# Preliminary Results on an Interactive Learning Tool for Early Algebra Education

Siva Meenakshi Renganathan (now at Apple),
**Christopher Stewart**, Arnulfo Perez, Rashmi Rao & Bailey Braaten

Department of Teaching and Learning +
Department of Computer Science and Engineering
The Ohio State University

# Overview

- We designed a web-based learning tool that provides **instant feedback** as students explore pegagogical concepts in early Algebra curriculum

- In 2017, we deployed at 5 schools for 1300 students

- Key principles to support interactive learning tools
  - Co-designing curriculum and systems
  - Regulated data transfer between client and server
  - Integration with Classroom Management systems

# Outline

- Background: STEM curriculum for Algebra I
- Design and Principles for an interactive tool
- Evaluation and Optimizations
- Deployment and Analysis

# Outline

- **Background: STEM curriculum for Algebra I**
- Design and Principles for an interactive tool
- Evaluation and Optimizations
- Deployment and Analysis

# Science and Engineering Driven Mathematics Curriculum

- Absent and/or negative STEM experience in primary education drives students away in college
  - E. Seymour and N. M. Hewitt, “Talking about leaving,” 1997

- Students struggle to link STEM curriculum to applications in real world

# Science and Engineering Driven Mathematics Curriculum

- **We designed a new engineering driven Algebra curriculum**
  - **Explores STEM concepts and applications simultaneously**
  - **Engage, Investigate, Model and Apply framework**
  - **A. Perez, K. Malone, S. M. Renganathan, and K. Groshong, "Computer modeling and programming in algebra", CSEDU, 2016.**
  - **C. V. Schwarz and Y. N. Gwekwerere, "Using a guided inquiry and modeling instructional framework to support preservice K-8 science teaching," Sci. Ed. 2007**
- **Key idea: Teach mathematical concepts alongside STEM applications**
  - **Use a smart classroom portal to link representations of data:** ***Equations, Graphs and Physical data***

# OSU STEM+C Curriculum

1) Guide students to setup the apparatus for scientific experiments
2) Data collection: Run experiments with different inputs and measure corresponding output
3) Interactive visualization: Chart data, plot graph, view equation
4) Update regression curve when data is manipulated; Observe the variation of curve in the graph as the parameters like slope and y-intercept changes
5) Teach different representations of data: physical points, graph and equations

# Chapter 1 – Linear Algebra & Ohm's Law

- ***Experiment:*** **Students measure different current outputs for different input voltages (batteries) and use the graph to find an unknown resistor**

- **Linear algebra correspondence:**
  - **Y = ( m * X ) + c → I = ( 1/R * V )**
  - **I is the current in the circuit**
  - **V is the voltage in the circuit**
  - **R is the resistance in the circuit**
  - **m = 1/R , is the slope**
  - **c = 0, is the y intercept**

# Outline

- Background: STEM curriculum for Algebra I
- **Design and Principles for an interactive tool**
- Evaluation and Optimizations
- Deployment and Analysis

# Curriculum Technical Requirements

- An interactive smart classroom portal where students:
  1. Enter their experimental data
  2. Visualize it on to a graph
  3. Generate regression lines and their equations
  4. Interactively manipulate slope and y-intercept
  5. Update axes to better understand negative slopes
  6. Save work and retrieve it later
  7. Collaborate and share data with other students
  8. Observe **instant feedback**

# Curriculum Technical Requirements Translated to Computer Systems Jargon

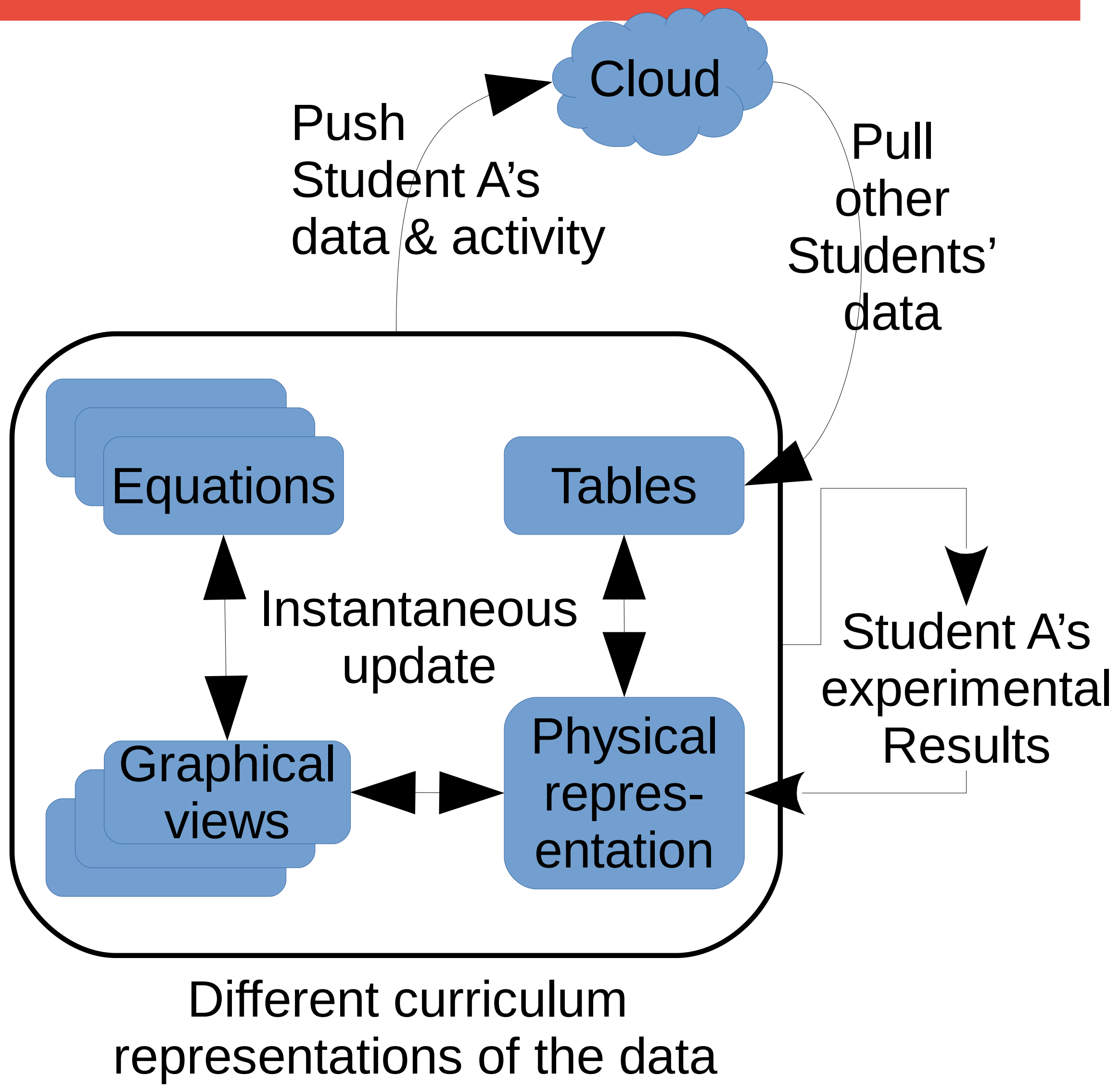
Response time below 50ms (20 frames per second)

- Data Input
- Data Visualization
- Interactive updates on the visualized graph
- Share data between users in real time
- Save data and session to continue later
- Integration with classroom management systems for easy adoption

- Demo

# System Design

- Design Principles:
  - – Client-side scripting
  - – Curriculum and system co-design
  - – Asynchronous transfer between Client and Server
  - – Integration with classroom management systems



# Curriculum and System Co-Design

## Response time challenges

- Graphing tools like excel does not provide interactive charts
- Primary focus should be curriculum demands more than other features
- Co-design a system with graphing and Interactivity at core that guarantees minimal update times

## Client-side Scripting

– JavaScript enabled client that uses D3.js visualization library

– Client visualization enables client to graph data without sending requests to

server

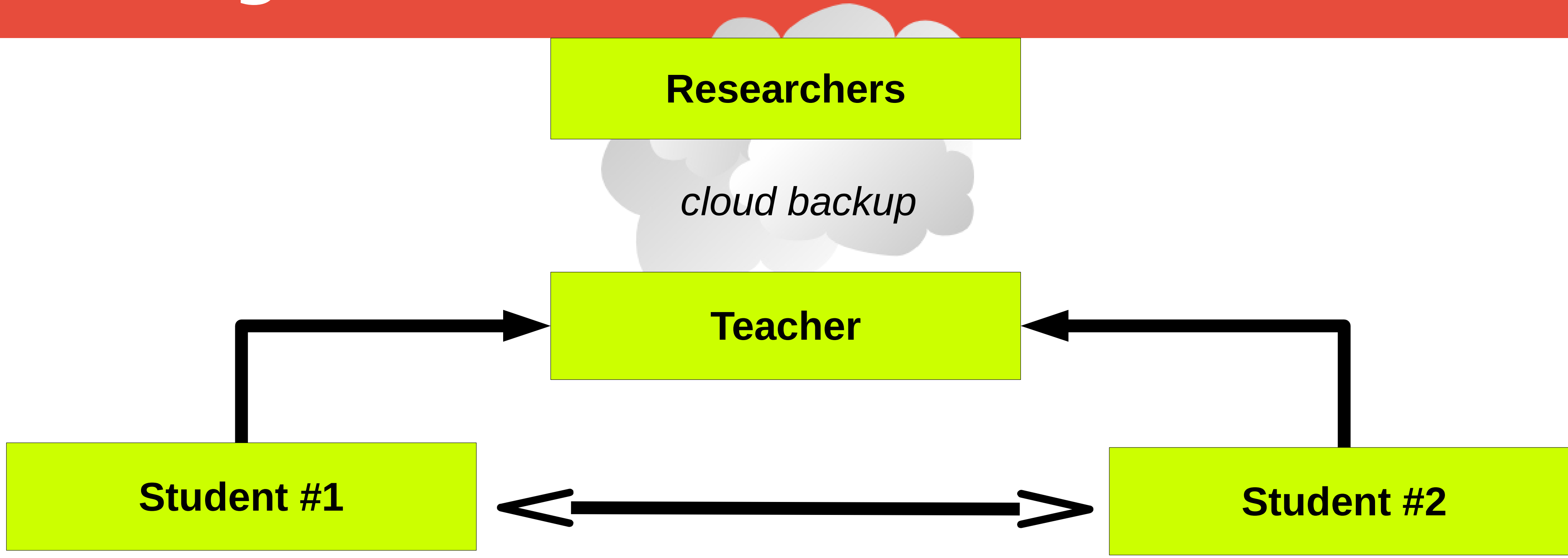
– No RTT latency in visualization

– All updates to graphs are processed by client in few milliseconds

# Curriculum and System Co-Design

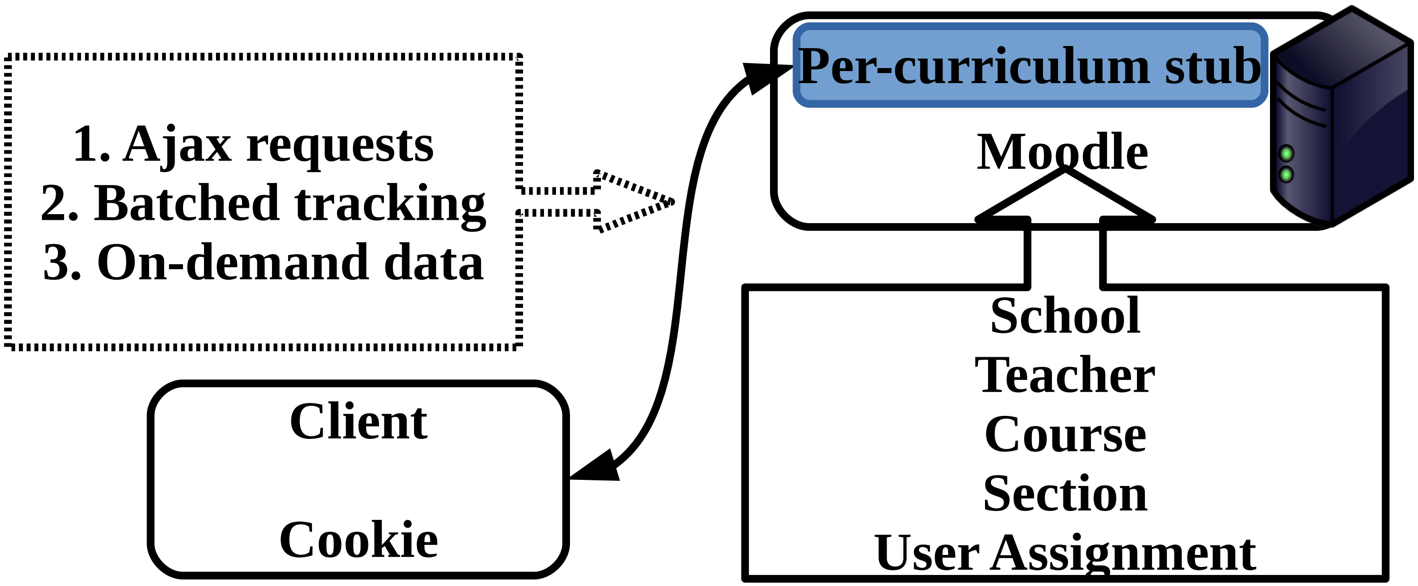
| Curriculum Requirements | Number of clicks in | |
|---|---|---|
| | Excel | Our system |
| Input n points | 2n | 2n |
| Insert 1 point in between | 3 | 1 |
| Delete 1 point | 3 | 1 |
| Graph points to a line | 6 | 1 |
| View equation of line | 3 | 0 |
| Change slope | - | 1 |
| Change intercept | - | 1 |
| Update point and update graph | 0 | 0 |
| Add a new line to existing graph | 12 | 1 |
| Delete an existing line from graph | 3 | 1 |
| Update 1 extreme axis | 4 | 2 |
| Update all 4 axis | 9 | 6 |

# Sharing Data Between Students/Teachers




| Problems that slow down response time | Solution |
|---|---|
| Does every update require a page reload? | AJAX (Asynchronous Javascript and XML) can send and receive data without reload |
| Must capture ALL interactions from keystrokes to mouse movement. | Batch user interaction and transmit every 60 seconds |
| How frequently are shared tables updated? How is data kept consistent? | Add a button. Refresh tables and pull data on demand. |

# Sharing Data Between Students/Teachers

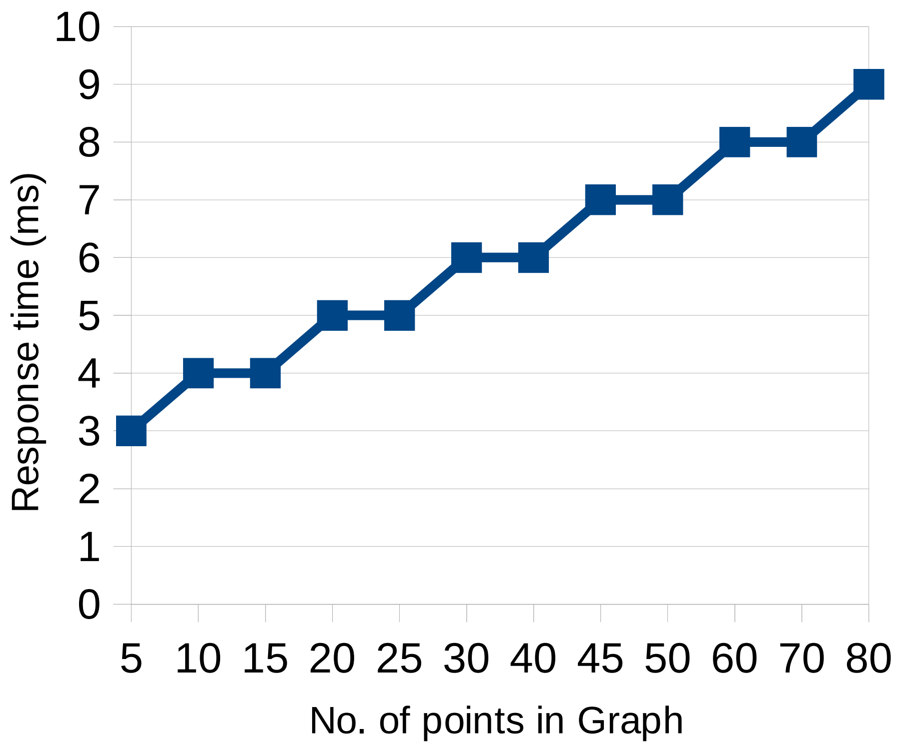



| Problems that slow down response time | Solution |
|---|---|
| Does every update require a page reload? | AJAX (Asynchronous Javascript and XML) can send and receive data without reload |
| Must capture ALL interactions from keystrokes to mouse movement. | Batch user interaction and transmit every 60 seconds |
| How frequently are shared tables updated? How is data kept consistent? | Add a button. Refresh tables and pull data on demand. |

# Outline

- Background: STEM curriculum for Algebra I
- Design and Principles for an interactive tool
- **Evaluation and Deployment**

# Performance Analysis



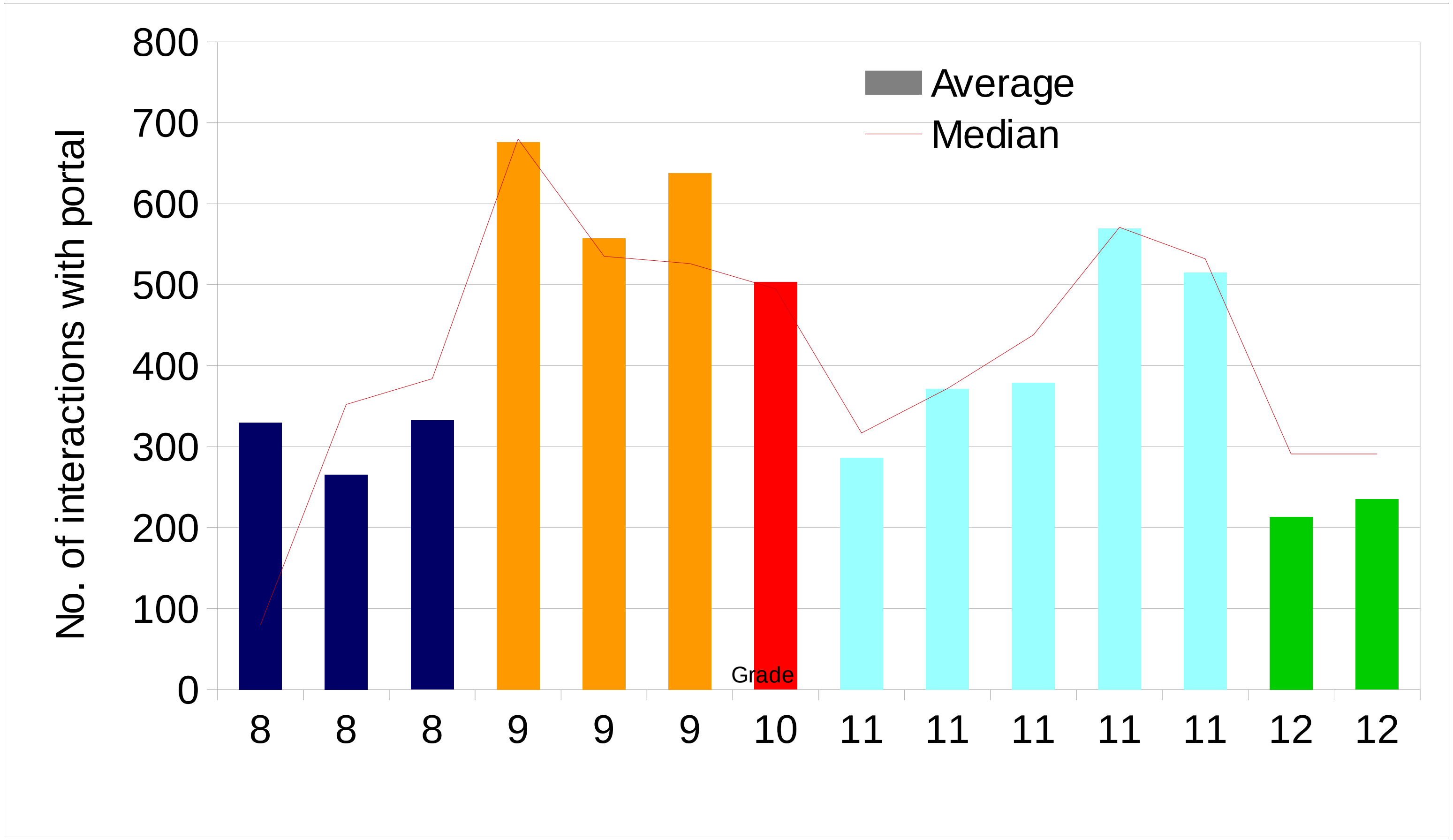

- **Avg. response time below 10 ms**
  - Graph creation
  - Update equation
  - Add/remove points
- **Tail latency**
  - Slowest requests were initial page loads
  - 99% below 100 ms
  - 99.9% below 150 ms

# Deployment and Analysis

- In 2016, piloted with 20 teachers
- Updated curriculum and tool with feedback
- More than 80% adopted our portal for their classes
- In 2017, over 1300 students in 5 schools in Columbus

# Deployment and Analysis

- Number of interactions across different classes



# Conclusion

- Developed an engineering driven curriculum
- Uncovered principles for interactive curriculum-aware smart classrooms
- Deployed to over 1300 students; Evaluated performance
- Future work: Add more chapters and deep IoT support