Resources/Data/Expeditions Working Group

High-level Summary of Current Activities

- Expeditions
  - Biodiversity: significant improvements in Lifemapper utility and portability
    - Student Challenge, Demo
  - Lake Ecology: Network Overlay (IPOP) + Condor + GLEON.
  - PRAGMA-ENT: First experiments now able to run. Testbed evolution.
- Technology Development
  - Personal Cloud Controller - multi-site virtual cluster creation via enhanced PCC-HTCondor (VM GAHP) and Condor DAG capabilities. IPOP integration
  - Fingerprinting Applications to discover Software dependencies. Demo
- Support
  - Virtual Cluster Allocated @ UCSD for Cyberlearning Expedition

Virtual Biodiversity Expedition
Lifemapper/Rocks update

Goal

Prepare Lifemapper components to be deployed alone or together to work on user-defined and restricted data, i.e. licensed satellite data and sensitive occurrence data for Mount Kinabalu, Sabah, Malaysia

Activities

- Generalize Lifemapper code
- Create installation process
  - Simplify configuration
  - Create installation scripts
  - Automate deployment
- Deploy at institution
- Deploy on a laptop for fieldwork and teaching
- Learn to work together efficiently
  - Meet in person for debugging sessions
  - Maintain ongoing conversation to improve process
  - Repeat mantra: Don't jump to conclusions

Virtual Biodiversity Expedition
Lifemapper/Rocks update

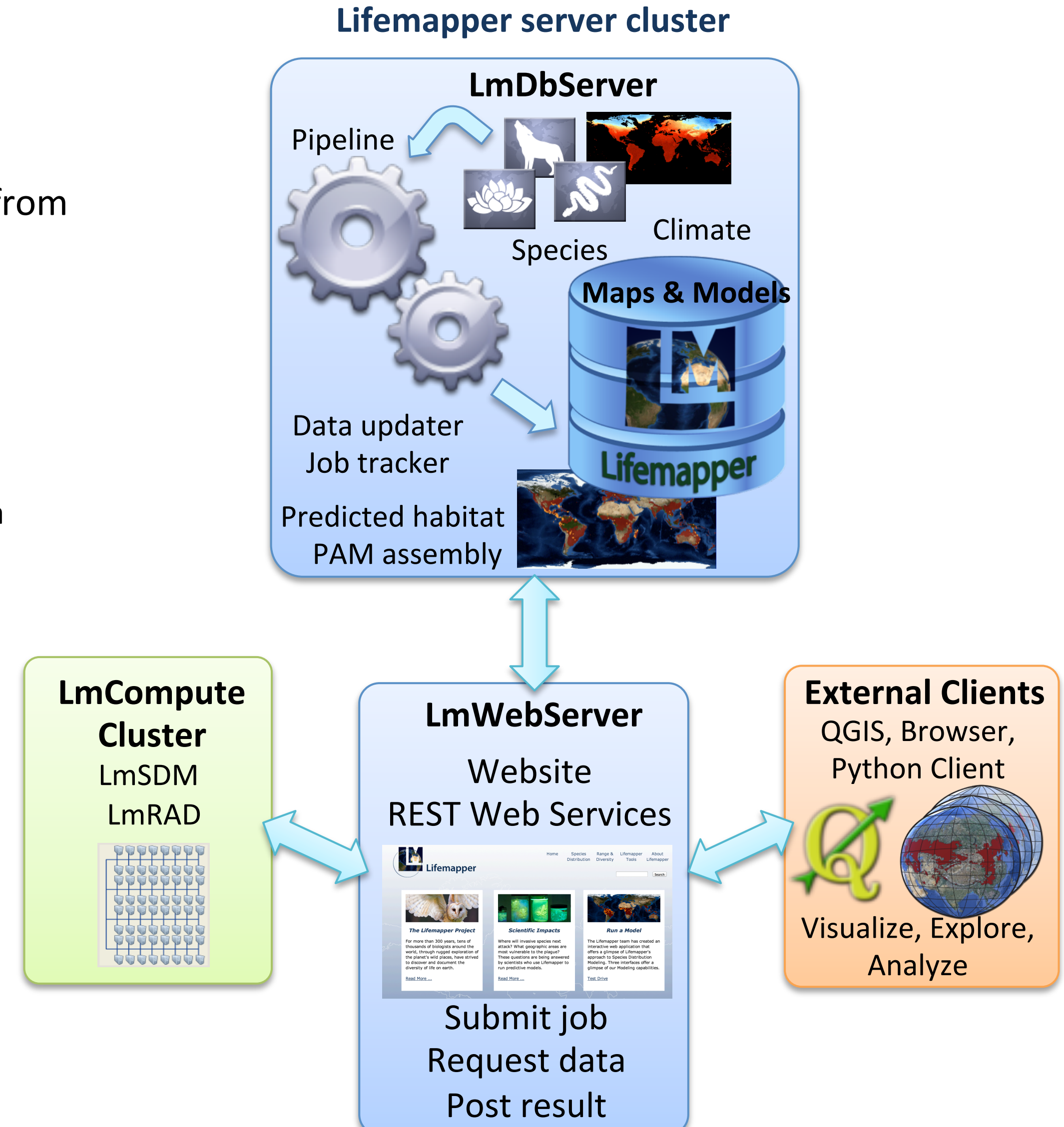
Goals

- Decouple LmDbServer and LmWebServer from KU-specific implementation
- Automate install and configure process
- Bootstrap database
  - Automate database definition
  - Automate static input data seeding
- Run job pipeline on LmDbServer
  - Read and insert dynamic input species data
  - Create job-chain for each input species
- Deploy and connect LmCompute
  - Request jobs from LmWebServer
  - Calculate jobs
  - Return results to LmWebServer

People

KU: Aimee Stewart

UCSD: Nadya Williams

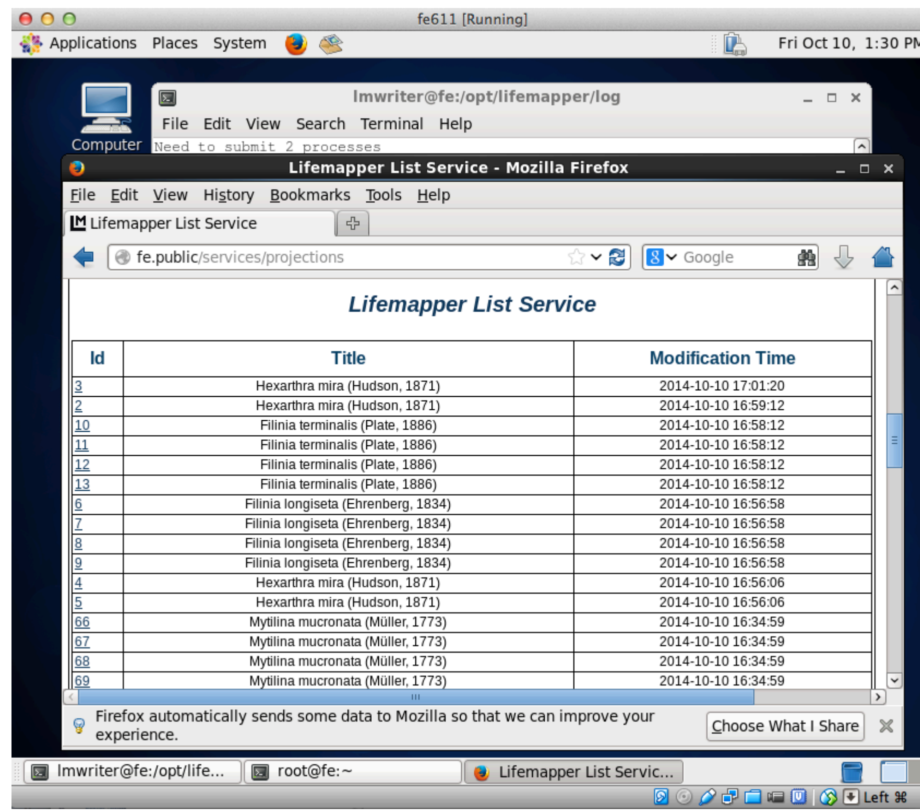


Virtual Biodiversity Expedition
Lifemapper/Rocks update

Accomplishments

- Run all Lifemapper components outside of KU
- Create Lifemapper component ISOs for remote installations
- Deploy in 2 virtualization tools
  - KVM: 2 clusters
    - rocks-201 as a compute component
    - rocks-204 as server component
  - VirtualBox:
    - Run a 2-node cluster
    - frontend as a server + LmCompute component
    - compute node as LmCompute for executing the jobs.
    - Time from VM import to running pipeline ~1 hr (includes data import and seeding)
- Run Lifemapper Server for student challenge

VirtualBox installation:
Lifemapper web page for computed projections



PRAGMA-GLEON Expedition Goals

- Beginning to learn how an extreme ecosystem phenomenon – algal blooms – is expressed in observational data from sensor networks
  - And just how computationally challenging reproducing those events in model simulations can be
- We apply PRAGMA technologies to tackle distributed computing in an innovative way
  - Enabling the community to pool together resources across institutional boundaries, and allowing researchers to continue to use a computer environment they are used to
  - Distributed HTCondor Windows pool connected by IPOP virtual network overlay

People

- U. Wisconsin
  - Paul Hanson, Craig Snortheim
- Virginia Tech
  - Cayelan Carey, Jon Doubek
- U. Florida
  - Renato Figueiredo, Ken Subratie, Youna Jung

Activities

- Close collaboration
  - Computer science and lake ecology researchers
  - Student engagement; excursion (this PRAGMA)
  - Community engagement: GLEON’16 workshop
- Documentation
  - How-tos, installation and use; GitHub Wiki
- IPOP development and deployment
  - Windows service, packaging, XMPP service
- HTCondor integration
  - Generation of HTCondor submit files for GLM model

Accomplishments

- Deployment and documentation of Windows HTCondor pool and IPOP overlay
  - Manager+workers at UF (growing to 48 cores)
  - Submit+workers at Uwisc and VT
  - https://github.com/GRAPLE/documentation/wiki
- Will present demonstration at this PRAGMA
- Joint workshop at GLEON'16

PRAGMA 27

PRAGMA Experimental Network Testbed (ENT)

Maurício Tsugawa (UF)

Kohei Ichikawa (NAIST)

ENT Goals

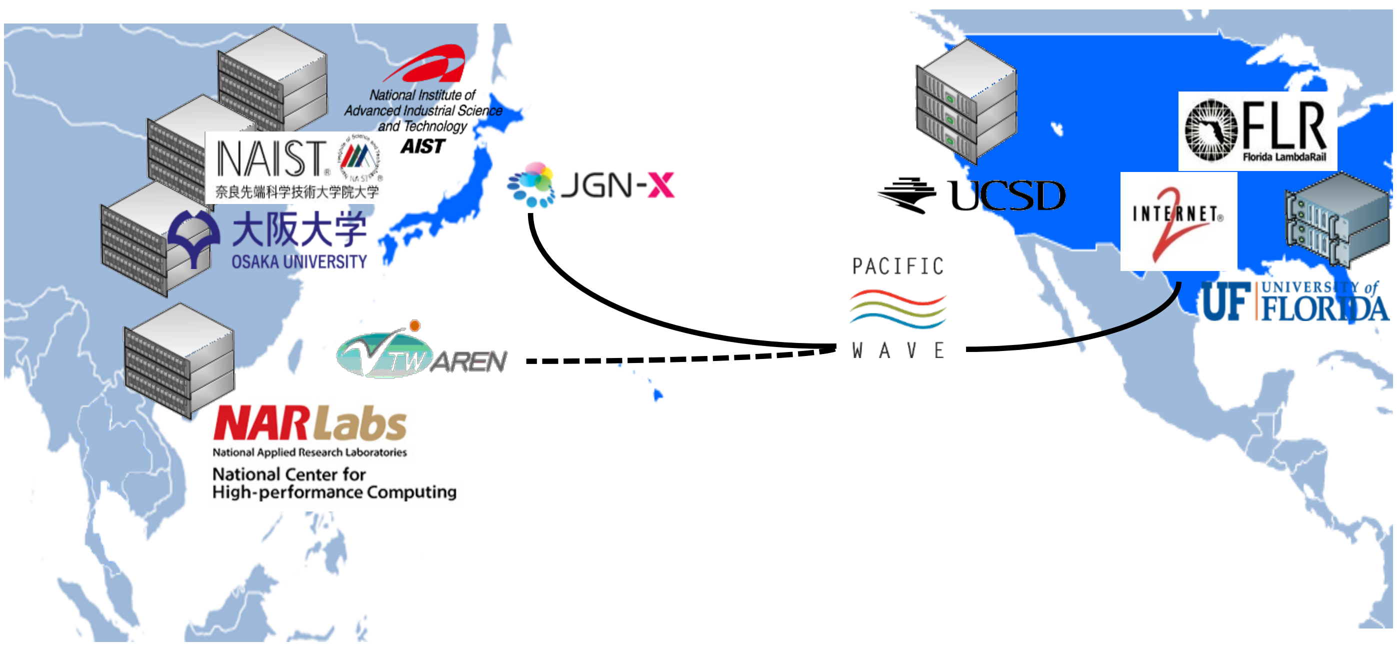
- Build a breakable international SDN testbed for use by PRAGMA researchers
  - By no means a production system
  - Complete freedom to access and configure network resources
- Provide access to SDN hardware/software to PRAGMA researchers
- Offer networking support for PRAGMA multi-cloud and user-defined trust envelopes

ENT Members

- University of Florida
  - Maurício Tsugawa
- Osaka University
  - Shinji Shimojo
  - Susumu Date
  - Yasuhiro Watashiba
  - Yoshiyuki Kido
- Nara Institute of Science and Tech
  - Kohei Ichikawa
  - Pongsakorn U-chupala
  - Chawanat Nakasan
- University of California, San Diego
  - Phil Papadopoulos
  - Luca Clementi
- Advanced Industrial Science and Tech
  - Atsuko Takefusa
  - Yoshio Tanaka
  - Jason Haga
- Indiana University
  - Jim Williams
  - Jennifer Schopf
- National Institute of Information and Communications Technology
  - Jin Tanaka
  - Hiroaki Yamanaka
- Jilin University
  - Xiaohui Wei
- Computer Network Information Center – Chinese Academy of Sciences
  - Ren Young Mao
  - Kejun Dong
- National Center for High-performance Computing
  - Fang-Pang Lin
  - Te-Lung Liu
  - Li-Chi Ku
- Kasetsart University
  - Putchong Uthayopas

ENT Activities

- ENT Backbone
  - VLANs allocated, mapped, and peered (FLR, Internet2, JGN-X, and Pacific Wave), connecting UF, UCSD, NAIST, AIST, and Osaka U.
  - TWAREN to be configured soon.



ENT Activities

- Controllers and applications
  - Rocks provisioning through ENT
  - Network slicing technology (AutoVFlow by H. Yamanaka) being evaluated in ENT
  - Virtual cluster across ENT to run dock (DEMO)
  - Controller for multipath routing (DEMO)

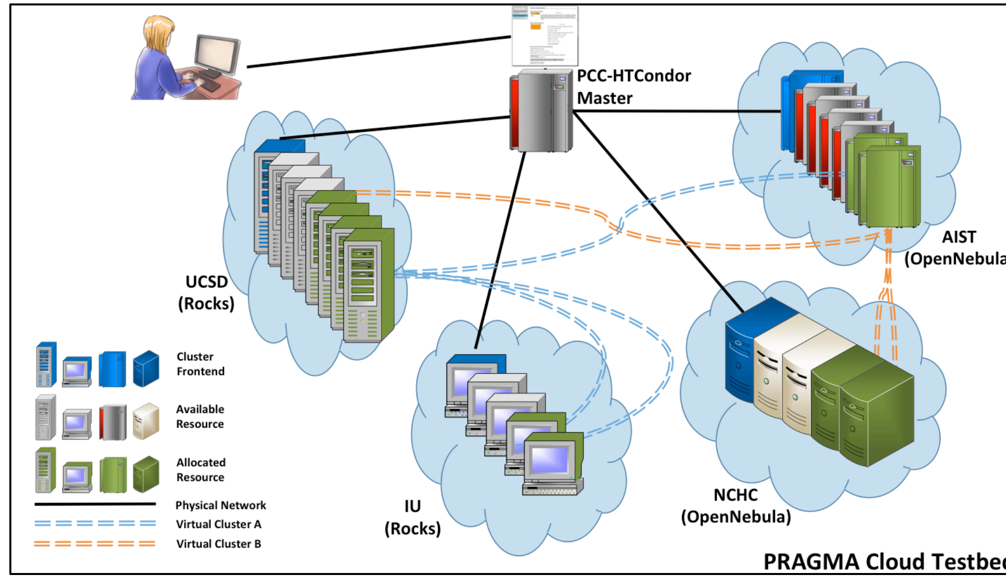
ENT Accomplishments

- ENT backbone live
  - Layer 2 connectivity between UF, UCSD, NAIST, AIST, and Osaka U.
  - Alternative paths established through GRE tunneling for multipath routing research
- Successful “remote” provisioning
  - Rocks front-end at UCSD provisioning a node at UF
- OpenFlow Controllers and Applications under development

Personal Cloud Controller (PCC)

(Yuan Luo, Shava Smallen, Beth Plale, Philip Papadopoulos)

- Goals:
  - Enable lab/group to easily manage **application virtual clusters** on available resources
  - Leverage PRAGMA Cloud tools: pragma_bootstrap, IPOP, ViNE.
  - Lightweight, extends HTCondor from U Wisc.
  - Provide command-line and Web interfaces
- Working Group: Resources



HTCondor High Throughput Computing

ROCKS

IPOP

pragma_boot

PCC Activities and Accomplishments

(since April 2014)

- Experimented with IPOP and measured initialization and bandwidth performance between IU and SDSC

- Drafted a paper "A Personal Cloud Controller Framework" for submission.

A Personal Cloud Controller Framework

Yuan Luo*, Shava Smallen†, Philip Papadopoulos†, Beth Plale*
*School of Informatics and Computing, Indiana University Bloomington, Bloomington, IN, 47405, USA
†San Diego Supercomputer Center, University of California, San Diego, La Jolla, CA, 92093, USA
{yuanluo, plale}@indiana.edu, {ssmallen, phil}@sdsc.edu

***Abstract*—User controllability, cloud interoperability and information sharing between resource and application together form an unmet need in management of applications on widely distributed computers. The personal cloud controller (PCC) addresses the need by creating virtual clusters that allocate virtual machines from multiple cloud platforms, building a network overlay, and managing the virtual cluster life-cycle in a fault-tolerant manner. A resource allocation algorithm is introduced that factors in resource specifications and application requirements and characteristics. The implementation of PCC extends HTCondor and integrates IPOP to build the network overlay. Experimental results show that the overhead of PCC during virtual cluster booting is reasonable and the overhead of running applications over IPOP is acceptable due to the minimum amount of configuration setup.**

***Keywords*-Cloud computing, Controllability, HTCondor, PRAGMA**

I. INTRODUCTION

Cloud computing is widely used in business and scientific settings to create scalable, flexible and cost effective environments to process large datasets. In this paper, we present a personal cloud controller (PCC) that provides user controllability, cloud interoperability and resource-application knowledge sharing. The direct motivating application of PCC is to facilitate user management of resources and optimize the performance of scientific applications in a multi-institutional academia cloud, PRAGMA Cloud [1]. The PRAGMA Cloud infrastructure is both heterogeneous and distributed, with resources from multiple institutes from the Pacific Rim and United States. The PRAGMA Cloud is loosely-coupled with differing cloud software, cluster sizes, and hardware architectures. PCC enables users to author their own application virtual machines (VMs) using their preferred VM platforms and then deploy these VMs on demand as a virtual cluster across different PRAGMA Cloud sites. The three main capabilities of PCC are described below.

User Controllability: A user (user application) takes full control of their allocated resources and easily manages the lifecycle of their applications. There is a gap between managing resources by resource providers and utilizing resources by applications, which is often manually filled by system administrators. In order to eliminate the gap, a personal cloud controller needs to be in place to maintain controllability of the resources and applications. The system needs to be stateful so that resource status and application deployment/execution are monitored and recorded to enable easy management and fault tolerance.

Cloud Interoperability: Cloud consumers use their data and services across multiple cloud providers using a unified management interface. A researcher typically has access to several research clusters at their institution, each of which consists of a small number of nodes and a limit on the maximum number of nodes a user can allocate at any one time. Since applications requires a larger number of resources, single-cloud solutions grow increasingly inadequate. Aggregating these isolated clusters into a single virtual cluster provides a powerful platform for data-intensive jobs that require access to large datasets stored across multiple locations and geographies. For instance, scientific communities within PRAGMA share data from multiple sources and locations so that it becomes critical to utilize data distribution information, notably, data locality, to optimize the performance of data-intensive jobs.

Resource-Application Information Sharing: To perform resource allocation and application execution efficiently, information needs to be shared bi-directionally between resource providers and applications. Resource allocation is a subject that has been addressed in many computing areas, such as operating systems, grid computing, and cloud computing. A resource allocation mechanism tries to guarantee that the users' minimum requirements are attended to correctly by the provider's infrastructure. To better allocate resources to users and thus minimizing the operational cost of the cloud environment, a resource allocation mechanism should also consider the current status of each resource in the cloud environment. The current Infrastructure-as-a-Service based cloud systems are usually unaware of the characteristics of the applications and therefore allocate resources independently of its application profiles, which can significantly impact performance and security for distributed data-intensive and data-sensitive applications.

When allocating resources from resource providers, the system applies certain methodologies to achieve optimization goals, such as system utilization, load-balancing, etc. When resources are allocated, another optimization effort is performed by applications. However, if performed independently, either one of optimization yield optimal performance. In a particular application for instance, two nodes that

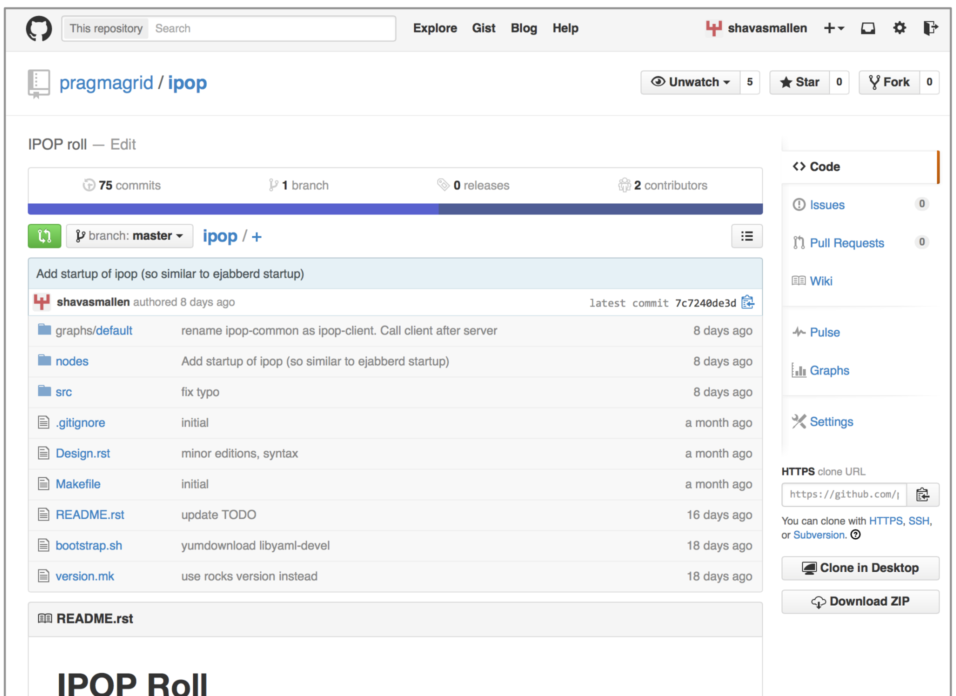
PCC Activities and Accomplishments (cont.)

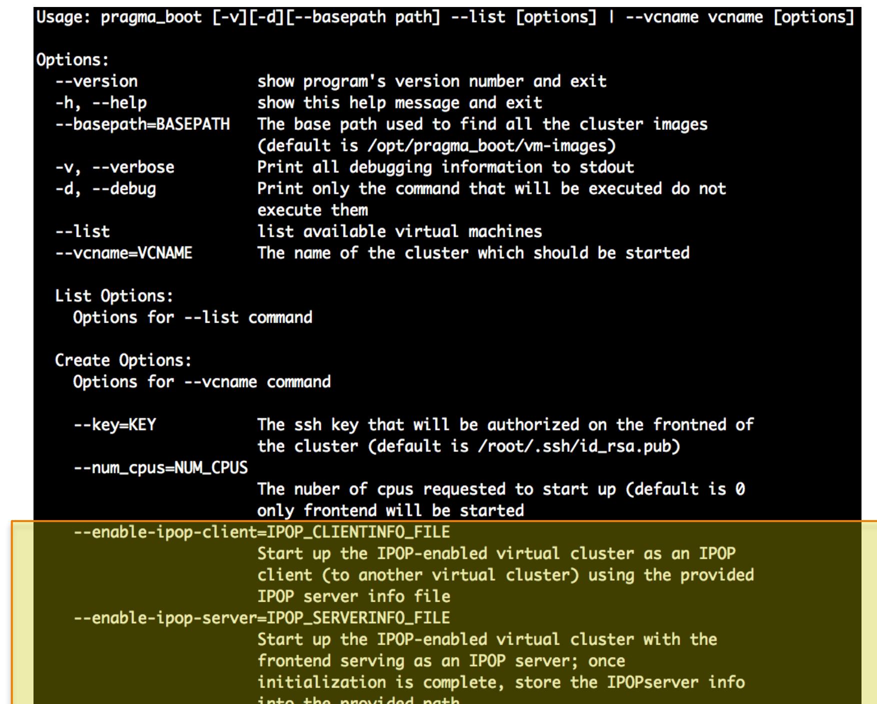
(since April 2014)

- Developed new IPOP Rocks roll for easy installation of IPOP to any Rocks virtual cluster

(thanks to Nadya Williams!)

- Added automated IPOP server/client initialization to pragma_boot





PCC Activities and Accomplishments (cont.)

(since April 2014)

- Enabled multi-site virtual cluster creation via enhanced PCC-HTCondor (VM GAHP) and Condor DAG capabilities
  - Will demo between SDSC, IU, and AIST

Parallel pragma_boot jobs