HTCondor at Syracuse University – Building a Resource Utilization Strategy

Eric Sedore
Associate CIO

HTCondor Week 2017

Syracuse University
Information Technology Services (ITS)

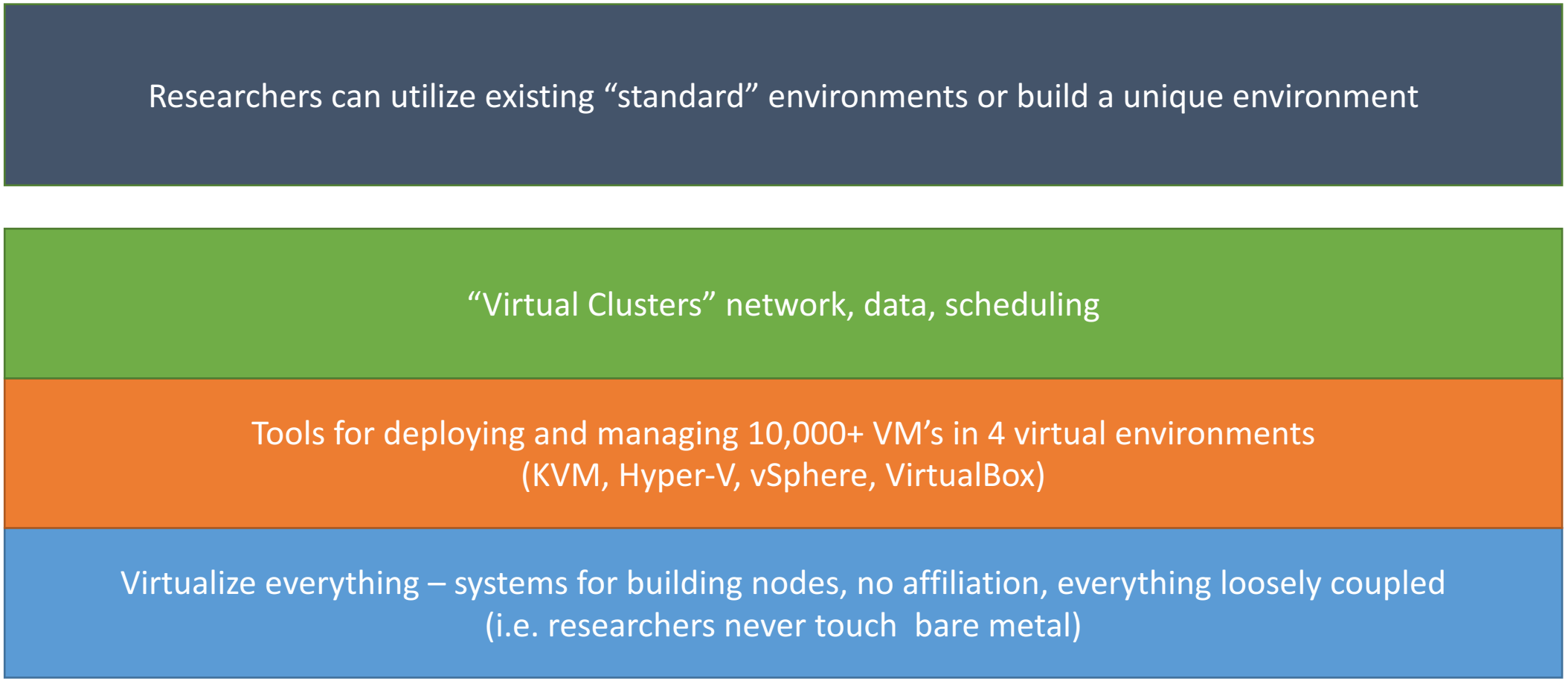
Research Computing Philosophy @ Syracuse

- Good to advance research, best to transform research (though transformation is not always related to scale)
- Entrepreneurial approach to collaboration and ideas
- Computing resources are only one part of supporting research
- **Strive to use computational resources at 100% utilization, 100% of the time**
- Computational resources must support multiple academic areas

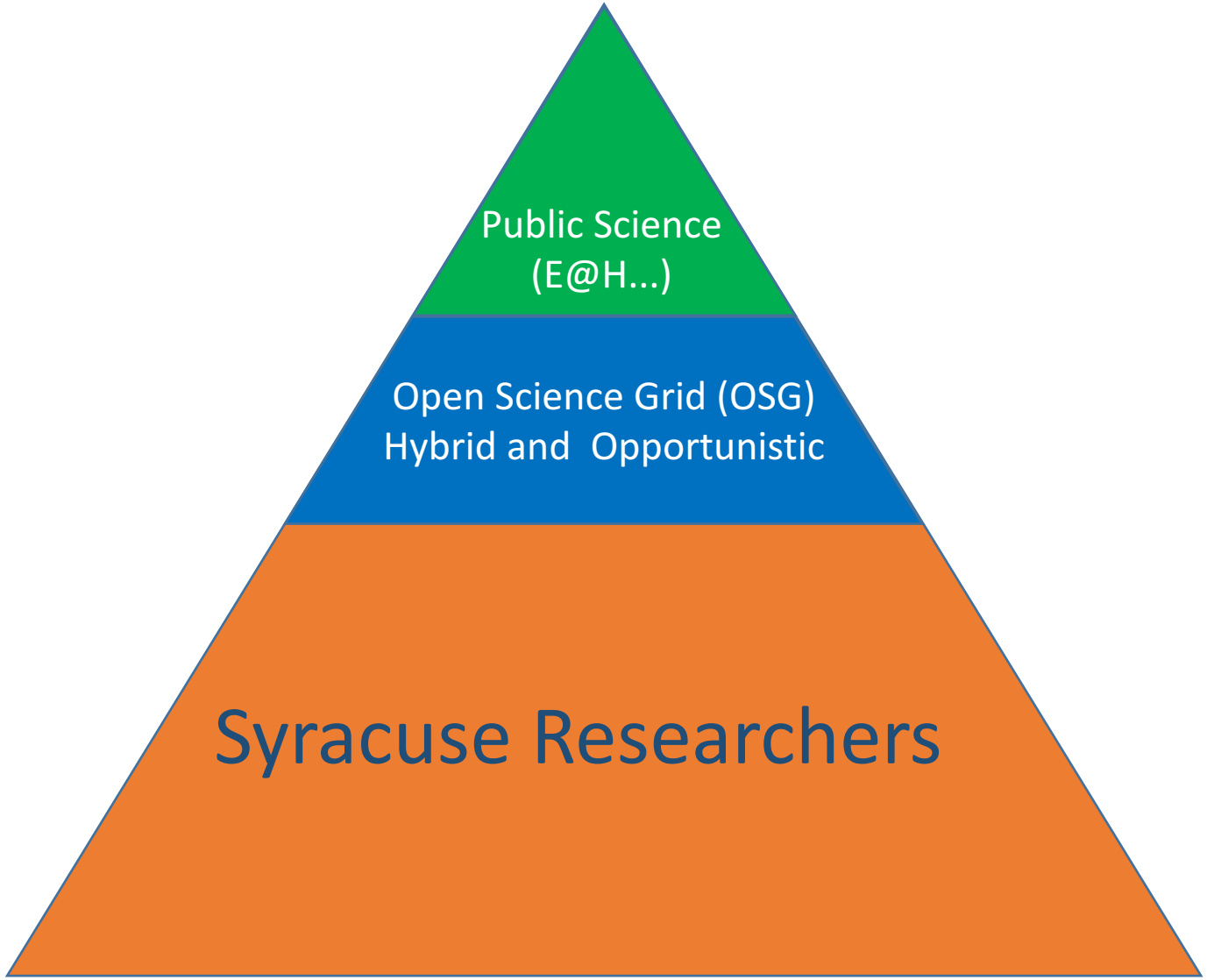
Computational Resources @ Syracuse

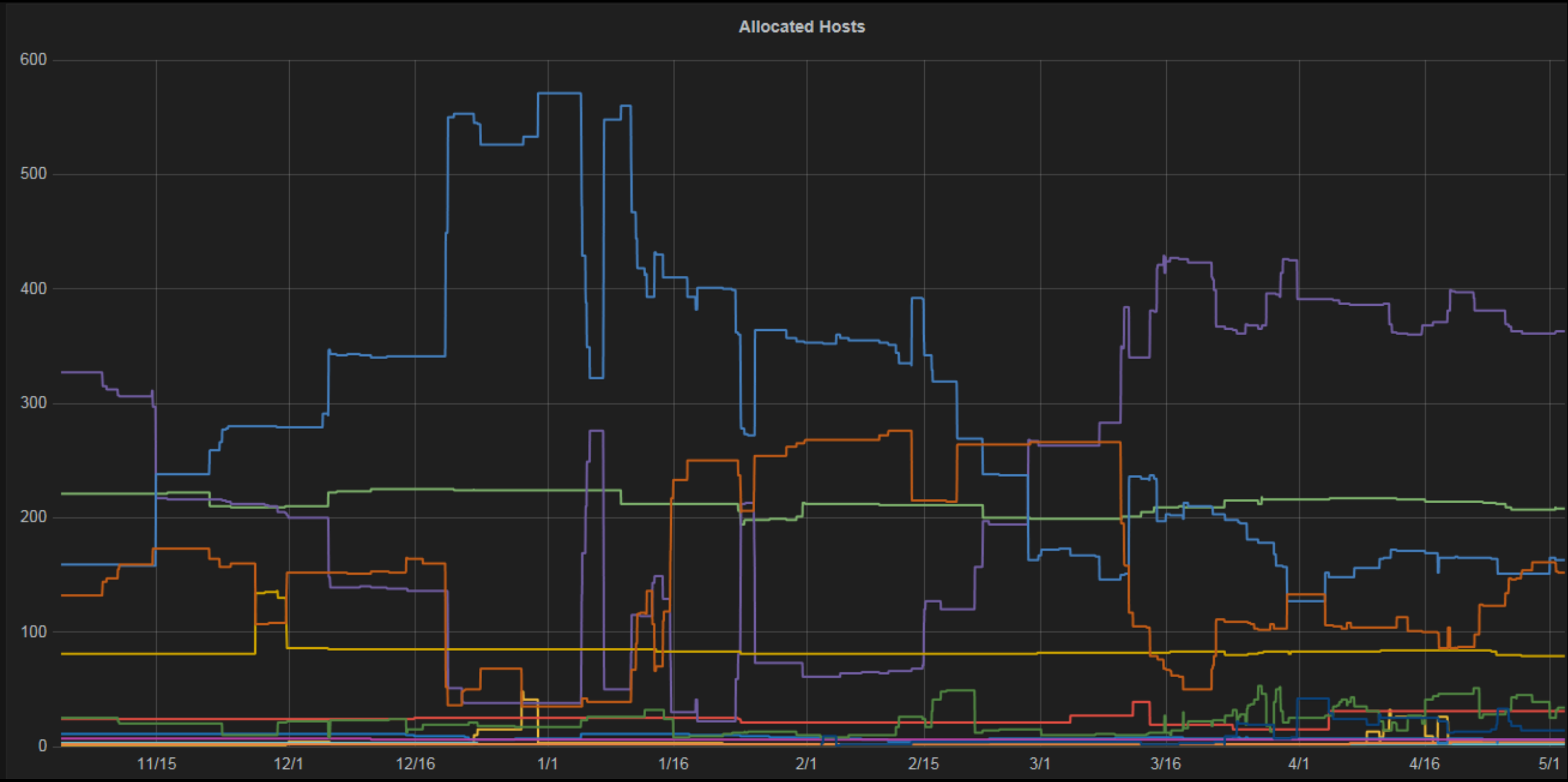
- Academic Virtual Hosting Environment (AVHE) – private cloud
  - 1000 cores, 25TB of memory
  - Individual VMs (students, faculty, staff), small clusters
  - 2 PB of storage (NFS, SMB, DAS per VM), multiple performance tiers
- OrangeGrid – high throughput computing pool
  - scavenged desktop grid, 13,000 cores, 17TB of memory
- Crush – compute focused cloud
  - Coupled with the AVHE to provide HPC and HTC environments
  - Made up of heterogeneous hardware, different areas within Crush are focused on different needs (high IO, latency/bandwidth, high memory requirements…)
  - 12,000 cores (24,000 slots with HT), 50 TB of memory
- SUrge – GPU focused compute cloud
  - 240 commodity NVidia GPUs
  - Individual VMs / nodes scheduled via HTCondor

Resource deployment

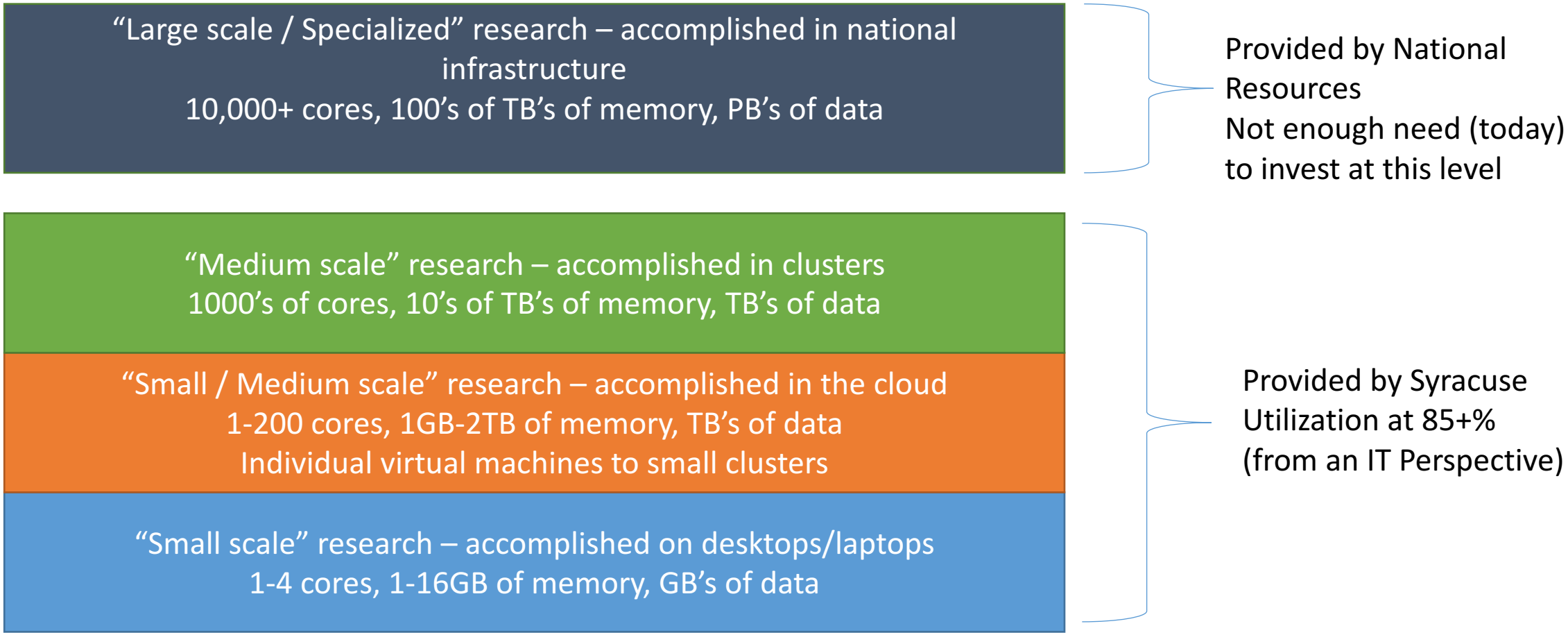


Allocation of resources




Allocated Hosts
600
500
400
300
200
100
0
11/15
12/1
12/16
1/1
1/16
2/1
2/15
3/1
3/16
4/1
4/16
5/1

What resources should Syracuse provide?



Core Elements

- HTCondor
  - Primary tool for resource scheduling – everything (almost) else is a pain!
  - Node advertising capabilities
  - Simplicity of addition/removal of nodes (part its scavenging roots)
  - Flexibility – small simple environments to larger more complicated environments

- Virtualization (KVM, Hyper-V, vSphere, VirtualBox)
  - Abstraction – shim allows us to easily reallocation resources, including networking and storage
  - Flexibility – easy to run multiple kinds of workload (Windows/Linux)

- In-house coding / scripting – primarily in management / deployment – interacting with hypervisors

Pain Points

- VM Management – we have ~20 VM environments within Crush alone
  - Versioning, automation, best of breed VM / monolith VM
  - What do we need? Singularity / Docker When do we need it? Now!
- Staff Expertise
  - Complexity, staff resources, single person dependencies - systems focused on being operated by a fraction of a staff member
  - Nuance/elegance is lost, often the “right way” is set aside in the necessity to move on to the next

Musings on Our HTCondor Experience

- Law of unintended consequences is alive and well – changes always have impact
- There is a knob for everything…
- Logging is spectacular, deep, voluminous - “a blessing and a curse”
- You can have multiple versions of HTCondor components in your environment, but anecdotally you will occasionally find “odd” interactions