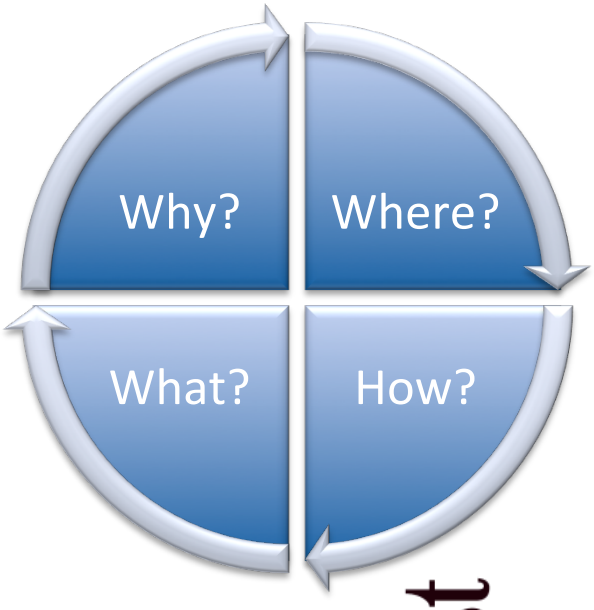
BUILDING VIRTUAL CLUSTER ON COMET

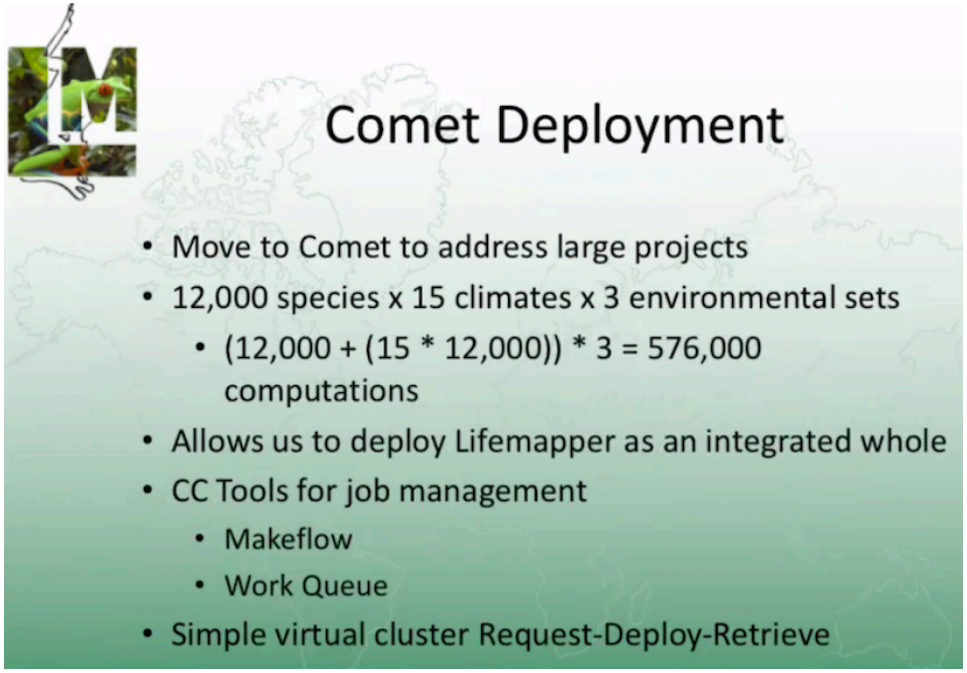
Nadya Williams, UCSD
nadya@sdsc.edu

Need to answer questions



Why Deploying Lifemapper on XSEDE resource Comet?

- Address a large number of small to medium size computations
- Access large data storage over 10Gb network



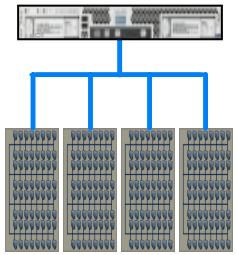
What do we use?

- Rocks http://www/rocksclusters.org
- Technology transfer of commodity clustering to application scientists
- Rocks is a cluster configuration on a set of CD
  - Clustering software (PBS, SGE, Ganglia, Condor, …)
  - Highly programmatic software configuration management
  - To build:
    - Put CDs in Raw Hardware (or access via Network)
    - Enter basic information (network, FQDN, etc)
    - Take a break for coffee
    - Have cluster
- Extensible using “Rolls”
- Large user community
  - Over 1PFlop of known clusters
  - Active user / support list of 2000+ users
- Active Development
  - ~2 software releases per year
  - Code Development at SDSC
  - Other Developers (external rolls)
- Supports Redhat Linux, Scientific Linux, Centos
- Can build Real, Virtual, and Hybrid Combinations (2 – 1000s)

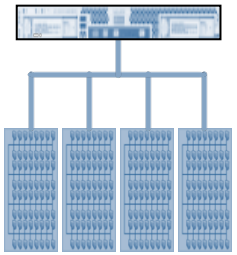
Rocks Core Development
NSF award #OCI-0721623

What is the advantage? Rocks and Virtual Hardware

- **Just another piece of hardware**.
  - If RedHat supports it, so does Rocks
- Allows mixture of real and virtual hardware in the same cluster
  - Because Rocks supports heterogeneous HW clusters
- Re-use of all of the software configuration mechanics
  - E.g., a compute node is a compute node, regardless of physical or virtual
- Virtual hardware must meet minimum hardware Specs:
  - 1GB memory
  - 36GB Disk space (can be bigger)
  - Private network Ethernet
  - Public network on Frontend



Physical



Virtual

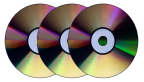
Rocks Rolls:

- **Kernel**
- **OS**
- **Base**
- **Kvm**
- **Ganglia**
- **Python**
- **SGE**
- **HTCondor**
- **Cuda**
- **Zfs-linux**
- **…**

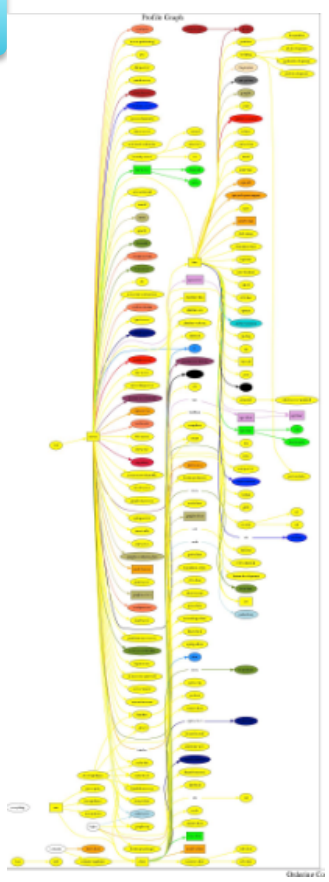
Why do we use Rocks – Key concepts

- Define components of clusters as **Logical Appliances**

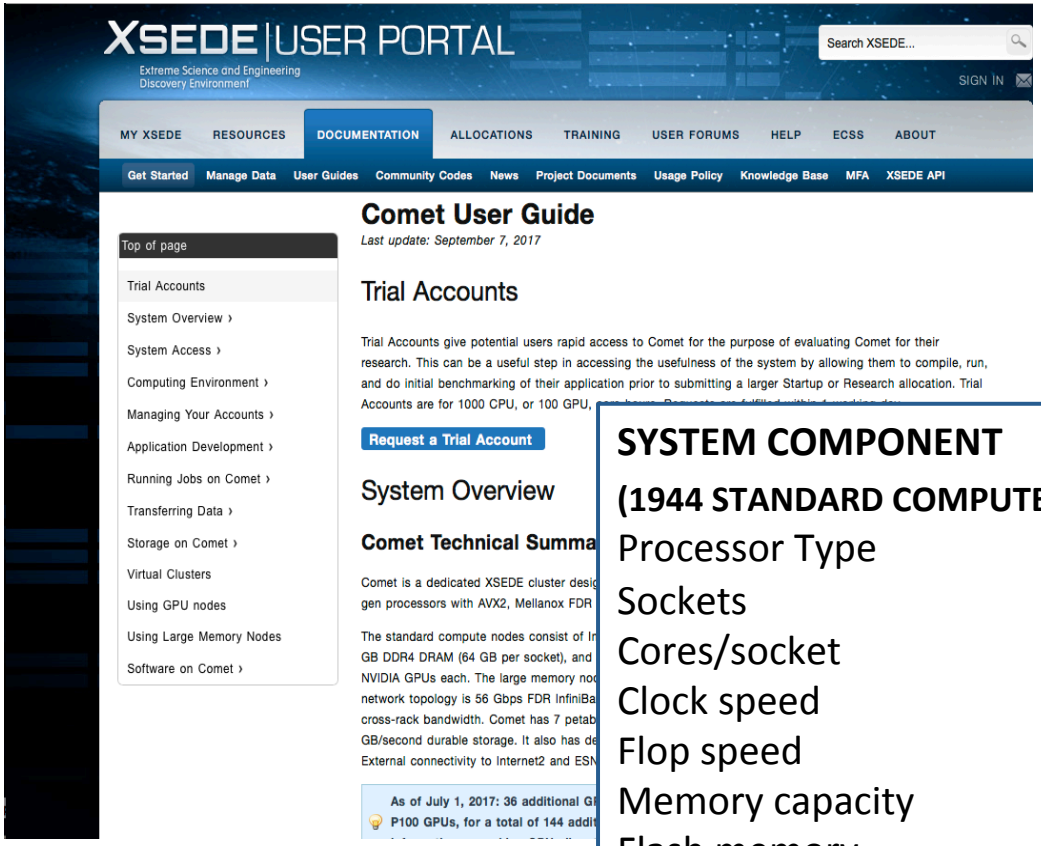
  - Share common configuration among appliances
  - Graph decomposition of the full cluster software and configuration
  - Rocks Rolls are the building blocks: reusable components (Package + Configuration + Subgraph)



- Use native (Redhat Anaconda installer) **text format to describe** an appliance configuration
  - Walk the Rocks graph to compile this definition
- Heterogeneous Hardware (physical or virtual) with no additional effort



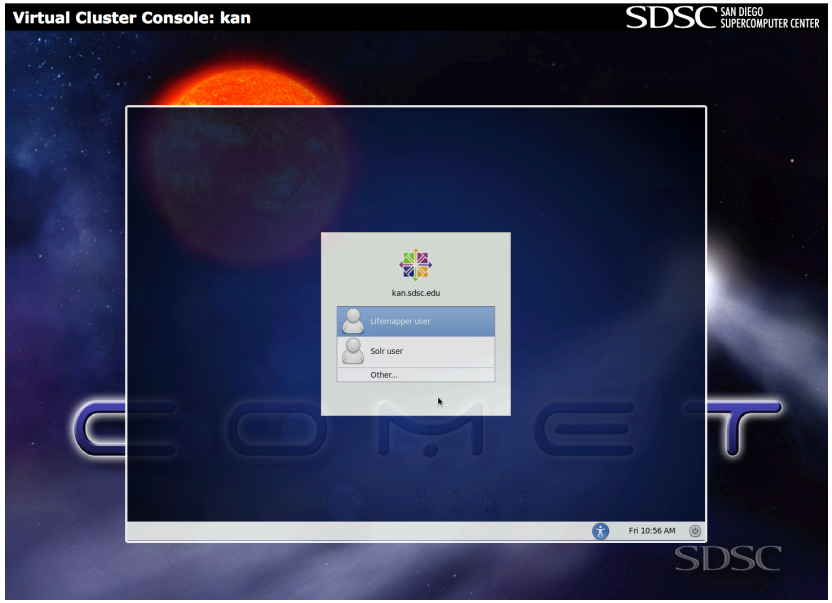
Where to build ?



SYSTEM COMPONENT	CONFIGURATION
(1944 STANDARD COMPUTE NODES)	
Processor Type	Intel Xeon E5-2680v3
Sockets	2
Cores/socket	12
Clock speed	2.5 GHz
Flop speed	960 GFlop/s
Memory capacity	128 GB DDR4 DRAM
Flash memory	320 GB SSD
Memory bandwidth	120 GB/s
STREAM Triad bandwidth	104 GB/s

Comet Virtual Cluster

- Software stack
  - Rocks 6.2 (OS, kernel, base…)
  - SGE
  - Web server
  - Python
  - Cloudmesh

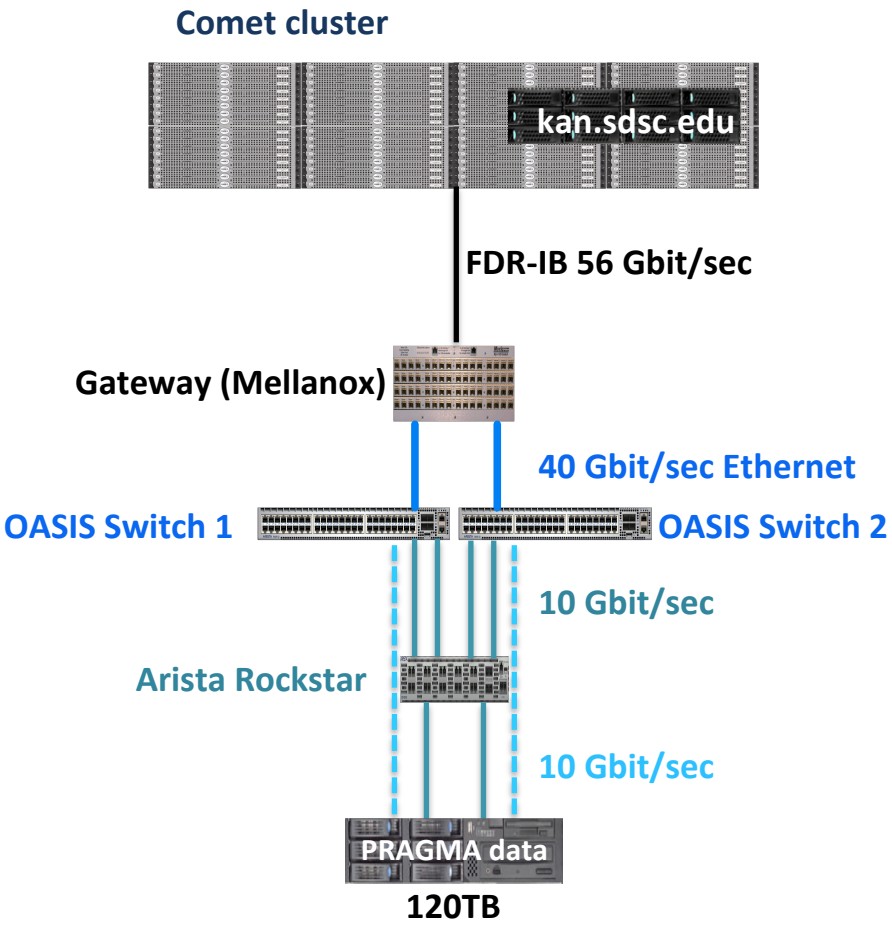


Identical to PRAGMA clusters

See cluster install movie: http://goo.gl/fPiYtj

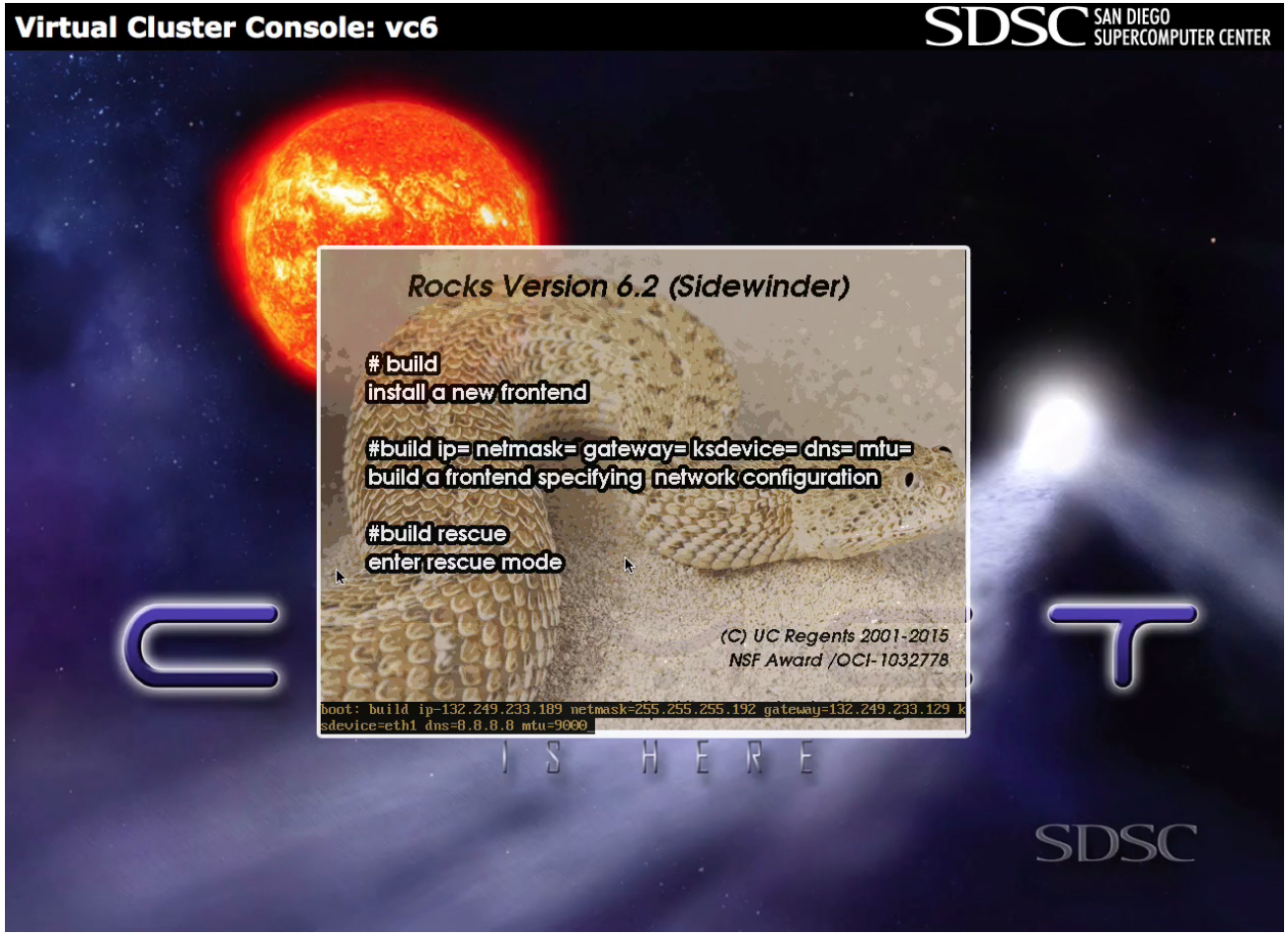
Name	Kind	CPUs	RAM(Gb)	Disk(Gb)
kan	1 Frontend	4 E5-2640 v3 @ 2.60GHz	32	36
vm-kan-0-x	8 compute	24 E5-2640 v3 @ 2.60GHz	96	36

Virtual Cluster Connectivity to large PRAGMA storage



- PRAGMA data 120Tb (raw 160Tb) $10k
- NFS read/write ~ 5 Gbit/sec from kan.sdsc.edu to PRAGMA data server
- Iperf
  - 1 stream ~ 9.8 Gbit/sec
  - 2 streams ~ 15.5 Gbit/sec

90 Second Cluster Build Movie



Related Links

Rocks clusters

https://github.com/rocksclusters

PRAGMA github

https://github.com/pragmagrid

XSEDE Portal

https://portal.xsede.org/sdsc-comet

Lifemapper PRAGMA rolls

https://github.com/pragmagrid/lifemapper-compute

https://github.com/pragmagrid/lifemapper-server

Lifemapper code

https://github.com/lifemapper