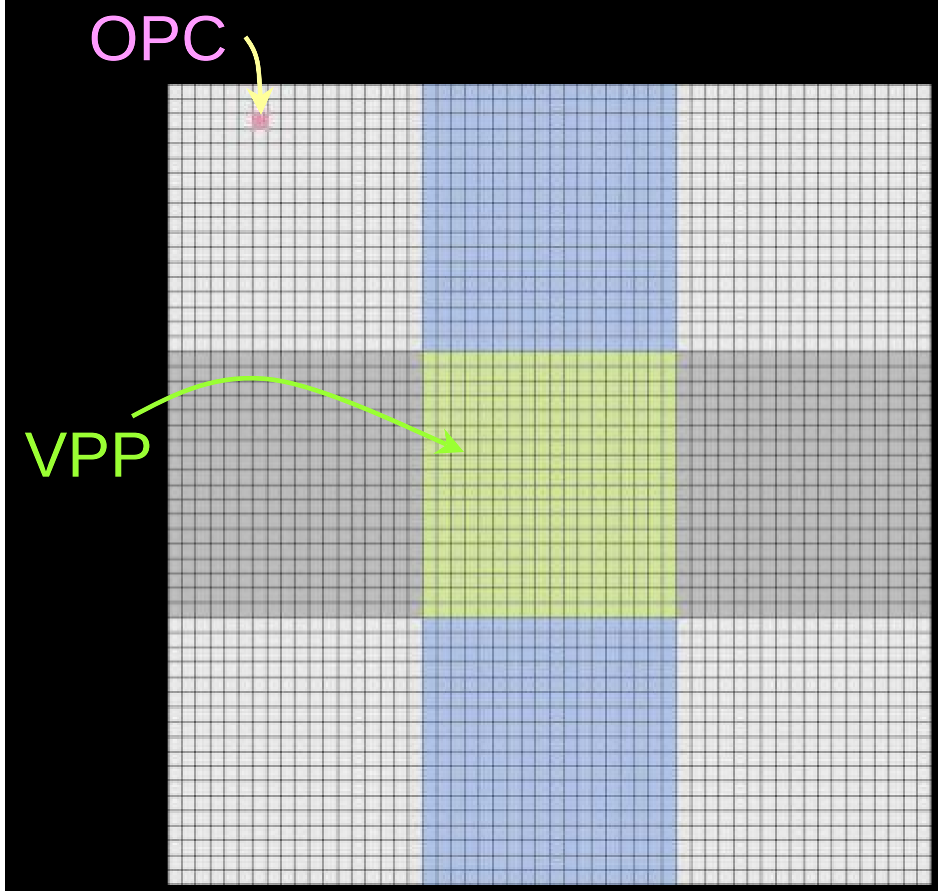


Terminology

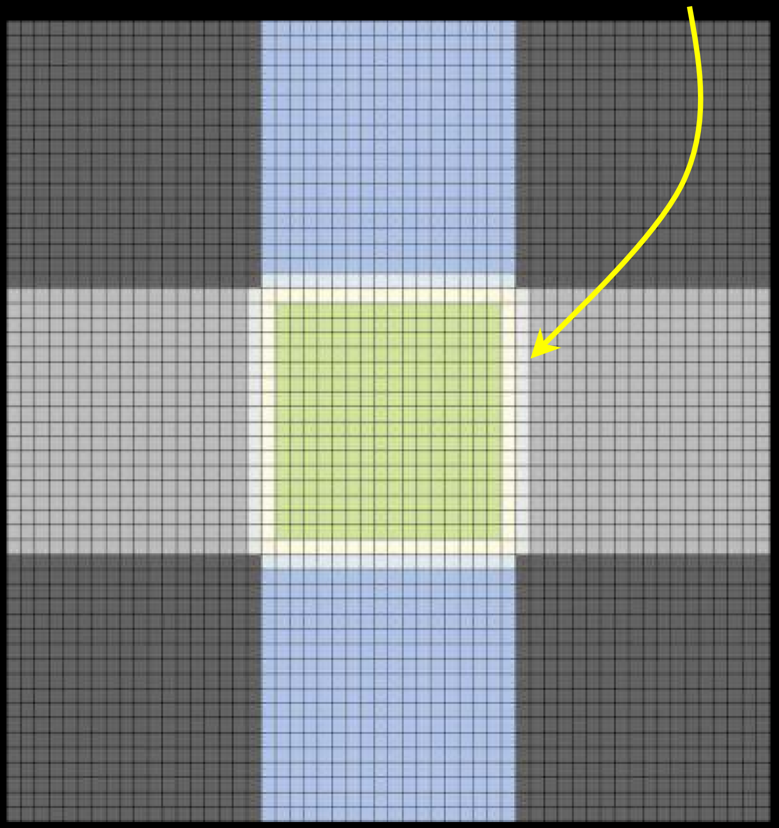
- Original Problem Cell (OPC): The smallest piece of a problem (the atomic, indivisible piece). An OPC contains state data, iterate methods, and a list of neighboring OPCs.
- OPC Collections: A set of OPCs that represents a fixed part of the problem. An internal structure, used for problem segmenting and rebalancing.
- Variable Problem Partitions (VPPs): A group of OPC Collections that one compute agent gets to process. The more powerful the compute agent, the larger its VPP (more collections in it).



The OptimalGrid View of a Problem

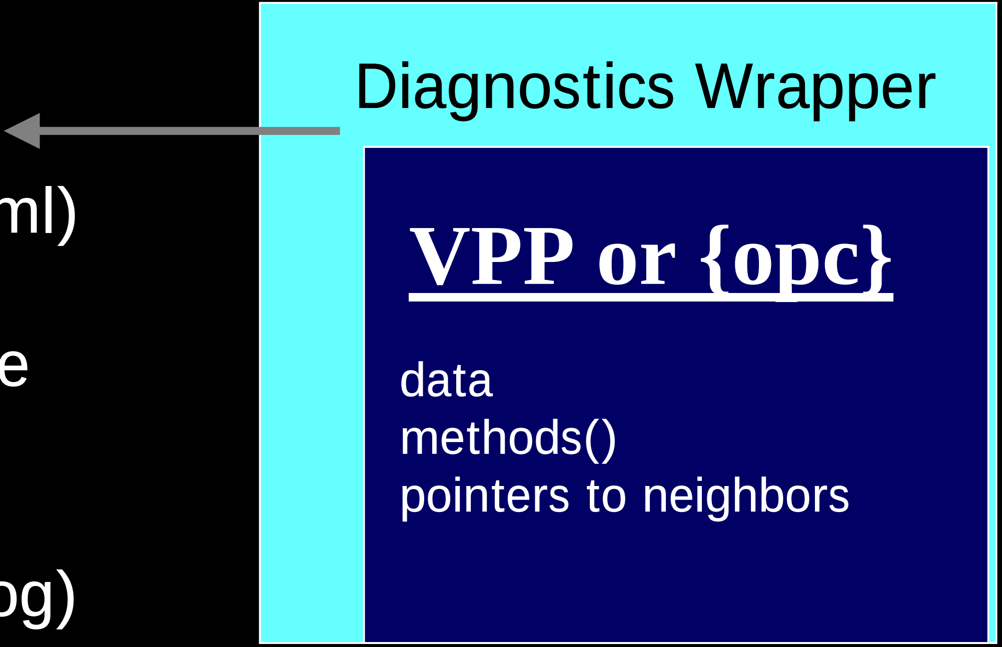


The VPPs communicate with each other through vectors of “remote” OPCs at the edges.



Original Problem Cell Collections (VPPs) and The Diagnostics Wrapper

1. CA Self Test
2. CA Self Description (xml)
3. VPP Complexity
4. Estimated Performance
   i. Computation
   ii. Communication
5. Actual Performance (log)
   i. Computation
   ii. Communication
   iii. As a function of Problem Complexity!



Whiteboards (TSpaces) in the OptimalGrid Runtime

- Manages the “edge” communication
- Service Registration (all components register as services)
- Service/Node management (all components get direction via TSpaces)

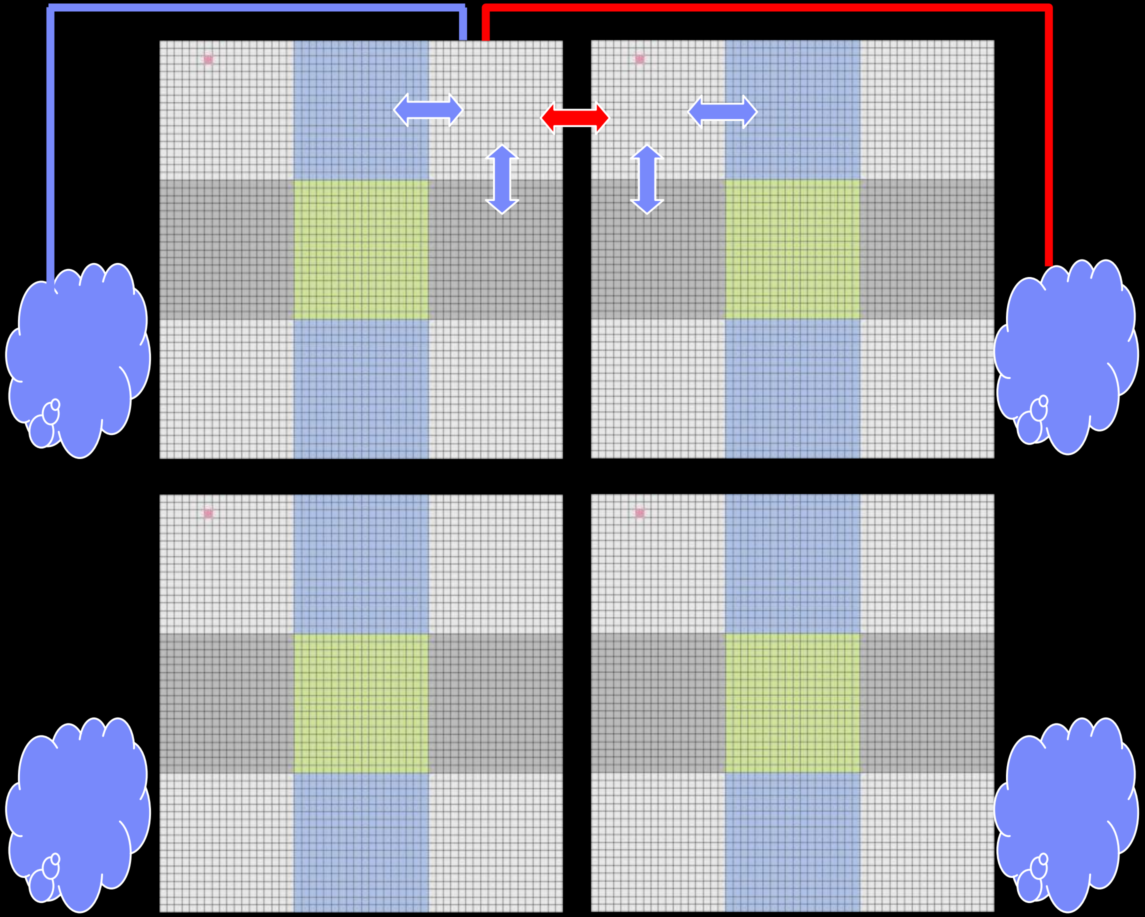
Communication Options:

(1) Peer-to-peer

+ Faster
+ Lighter weight
+ More Choices
- No recovery
- No monitoring
- No synchronization

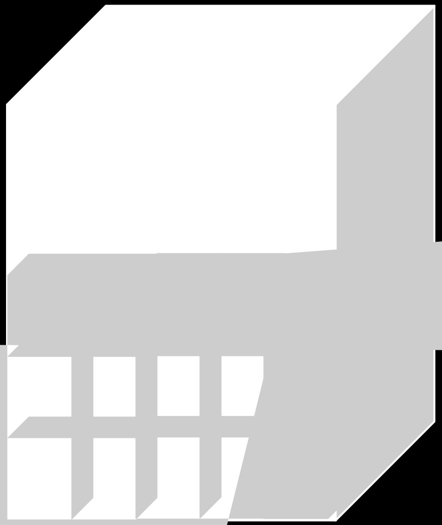
(2) Whiteboard (TSpaces)

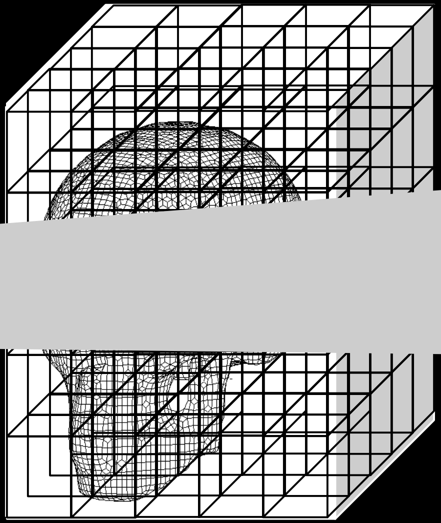
+ Monitoring
+ Recovery
+ Aux services/utilities
- Speed



(1) Partition a Parallel Problem

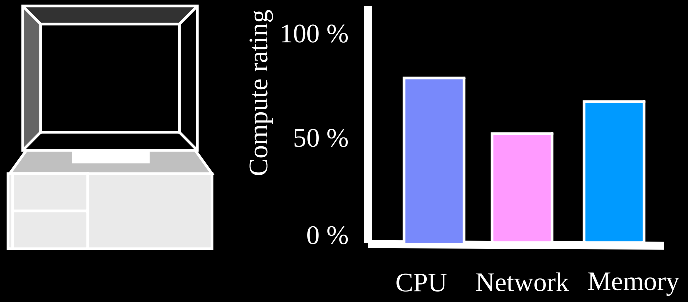
- Use Spatial filter to automatically subdivide into “collections”
  - (a)Take a shape, apply the bounding box (volume)
  - (b) Segment the space into collections
  - (c) Pass the problem thru the filter to create partitioned data subspaces

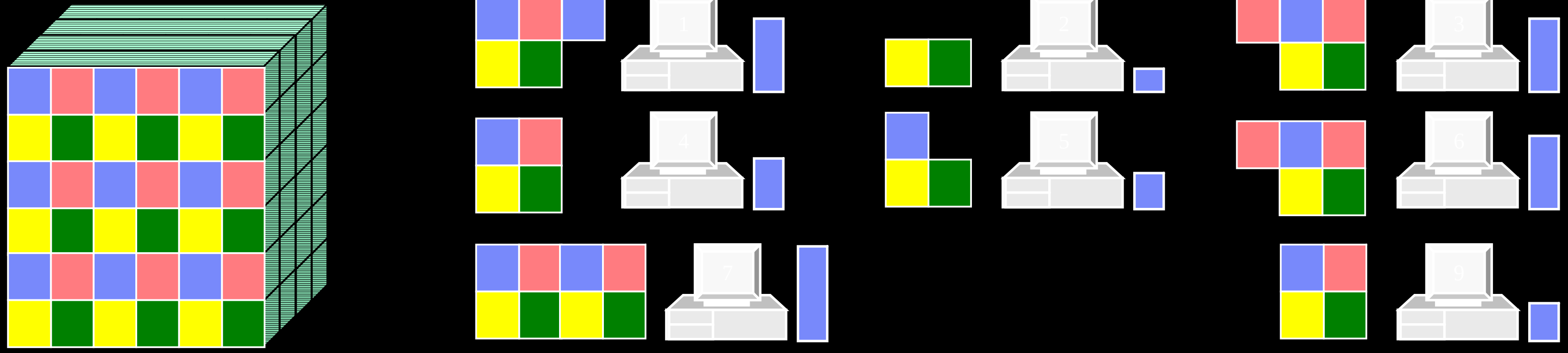




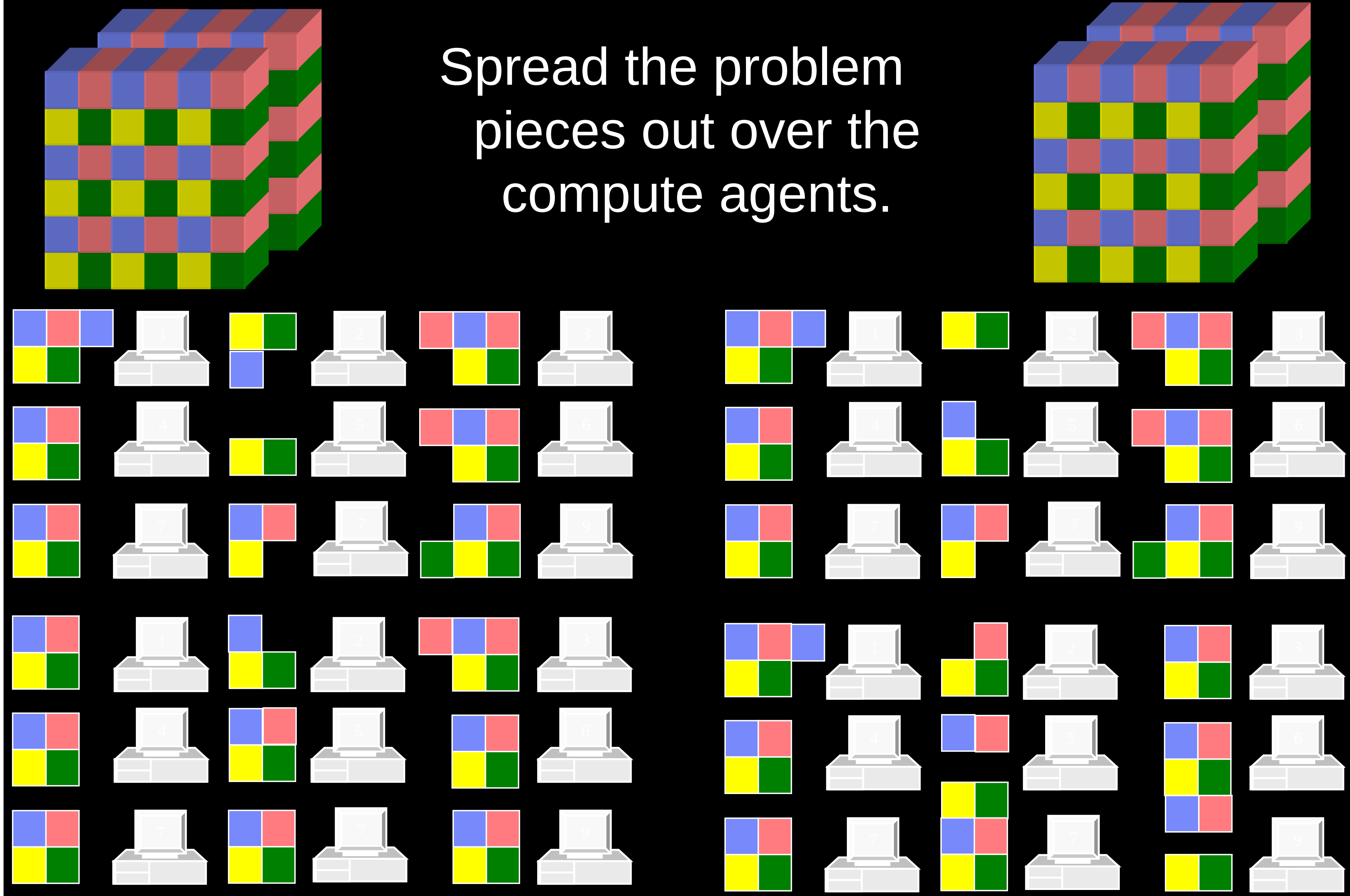
(1) Partition a Parallel Problem (continued)

- Assign OPC Collections to different VPPs (1 VPP per agent)
  - Base decision on agent capabilities and capacities





Spread the problem pieces out over the compute agents.



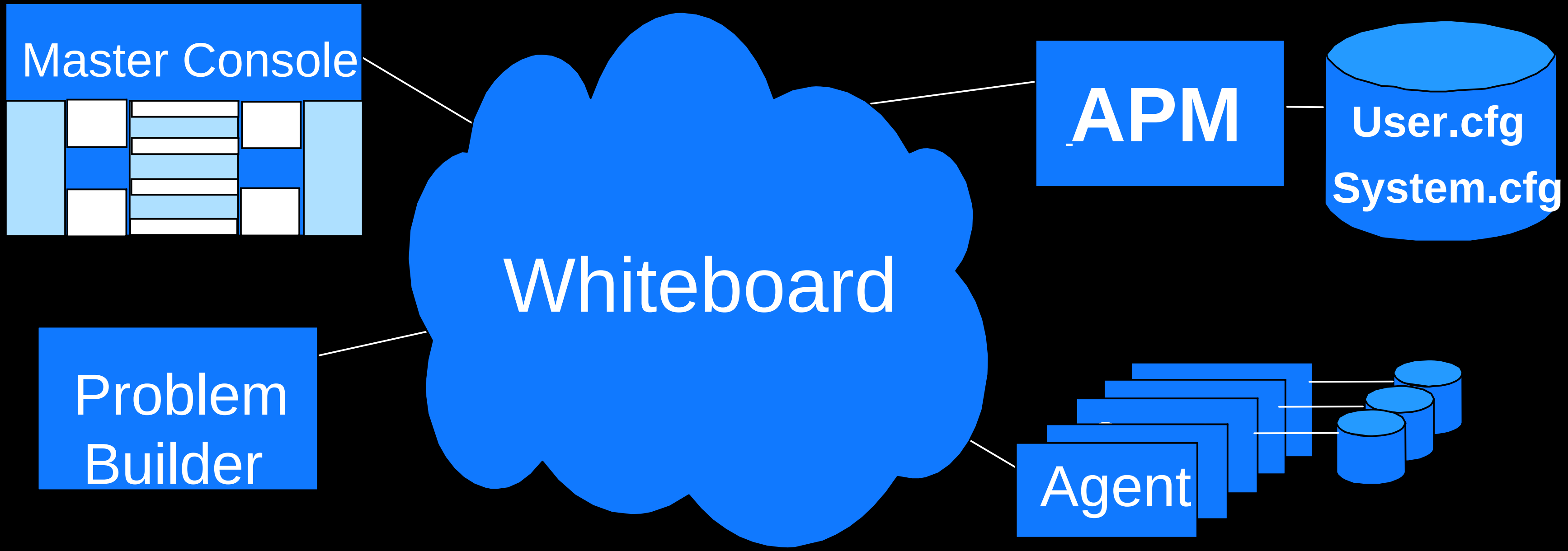
, as n roles

exec plan;

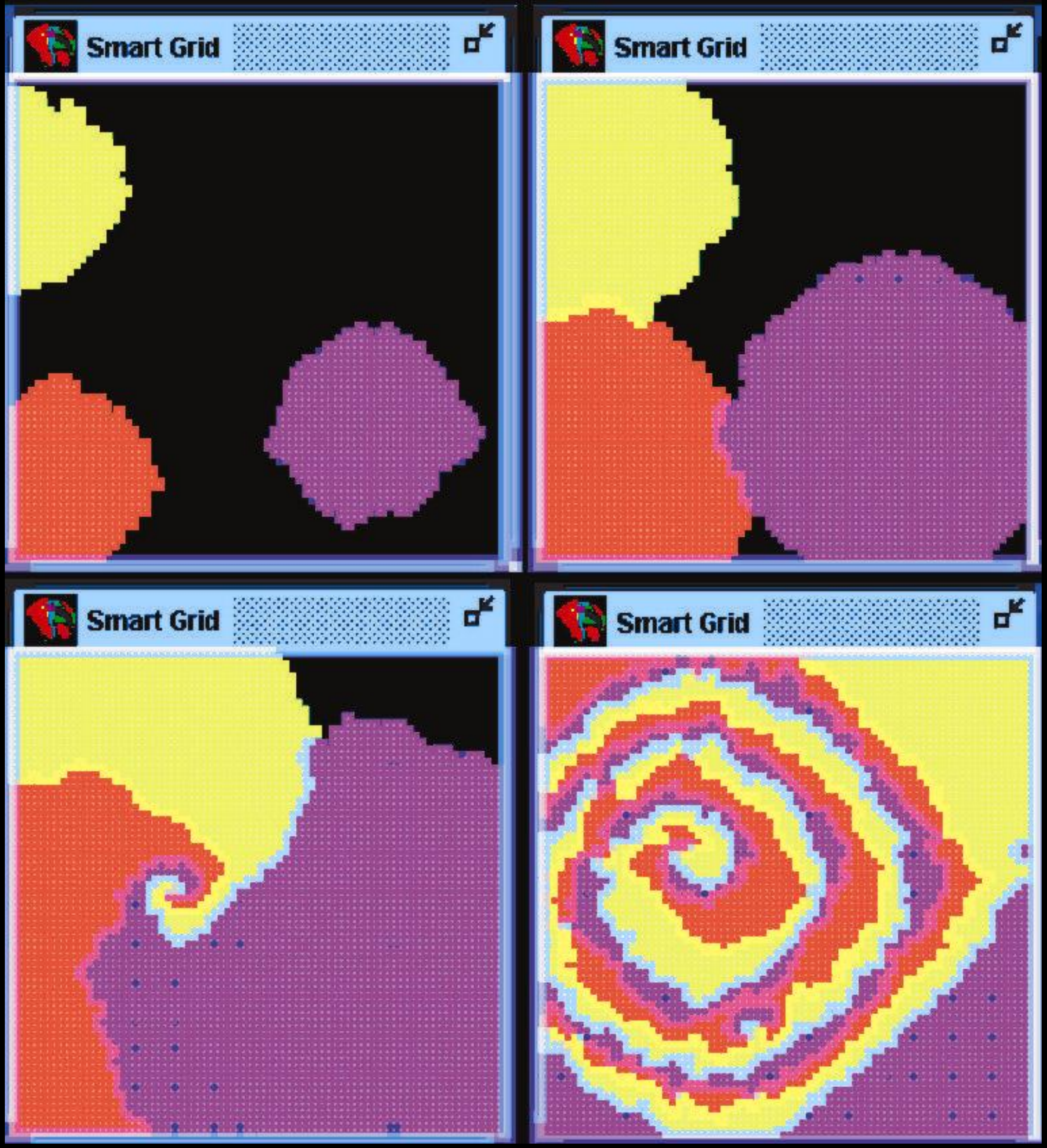
ux s s.

(3) Manage, monitor and orchestrate the running problem

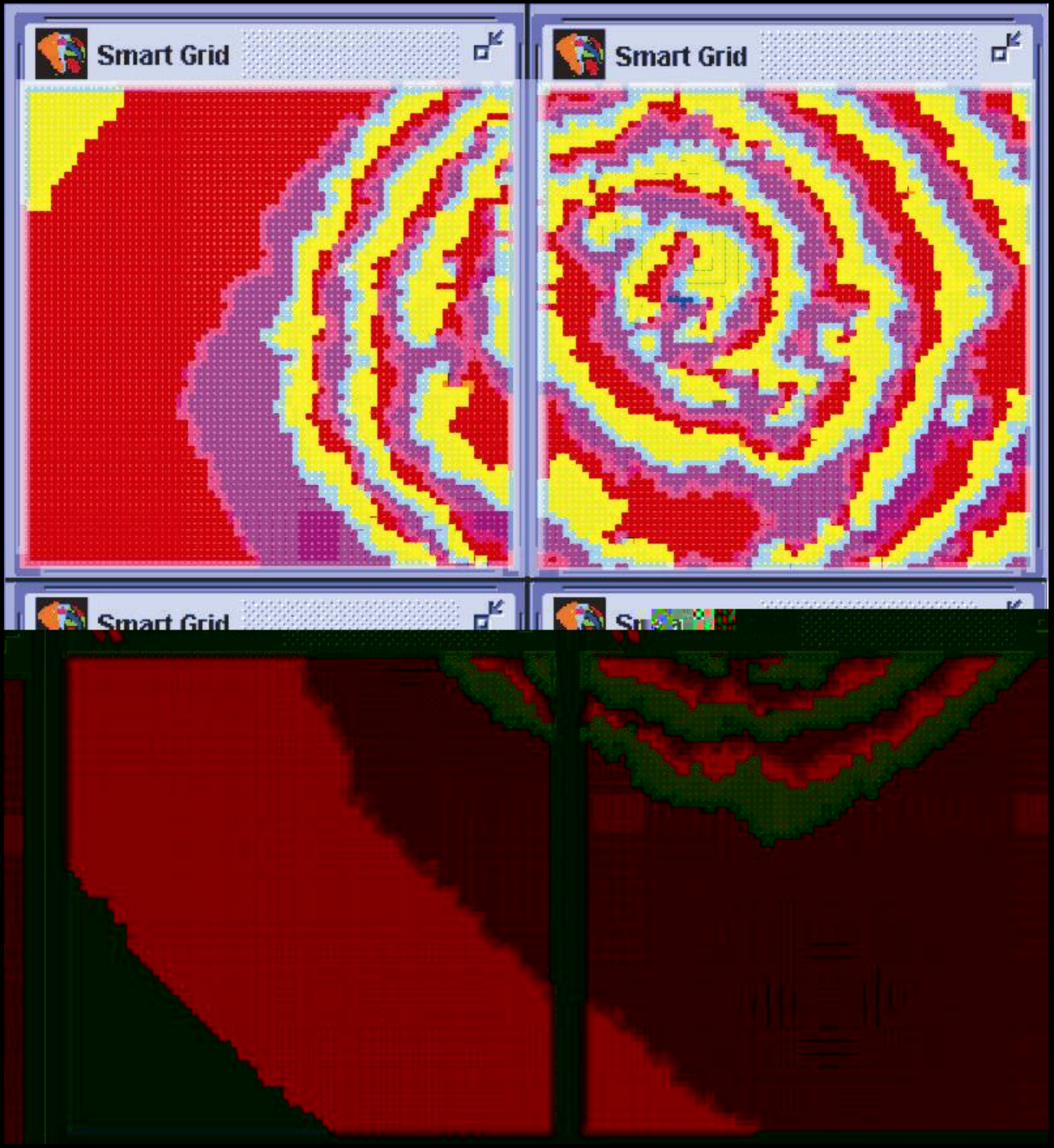
- Problem builder sets up problem
- User monitors/manipulates via master console
- APM monitors, manages and orchestrates problem via whiteboard
- Agents run the various specified services
- APM watches performance, rebalances when necessary



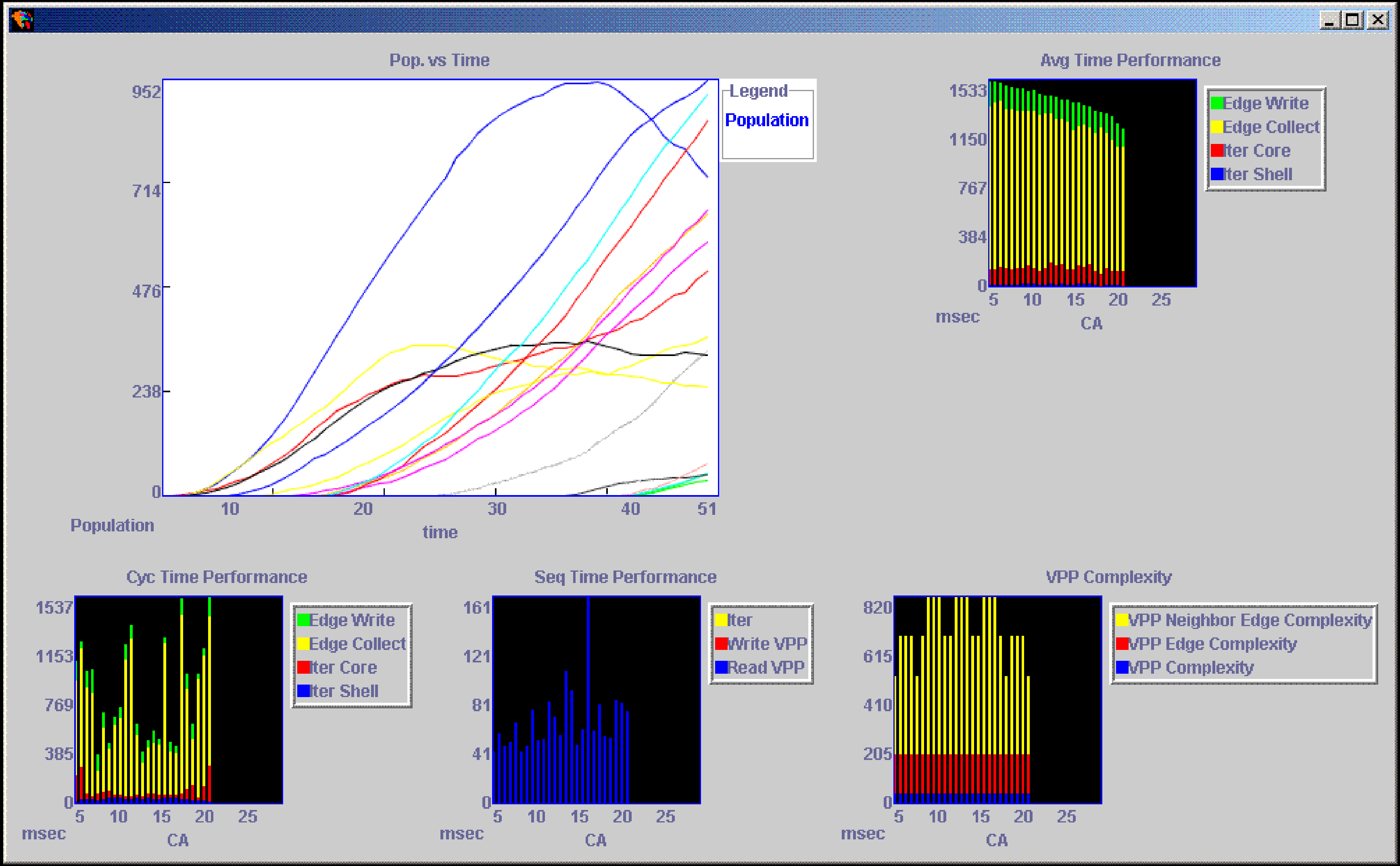
- Simple “Eden Model” simulation
- One machine, four time slices
- Predator/Prey
  - A eats B
  - B eats C
  - C eats A



- Same Eden Model
- Problem four times larger
- Four machines, one time slice.
- Notice the edge communication.



SmartGrid A.P.M.



Conclusion

- Solve Large Connected problems (e.g. FEM problems) on the Internet grid.
- Performance vs flexibility and portability: Just like the “Java vs C/C++” arguments, OG makes it easier to get up and run on the grid, but probably not faster than a hand-written system, tuned to the hardware.
- New Style of working (plug-in vs main). OG “owns” the system, user just supplies the plug-in algorithm and the data
- Brings the new age of On-Demand Computing to the forefront with a grid infrastructure that dynamically adapts to application, compute and resource load.
- http://www.almaden.ibm.com/software/ds/OptimalGrid