



The Legion System

Scott McMaster

CMSC 818S

modified by A. Sussman 2/20/07

References

1. Grimshaw, Wulf, and the Legion Team. *The Legion Vision of a Worldwide Virtual Computer*. CACM January, 1997.
2. Grimshaw, Ferrari, Knabe, and Humphrey. *Wide-Area Computing: Resource Sharing on a Large Scale*. IEEE Computer May, 1999.
3. Natrajan, A. and Humphrey, M. "Legion: The Grid OS Architecture and User View" 2nd International Global Grid Forum Meeting, July 15-18, 2001. (Powerpoint)
4. Natrajan, A. "Legion: The Grid Operating System" Lecture at UMBC, April, 2001.
(<http://legion.virginia.edu/presentations/UMBC-Apr2001/>)
5. The Legion Web Site (<http://www.cs.virginia.edu/~legion/>)

Legion Project Background

- ◆ University of Virginia
- ◆ Then commercialized as a business intelligence and data integration framework (<http://www.avaki.com/>)
 - As far as I can tell, you can't download it anymore.
- ◆ Avaki bought by Sybase, that now sells it as a data integration product

Buzzphrases

- ◆ *World-Wide Virtual Computer*
- ◆ *Metasystem Software*
- ◆ *Wide-Area Operating System*

→ These things just mean “grid” or “grid infrastructure middleware”.

10 Design Objectives

- 
- ◆ Site autonomy
 - ◆ Extensible core
 - ◆ Scalable architecture
 - ◆ Easy-to-use, seamless environment
 - ◆ High performance via parallelism
 - ◆ Single persistent name space
 - ◆ Security for users and resource owners
 - ◆ Management and exploitation of resource heterogeneity
 - ◆ Multiple language support and interoperability
 - ◆ Fault tolerance

3 Design Constraints

- ◆ Must work within host operating systems
- ◆ Must work with existing communication protocols
- ◆ Must run with least privilege

Component-Based System

- ◆ Object-oriented
 - Everything is an object – files, processors, storage devices, networks, users, etc.
- ◆ Core objects are specified as interfaces rather than implementations
 - Has an Interface Description Language (IDL)
 - CORBA for the Grid?
- ◆ Objects are extensible and replaceable
- ◆ Support access control on the function invocation level via the “MayI” method

Some Object Types

- ◆ Host
 - Abstraction of a processing unit
- ◆ Vault
 - Abstraction of persistent storage
- ◆ Class Manager
 - Responsible for object creation, monitoring, and lifetime management

Object Identification

- ◆ Context Name
 - Human-readable
- ◆ Legion Object ID (LOID)
 - Location-independent
- ◆ Object Address
 - Machine-and-communication-specific

Available Tools [3]

- ◆ MPI
- ◆ PVM
- ◆ Remote compilation
- ◆ Postmortem debugger
- ◆ Accounting
- ◆ Licensing

Legion Process (from [3])

- ◆ User initiates “run”
- ◆ User/Legion selects site
- ◆ Legion copies binaries
- ◆ Legion copies input files
- ◆ Legion starts job(s)
- ◆ Legion monitors progress
- ◆ Legion copies output files

Context Space

- ◆ Where the user works with Legion
- ◆ Grid-wide hierarchical directory service
- ◆ Global namespace
 - All resources can be named and referenced using Unix-like paths
- ◆ Contrast with “local space”, the user’s local machine
 - Can transfer files/directories from context to local and vice versa
 - Can also map a local directory to context space temporarily