vgES Demonstrations

Andrew A. Chien, Henri Casanova, Fran Berman

Yang-Suk Kee, Kenneth Yocum

Richard Huang, Dionysis Logothetis, and Jerry Chou

CSE, SDSC, and CNS
University of California, San Diego

VGrADS Site Visit

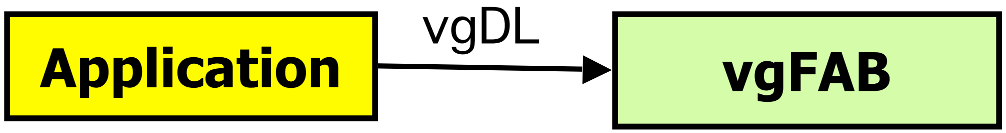
April 28, 2005

VGrADS
Virtual Grid Application Development Software Project

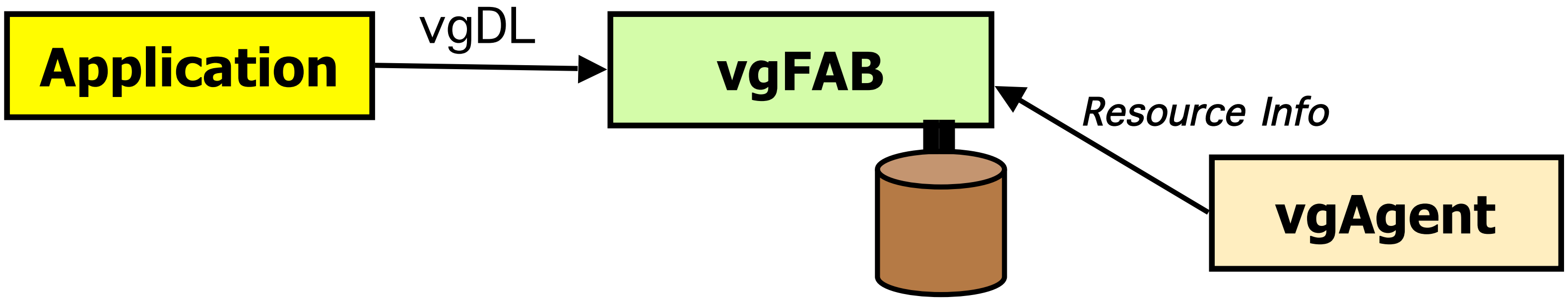
vgES: Prototype Research Infrastructure

- We have developed a functional prototype of vgES
  - vgES 0.7, March 2005
- Two demonstrations:
  - vgFAB: Finding and Selecting Resources
  - vgES: Full application run on the VGrADS testbed

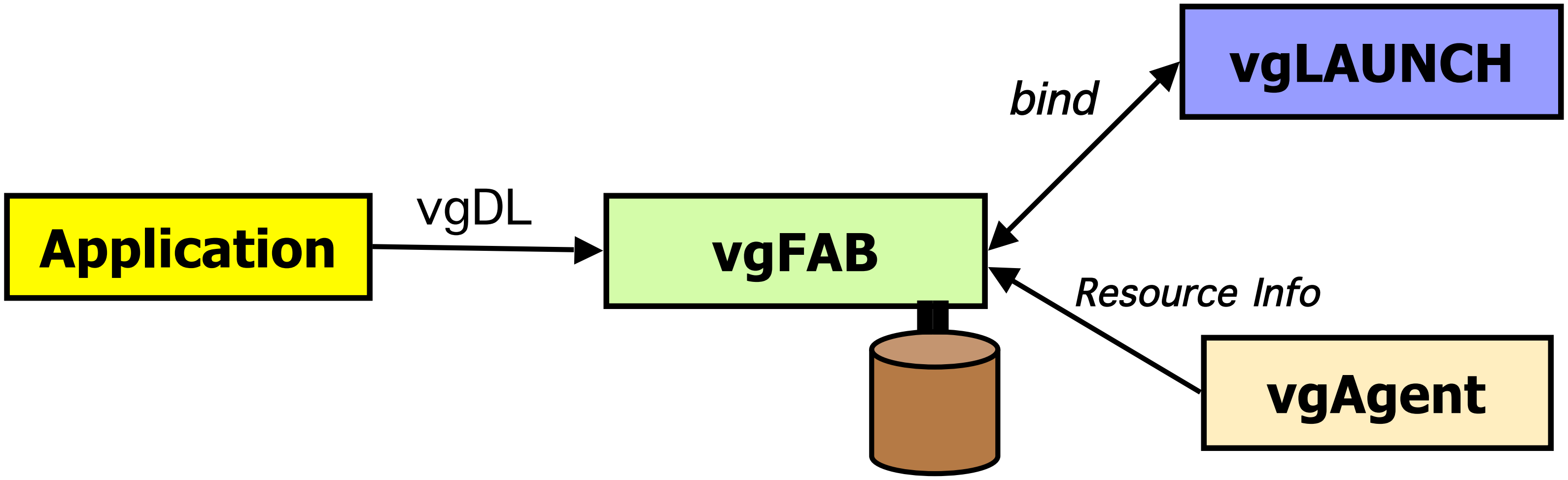
The vgES Research Prototype



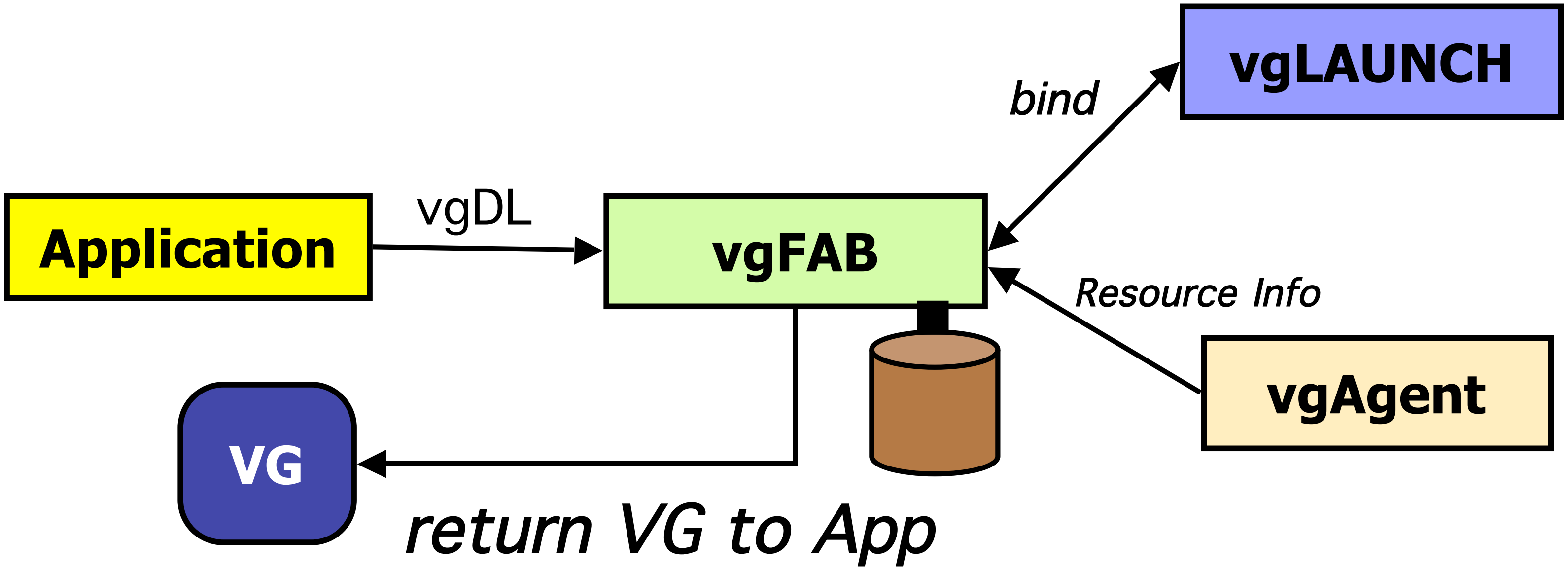
The vgES Research Prototype



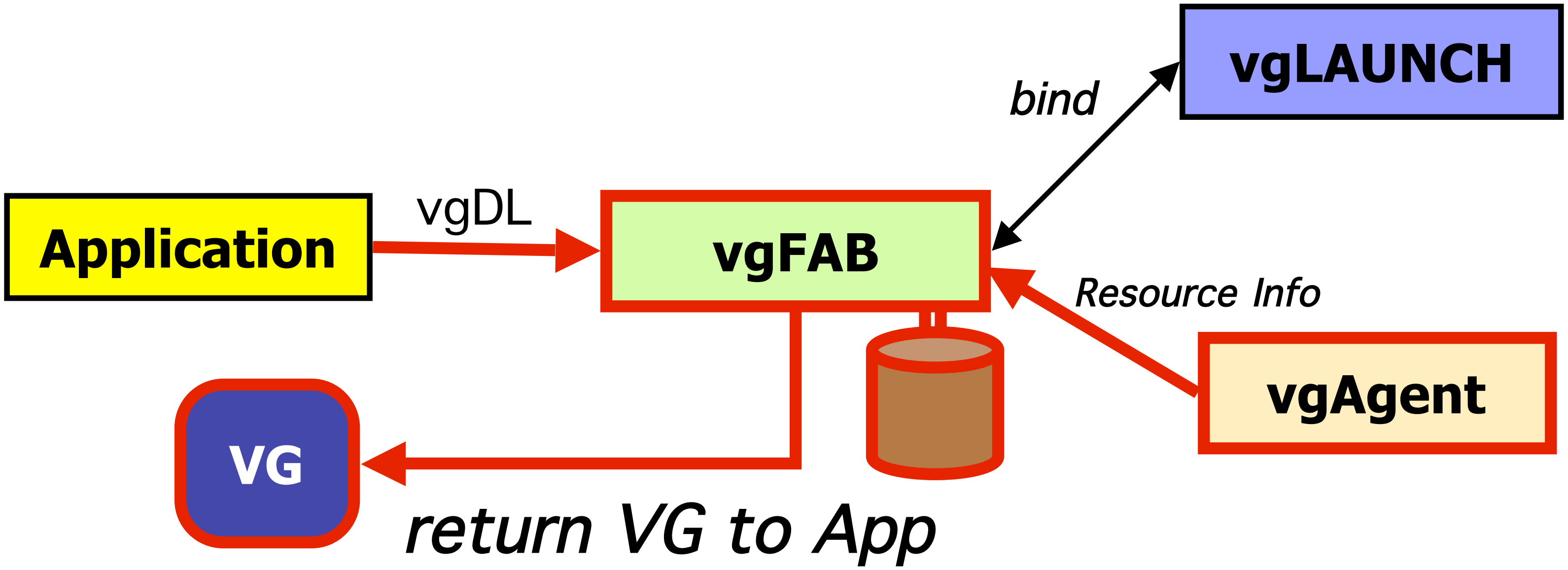
The vgES Research Prototype



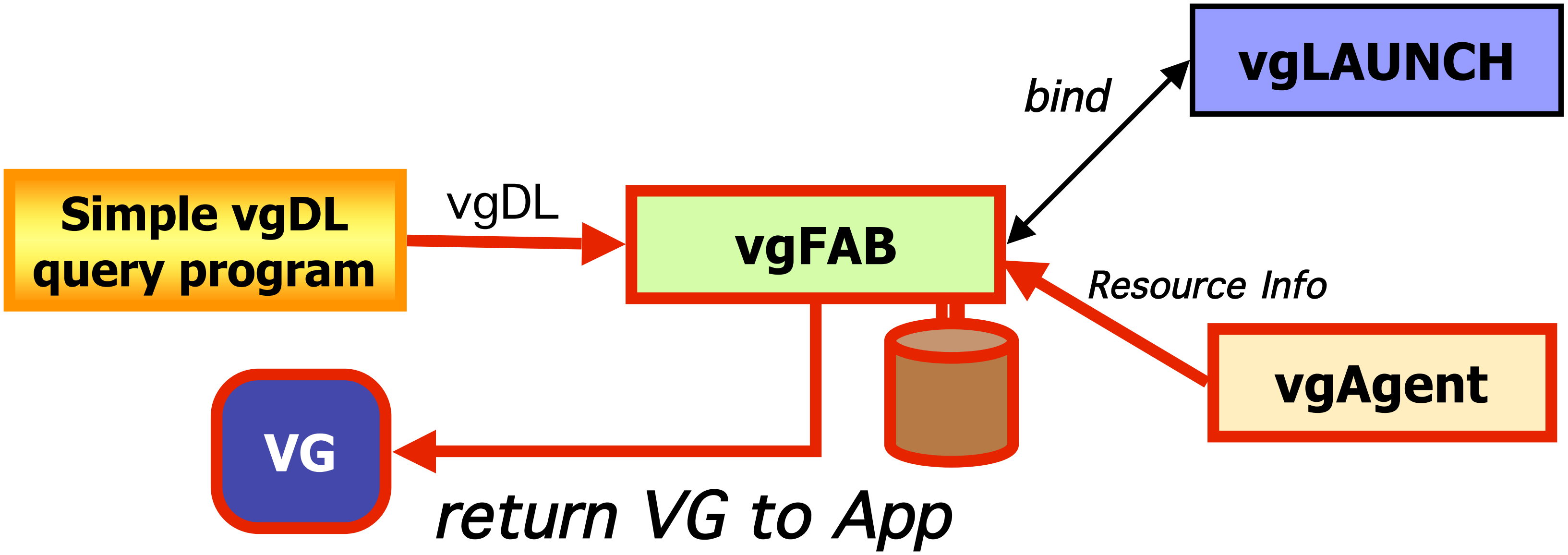
The vgES Research Prototype



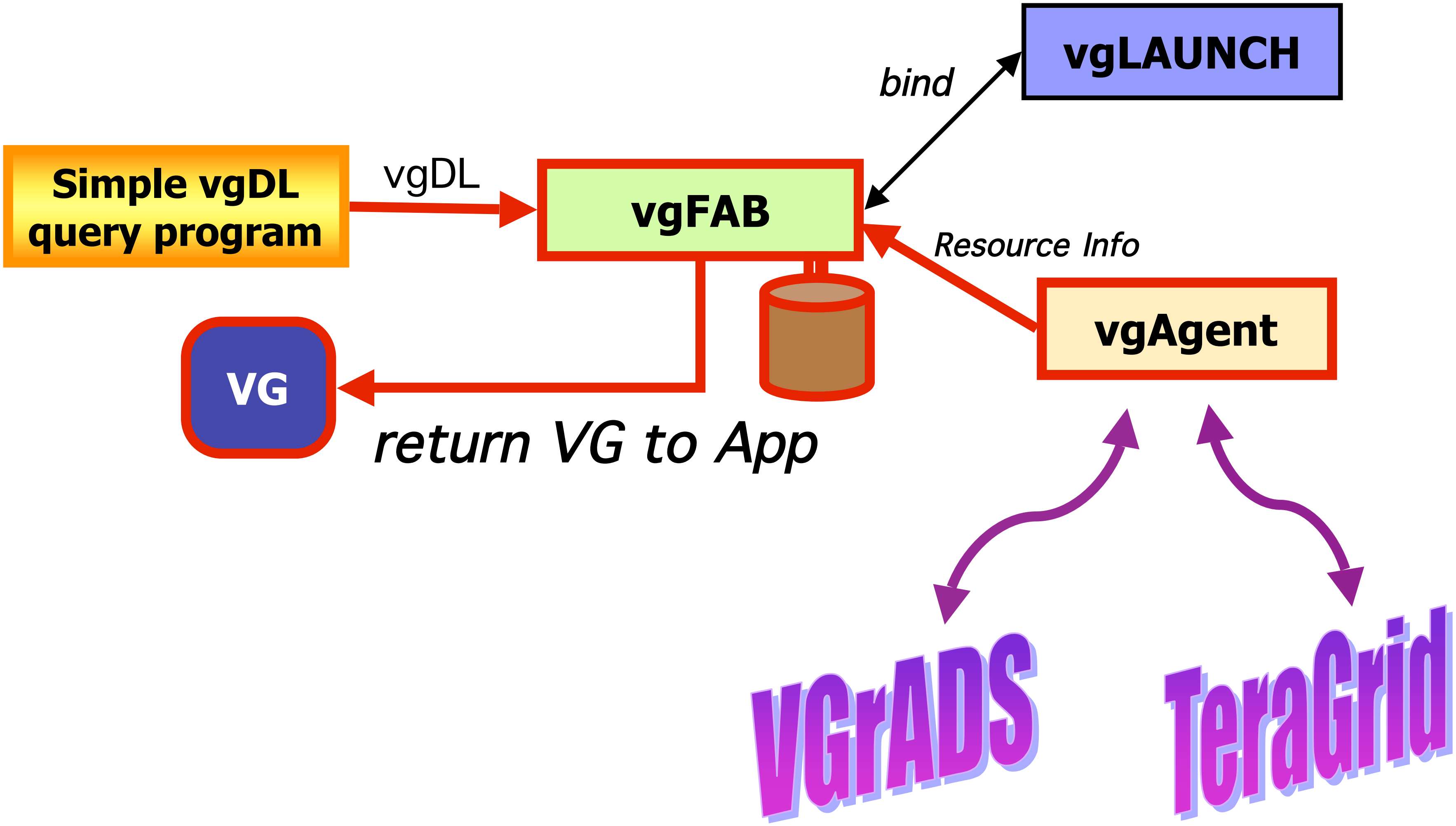
Demonstration #1: vgFAB



Demonstration #1: vgFAB



Demonstration #1: vgFAB



Simple vgDL query program

vgDL

vgFAB

bind

vgLAUNCH

Resource Info

vgAgent

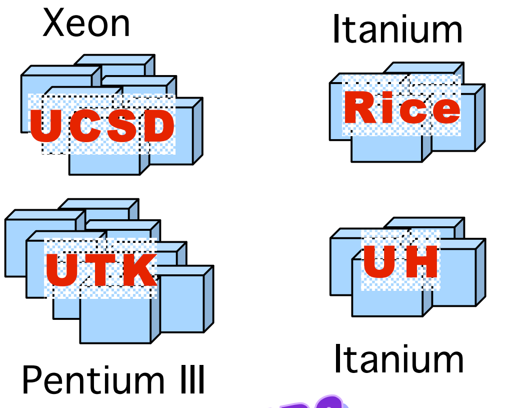
VG

return VG to App

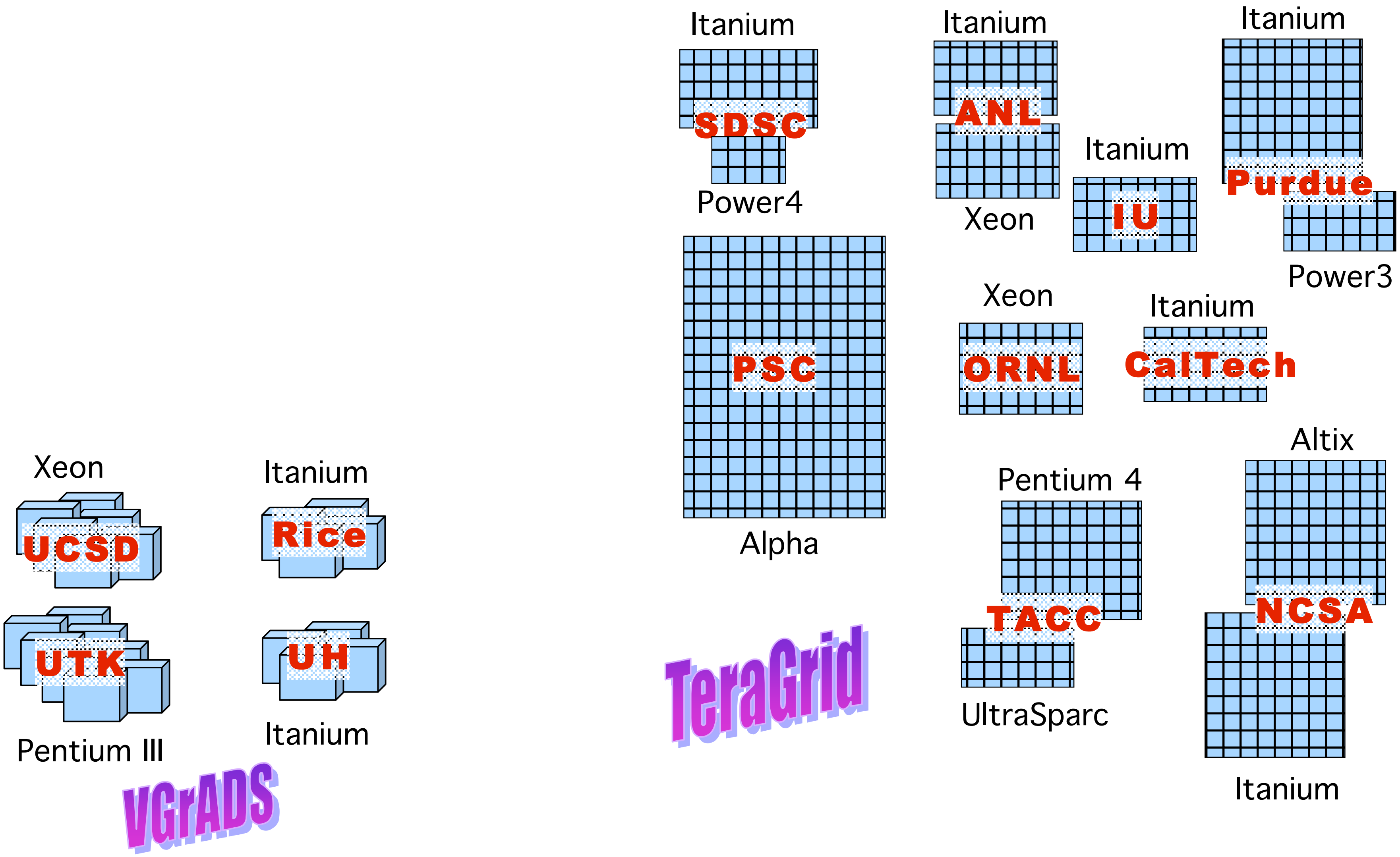
VGrADS

TeraGrid

VGrADS and TeraGrid Resources

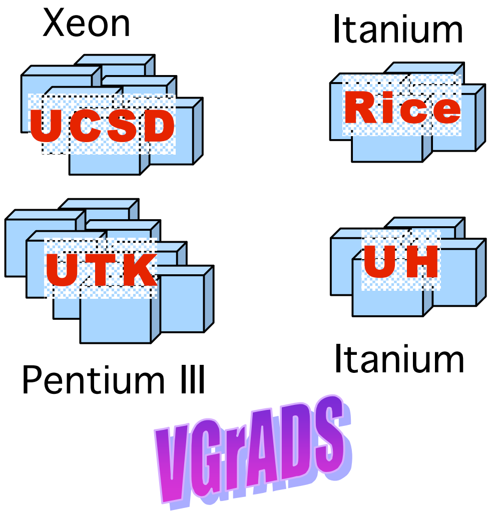


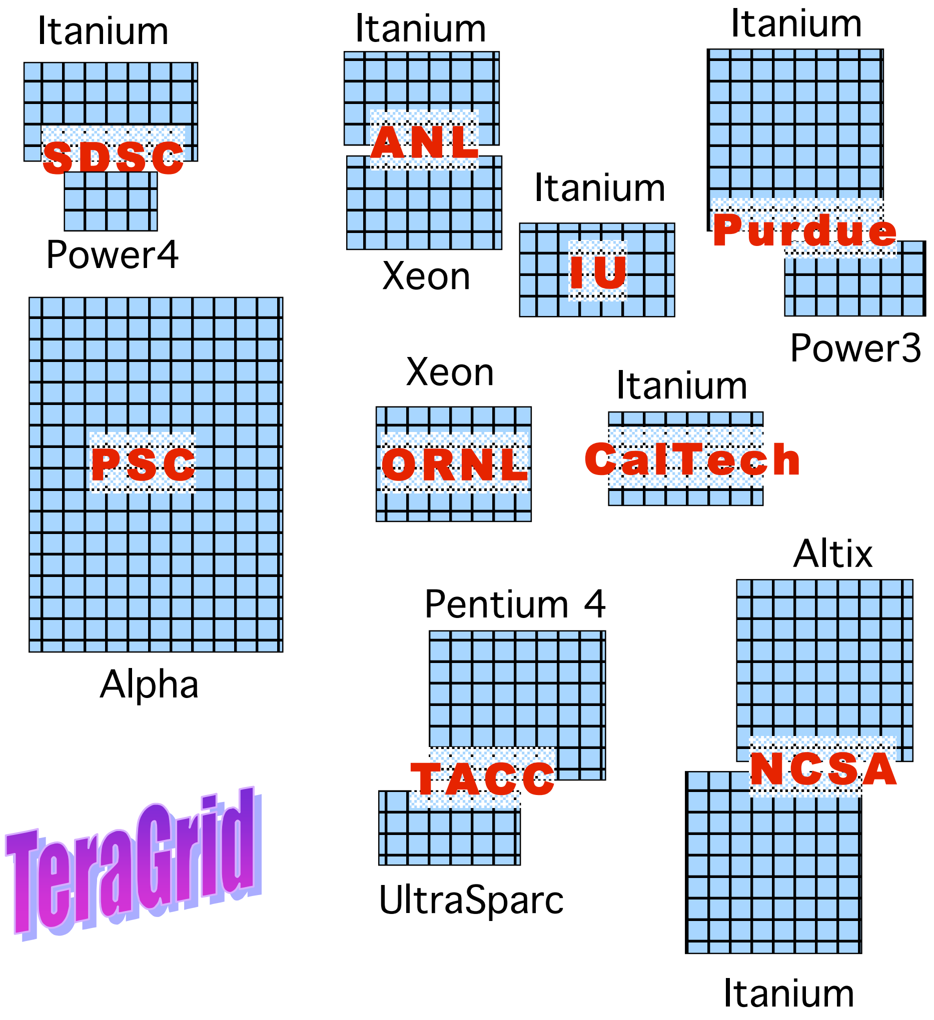
VGrADS and TeraGrid Resources



A simple vgDL query

```
VG1 = ClusterOf(node) [4:64]
   {
       node =[
           (Processor == Xeon) &&
           (Clock >= 1000) &&
           (Memory >= 1000)
       ]
   }
```

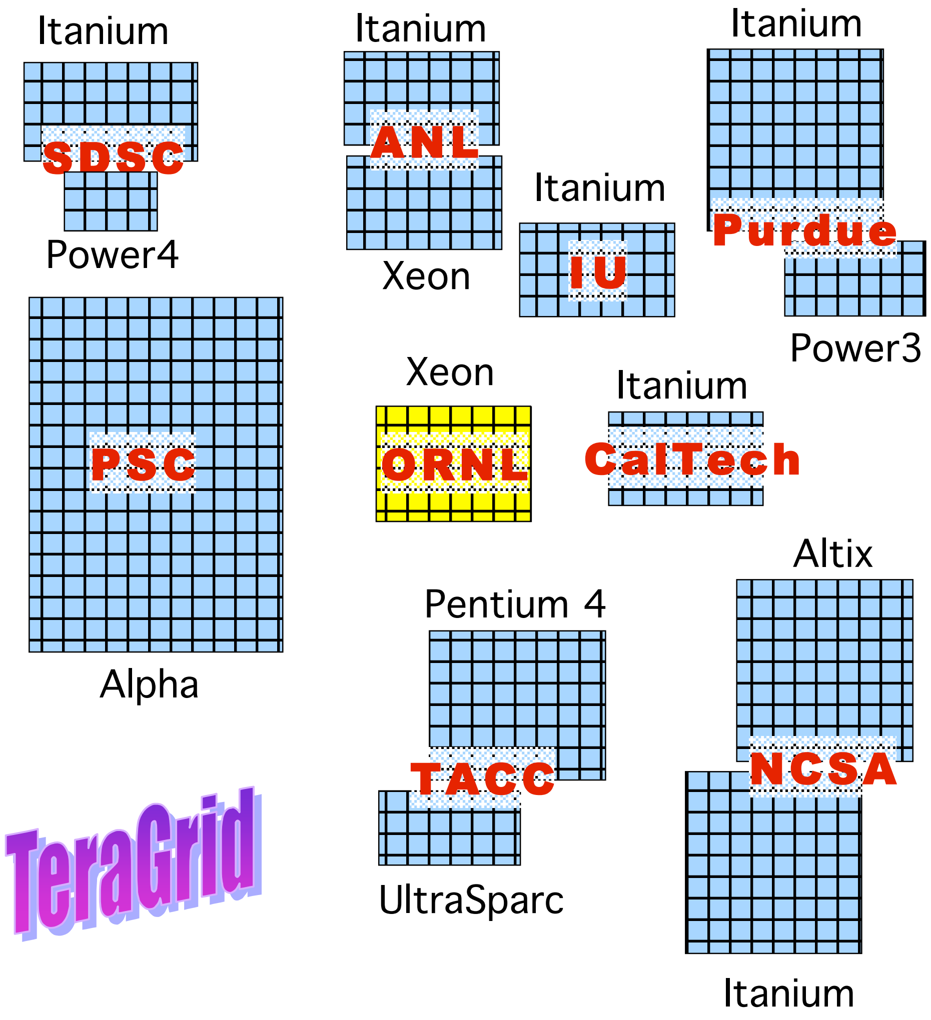


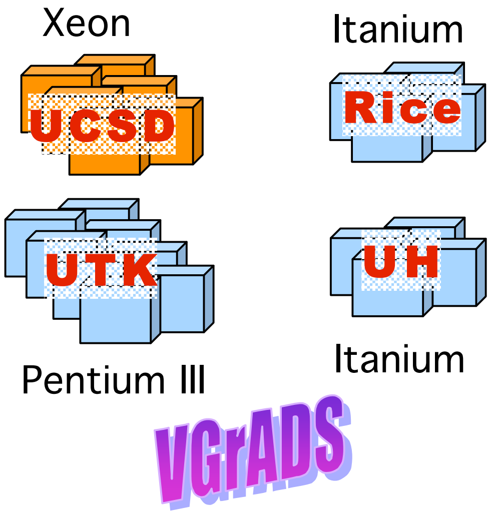


Switch to live demo

A simple vgDL query

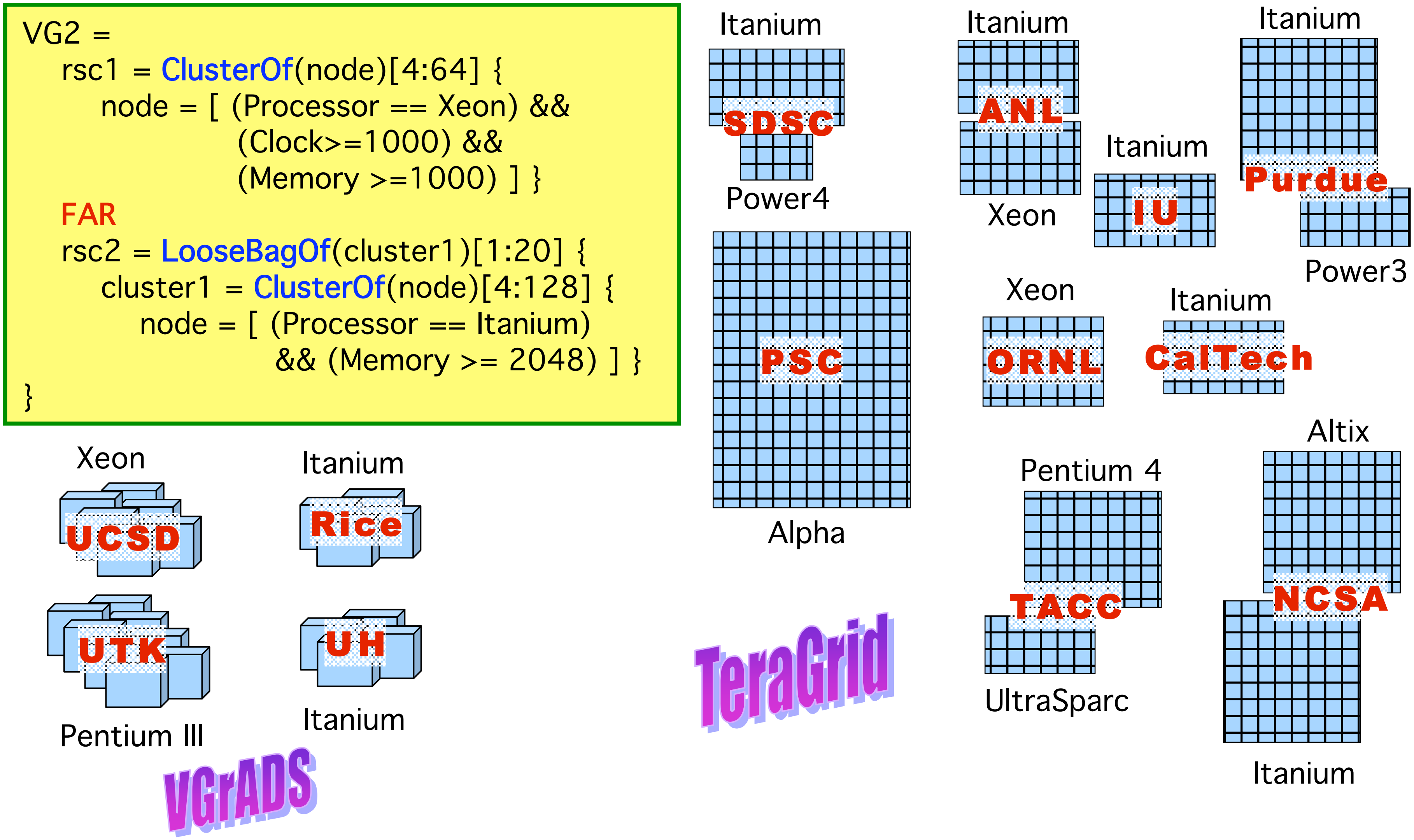
```
VG1 = ClusterOf(node) [4:64]
   {
       node =[
           (Processor == Xeon) &&
           (Clock >= 1000) &&
           (Memory >= 1000)
       ]
   }
```





A more complex vgDL query

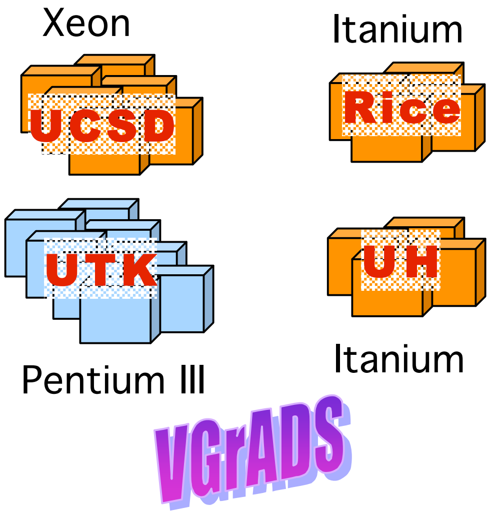
```
VG2 =
    rsc1 = ClusterOf(node)[4:64] {
        node = [ (Processor == Xeon) &&
                     (Clock>=1000) &&
                     (Memory >=1000) ] }
    FAR
    rsc2 = LooseBagOf(cluster1)[1:20] {
        cluster1 = ClusterOf(node)[4:128] {
            node = [ (Processor == Itanium)
                         && (Memory >= 2048) ] }
}
```

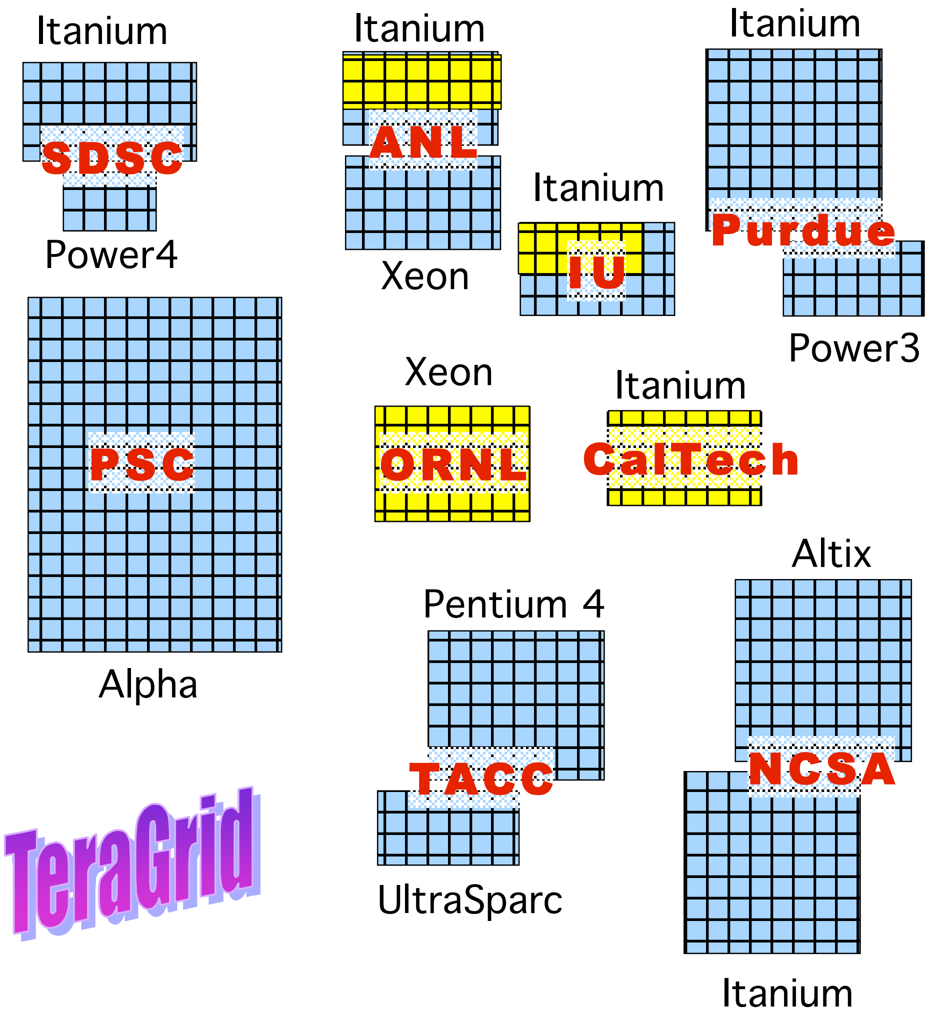


Switch to live demo

A more complex vgDL query

```
VG2 =
    rsc1 = ClusterOf(node)[4:64] {
        node = [ (Processor == Xeon) &&
                     (Clock>=1000) &&
                     (Memory >=1000) ] }
    FAR
    rsc2 = LooseBagOf(cluster1)[1:20] {
        cluster1 = ClusterOf(node)[4:128] {
            node = [ (Processor == Itanium)
                     && (Memory >= 2048) ] }
}
```





Synthetic Resource Environments

- **Resource selection is a difficult problem**
  - It's NP-hard
  - We have a heuristic and a prototype implementation
- **Research: result quality, scalability, response time, contention, ...**
  - What is needed: Simulation studies in large and realistic environments
- **We have developed a "resource environment generator"**
  - Based on survey of existing systems and analysis of technology trends
  - Kee, Casanova,Chien, Realistic Modeling and Synthesis of Resources for Computational Grids, SC2004.
- **A sample synthetic environment**
  - 1 million hosts, 10,000 clusters, Pentium 2-4, Itanium, Opteron, Athlon.
  - VG3 = rsc1= **ClusterOf** (node) [4:64] { node = [(Processor==Pentium4) && (Clock>=2000) && (Memory>=4096)] } **FAR** rsc2 = **LooseBagOf** (nest_cluster) [1:20] { nest_cluster = **ClusterOf** (node) [4:128] { node = [(Processor==Itanium) && (Memory>=8192)] } }

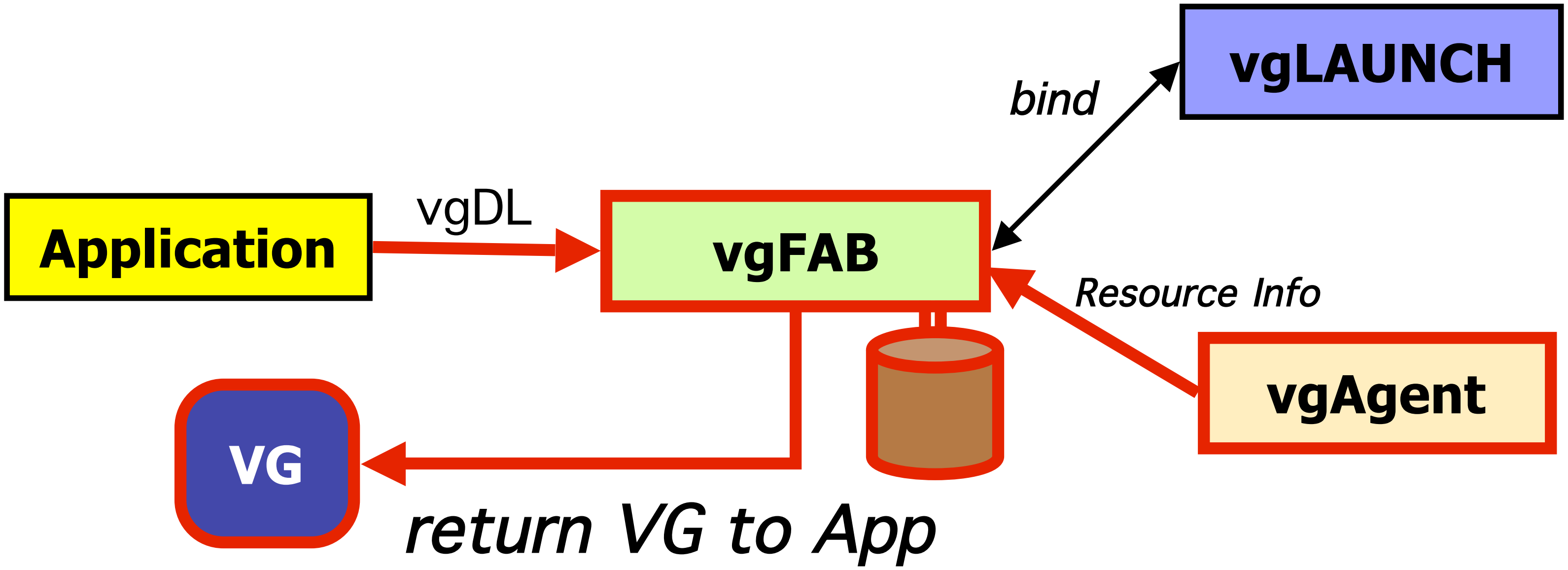
Switch to live demo

Take-away from Demonstration #1

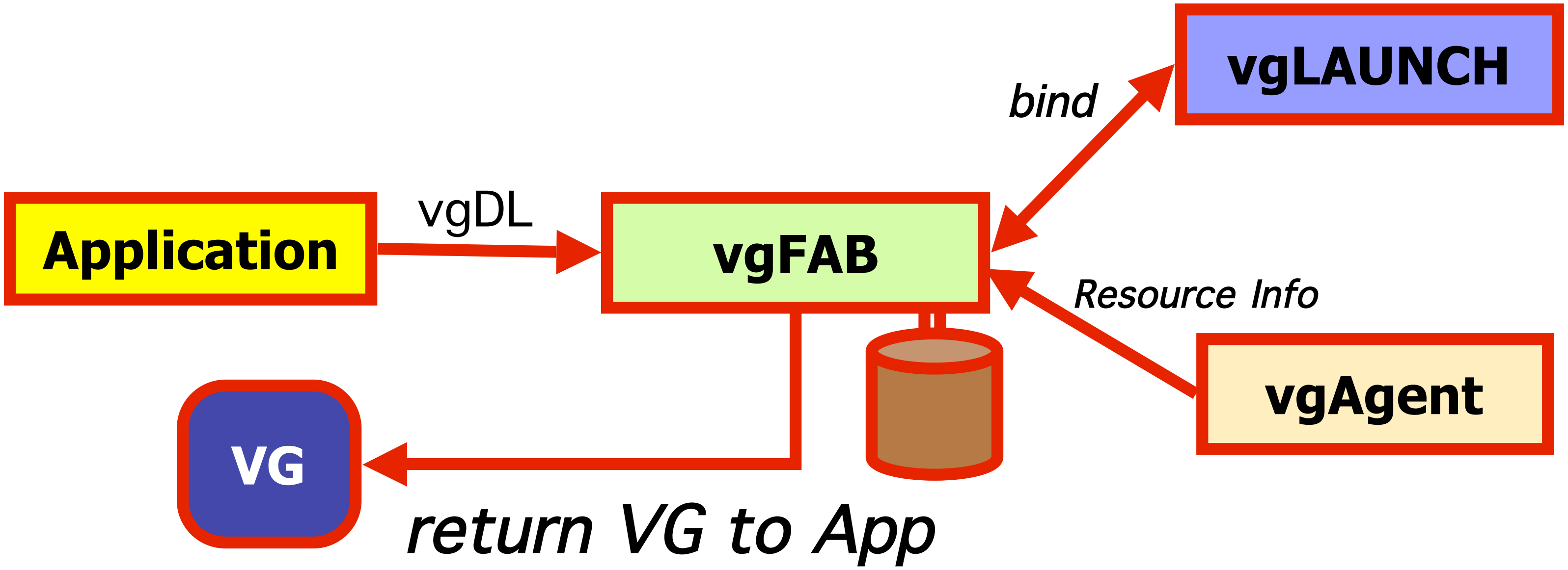
- We have a working research prototype for vgFAB within vgES
  - interfaces with the application via vgDL
  - interfaces with resource information systems
  - returns sets of matching resources
- Makes it possible to use a high-level description of resource requirements
- Makes it possible to find resources over different resource environments
- Research
  - evaluating scalability, result quality, etc.

[CCGrid'05] [SC'05 submission]

Demonstration #2: vgES

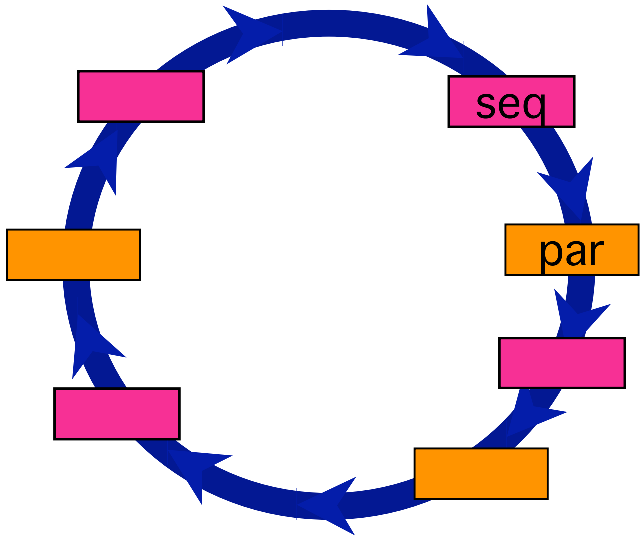


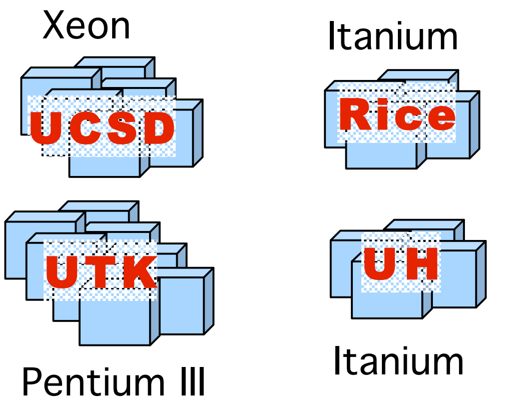
Demonstration #2: vgES



Application: EMAN

- EMAN: Electro Micrograph ANalysis
  - Performs 3-D reconstructions
  - Workflow Application
  - See Chuck Koelbel's talk this afternoon, and the EMAN poster
- Demonstration: Run EMAN on the VGrADS testbed with vgES





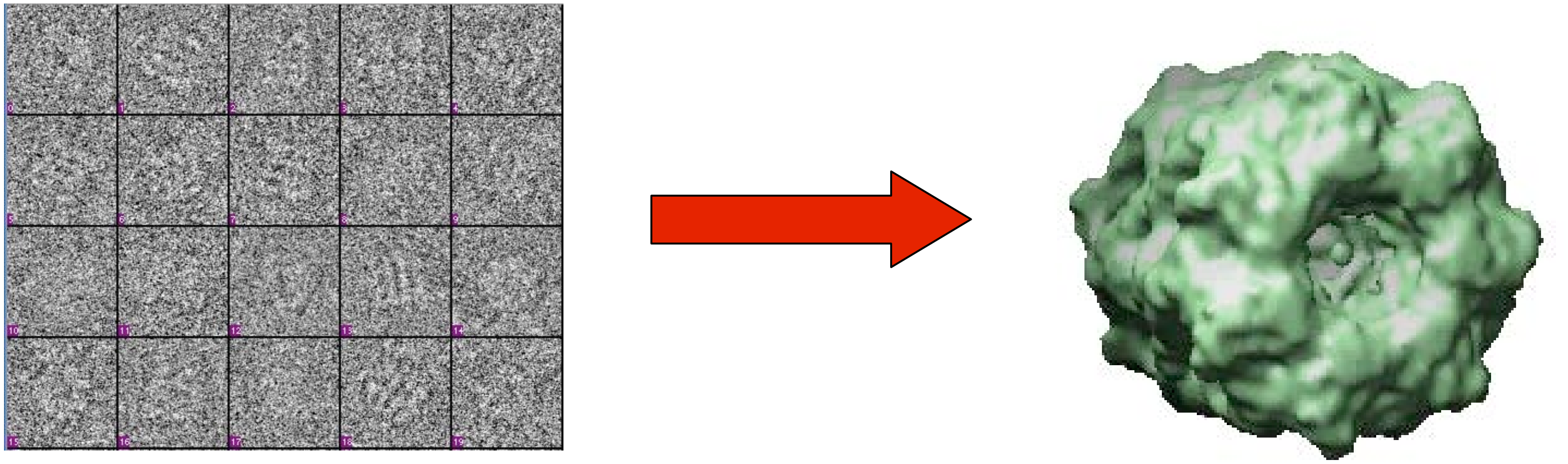
EMAN and the Virtual Grid

```
VG = LooseBagOf (cluster) [1:4] {
          cluster = ClusterOf (node) [4:10] {
                 node = [Clock >= 900]
          }
     }
```

```
VGES myVGES = new VGES()                    // new vgES instance
myVG = myVGES.createVG( vgDL_string )       // found & bound VG
vgRoot = myVG.getRoot();
...   // Traverse VG tree to find resource information
vgES.copyToNode(someNode, “input1”);        // Send input files
vgES.runCmd(someNode,”command”);            // Start command
vgES.copyFromNode(someNode, “output1”);     // Get output files
vgES.terminateVG(myVG);                     // Destroy VG
```

Switch to live demo

Take-away from Demonstration #2



2-D images

3-D model

- **The vgES prototype is functional for a real-application**
- **VG provides a simple abstraction that integrates**
  - access to resources (Globus)
  - access to resource information (MDS, Ganglia, NWS, etc.)

VGrADS and the Virtual Grid

- The VG abstraction and its runtime implementation are the focal point of the VGrADS multi-team collaboration
  - see afternoon presentations and posters

VG

- Interface to applications (vgDL + VG)
- Program preparation and optimization
- Application scheduling
- Monitoring, fault-tolerance, and adaptation
- Resource Management
- Single access/interface to various resource information sources
- Research platform for all the above