Amy Apon
Patrick Bridges
Charles Leiserson
Renato Figueiredo
Joseph Kizza
Amr El Abbadi
Mike Zink
Jay Akut
Rob Ricci
Remzi Arpacı-Dusseau
- Requirements Suggestions
- Example Classes/Exercises
- Guidance
Basic features

- Isolation – so students can break things without hurting anyone else
- System level hardware access, e.g. VM-level access, introspection
- Predictable, measureable environment
- Minimal setup time, good startup documentation
- Collaborations, community development and support
  - Interfaces similar to other systems used for education, e.g. GENI
  - Lessons learned from GENI
  - Portability between NSFCloud/commercial cloud services like AWS
  - Shared materials/modules/exercises, both from the facilities and contributed by community
  - TA/instructor training sessions
  - Community forums for educators
  - Good data sets, benchmarks, course modules for educational exercises
  - Tie in to SIGCSE, GENI, SC, NSDI etc. related educational events, conferences, planning
  - Community events for NSFCloud community (perhaps co-locate with something else? SC, NSDI BOFs?)
- Integrated course management console
  - control/dole out resources
  - Provision starting images, update management
  - Testing/grading support
  - Hierarchical scheduling and access support (faculty/TA/student):
    - team vs. individual allocations
    - Account monitoring capabilities
    - Control errant student jobs
  - Instructor management of scheduling/deadlines/allocation
Resource allocation

- Short turn-around allocation time
- Longer-term resource allocation for semester courses
- Sufficient capacity to handle class deadlines, capacity
- Support for scalability exercises
- VM development
- Eucalyptus deployments
- Cloud design classes
- Classes that use FPGAs/Accelerator/specialized hardware
- Large classes with high latency requirements
- Performance engineering
- Data center networking classes
Focus on things that can’t be run on commercial or other mid-scale NSF platforms
  - That is, leverage access to XSEDE, AWS, etc., where possible, and use CloudLab resources where they are unique