

1590 Course Structure

Marlon Pierce, Suresh Marru

marpierc@iu.edu, smarru@iu.edu

Projects, Presentations, and Assignments

- Students will be divided into teams.
 - Name your team and send to instructors
- Each team will have a project with intermediate milestones due every two weeks.
 - 70 points: 7 milestones including mid-term and finals
- There will also be a midterm and final presentation.
 - 20 points: 1 team mid-term presentation (10 points) and 1 team final presentation (10 points)
- Smaller assignments will be made on alternating weeks.
 - 10 points total
- There will be occasional opportunities for bonus points.
 - Optional, 1 point each
- Grades:
 - 90-100 points: A
 - 80-89 points: B
 - Etc

Project Milestones: 70 points total

- There will be 7 project milestones, including mid-term and final milestones, scheduled biweekly.
- Use open source tools, languages, etc.
 - Document your license requirements
- Each project milestone is worth 10 points.
 - The team receives the points
 - Projects must be checked into github, must be reproducibly executable on the deadline day by the instructors.
 - Linux, UNIX compatible
 - If the instructors cannot execute your project and verify you have met the success criteria, the team receives 0 points.
 - A team may resubmit their assignment at any time before the next milestone. Each resubmission gets -1 point; i.e., 9 points if you get it right on the second try, 8 points on the third try, etc.
- Individual students who show no activity (no github commits, no communication about design on GitHub issues, no email discussions, etc) for the milestone will receive 0 points for their team's assignment.
- Milestones build toward the final project, so you must complete each

Mid-Term Presentation: 10 points total

- These will be demonstrations to the class and presentations. Each team member must contribute.
- Mid-Term Presentation and Demo: Basic working gateway
 - Gateway consisting of a web user interface and application server that allows users to
 - Login
 - Submit jobs to Big Red 2
 - Monitor jobs
 - View job archive
- Final Presentation and Demo: Gateway built with microservice architecture
 - Defined API
 - Defined internal, standalone component (microservice) version of application server.
 - Defined internal APIs, internal communication mechanisms
 - Demonstrated continuous deployment
 - Demonstrated fault tolerance, load balancing

Final Presentation and Demo: 10 points total

- These will be demonstrations and presentations to the class . Each team member must contribute to the presentation/demo
- Final Presentation and Demo: Gateway built with microservice architecture
 - Defined API
 - Defined internal, standalone component (microservice) version of application server.
 - Defined internal APIs, internal communication mechanisms
 - Demonstrated continuous deployment
 - Demonstrated fault tolerance, load balancing
- These are required. Failure to participate in the team presentation and demo will result in an “I” for the individual students

Gap Assignments: 10 points total

- Smaller skill and knowledge building assignments will be given in between project milestones.
- These will be worth 1 point each.
- There may be 1 or more of these per alternate week.
- These are earned by each student, not by the team.

Bonus Points

- As assigned throughout the semester.
- These opportunities will typically be for 1 point each.
- They do not count against you if you do not complete them.
- Bonus points do not allow you to skip the final presentation.
 - You'll get an "I" if you do that.

Assignment A1.1: 1 point

- **DUE:** January 19th
- Students will request accounts, submit jobs to <http://testdrive.airavata.org> using tutorial information
- Assignment: every student gets an account and runs through the demo, which we can verify.
- Need help or found a problem? Email the Apache Airavata developer mailing list.
- **Bonus Point:** email the mailing list with a bug or suggested improvements

Assignment A1.2: 1 point

- **DUE:** January 19th
- Get accounts on BR2, Karst, and submit the job described in the RT tutorial: <http://rt.uits.iu.edu/ci/training/index.php>
- IU accounts on Big Red II, Karst, Quarry - <https://kb.iu.edu/d/aczn#research>
- Accessing resources via SSH
 - Use SSH Keys - <https://kb.iu.edu/d/aews>
- Running jobs with batch queuing systems <https://kb.iu.edu/d/bcqt#jobs>

Assignment A1.3: 1 point

- **DUE:** January 18th
- Github workflow by updating student profile
- Git and GitHub - Walkthrough - <https://help.github.com/>

Project Milestone P1: 10 points

- **DUE:** January 26th
- Application that submits and monitors a job remotely on Karst
 - The application can run anywhere. Must be able to run on instructors' laptops.
 - Use SSH, SCP with SSH Keys (do not store your IU passwords in code or config files).
 - Design choice: wrap executable or use a library?
 - Design choice: how do you monitor jobs?
- You can use any language and build system of your choice. If you do not have a preference, we recommend Java/Maven.
- Must be checked into github
- Use GitHub issue management tools to discuss the project. Use GitHub for all communications.
- Have a build system, also in GitHub
 - Design choice: what do you use?
 - Maven, ant, make, ...