

# Reverse Lectures

(RL1, RL2, and RL3)

# Reverse Lecture (RL) – Are you ready?

Consider the viewpoint of the audience and what story you are telling

- Technical Significance
- Performance results
- Scaling and fault tolerance
- Expansion and adaptability of the software stack
  - Language demonstration
  - Cross stack integration

Envision: A talk you present to peers on achievements and performance.

# RL Continued

- Team size: 1-3
- Format: Code, paper, and a 5 minute presentation
- Hand in an archive (bz2) of your presentation, work (code, results), and report
- Teams can reform after each RL
- Each team contributes to the presentation
  - Slides should be clearly marked with the team's name and topic
  - A slide of references should be provided but not necessarily presented
  - Presentation and slides are factored into the grade
- Presentation covers an aspect of the work (not a summary) + questions
- Zoom presentations

# Reverse Lecture Grading Algorithm

- Defining a contribution algorithm for grading teams:

$$P_t = aT + bD + cS + dP$$

(where a,b,c,d are weighting factors)

- **T**echnical significance of depth – with validation
- **D**ocumentation (report) – 4-7 pages
- **C**ode
- ★ • **P**articipation of members (Code & Presentation)

# The project: So much more than a class and group effort

- Expectations:
  - Real-world project – messy, unknowns, open, and bound
  - Group size of 3-5
  - Group contribution with individual accountability
  - Class (Global) integration – all groups participate
  - Depth in research/implementation or specialization in each group
- Scope and requirements
  - Yes
- Measuring progress
  - Semi-weekly meetings on status and direction
  - Two integration sessions – bring your code!
  - Final demonstration