

# GPUUnit

Daniel Bagnell  
Jason Economou  
Rajkumar Jayachandran  
Tim McJilton  
Gabriel Schwartz  
Andrew Sherman

Advisor: Prof. Jeremy Johnson  
Stakeholders: Prof. Steve McMillan  
Alfred Whitehead  
The Leiden Observatory

January 25, 2011

# Overview

## Introduction

## Purpose

- Purpose of GPUnit

- Target Audiences

## Components and Features

- Experiment Management

- Diagnostics

- Command Line Interface

- Parallelism and Distribution

## User Interface

- Cluster Control

- Cluster Control

- Module Specification

## Final Goals

# Motivation

- ▶ Astrophysics researchers need to simulate star clusters and galaxies.
- ▶ Every star pulls on all of the others:  $O(n^2)$  for the naive case.
- ▶ Stars evolve over time, mass changes.

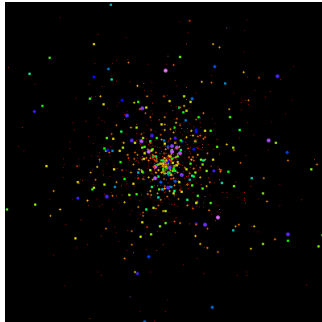


Figure: N-Body Simulation: 1024 Stars

# Astrophysical Multipurpose Software Environment (AMUSE)

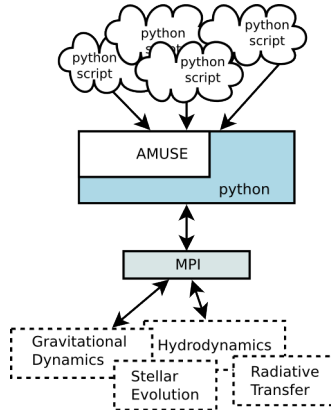


Figure: AMUSE Architecture

# State of AMUSE

- ▶ Currently used by researchers to run large-scale simulations.
- ▶ Scripts, diagnostics, logging are all written by hand.
- ▶ AMUSE API/programming knowledge is required to create experiments.

# Purpose of GPUnit

- ▶ Ease the use of AMUSE
- ▶ Create/Design/Modify experiments
- ▶ Select, configure, swap out modules and initial conditions

# Target Audiences

- ▶ Physics Students
- ▶ Observational Astrophysicists
- ▶ Theoretical Astrophysicists

# Experiment Management

- ▶ Experiments include a specification for:
  - ▶ Initial Conditions, Variables
  - ▶ Module Selection
  - ▶ Logging
  - ▶ Diagnostics
- ▶ Allows experiments to be repeated and shared.



# Diagnostics

- ▶ Diagnostics read the state of the system while it is computing, and output selected metrics.
- ▶ Features include:
  - ▶ Configurable output data formats
  - ▶ Sensible default diagnostics provided
  - ▶ An interface to create custom diagnostics
- ▶ Diagnostics can include a visual display of the system state.

# Command Line Interface

- ▶ The command line interface gives the user ability to:
  - ▶ Make small changes to the experiment before running
  - ▶ Run headless experiments
  - ▶ Control experiment distribution across a cluster
  - ▶ Easily repeat experiments

# Parallelism and Distribution

- ▶ Modules are independent, can run at the same time.
- ▶ No need to share data until the end of a time interval.
- ▶ Some exceptions if a special event occurs
- ▶ User interaction can be required.
- ▶ GUI will provide a display of all nodes on the cluster.
- ▶ Command line provides a list of free nodes if desired.

# Experiment Editor

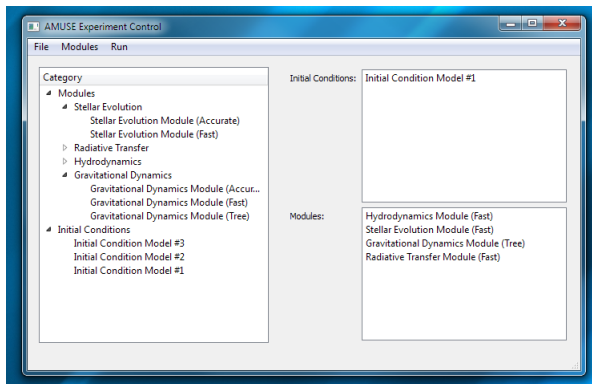


Figure: Experiment Editor

# Cluster Control

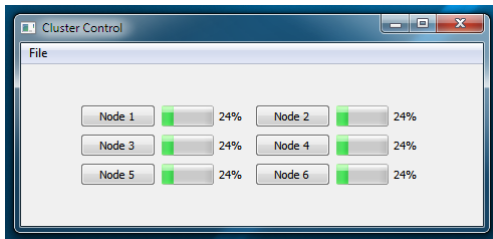


Figure: Cluster View

# Module Specification

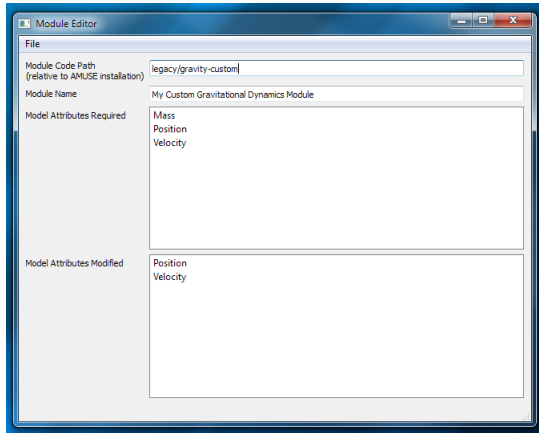


Figure: Module Editor

# Final Goals

- ▶ Integration into AMUSE
- ▶ Users can download AMUSE and create/run experiments right away.
- ▶ Open AMUSE up for use by a non-programmer audience.