SAND2015-7822PE

Marching Cubes Mantevo Miniapp

Steven Munn

Sandia National Laboratories

sjmunn@sandia.gov

September 14, 2015

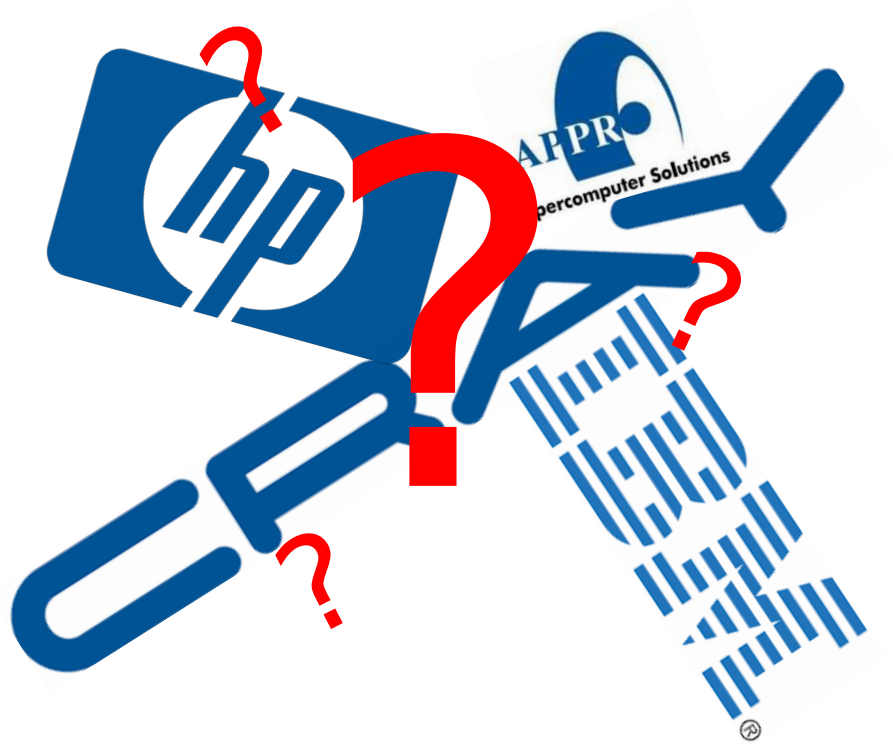
Overview

1. The Mantevo Project
   - Its Purpose
   - Its Solution
2. The Marching Cubes Miniapp
   - Overall Design
3. Miniapp Results
   - With the Stanford Bunny CT-scan

The Mantevo Project

Many HPC Hadware Vendors

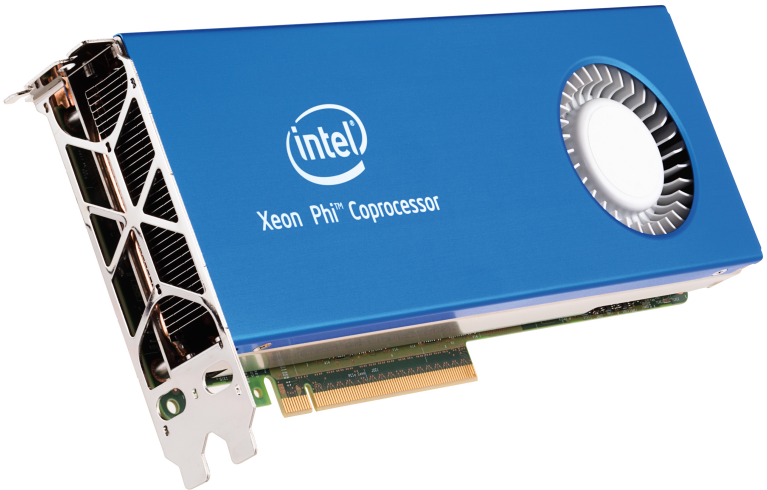
Suppose you want to run software on a supercomputer or cluster,



High Performance Computing hardware manufacturers each claim better results than their competition.

New Hardware Architectures

Or suppose you want to use new hardware architectures in your cluster



Goals for Mantevo

Goal 1

Provide tools to compare hardware solutions for a specific task.

Goal 2

Enable users to test adaptations to their algorithm.

Mantevo Miniapps

The Mantevo project's solution is to characterize performance with proxies called Miniapps.
A Miniapp comprises the most computationally expensive subroutines in a stand-alone software package.

Miniapp Guidelines

Mantevo miniapps must be:

Simple to build

They are self-contained and rely on a simple sequence of compiling commands.

Stable

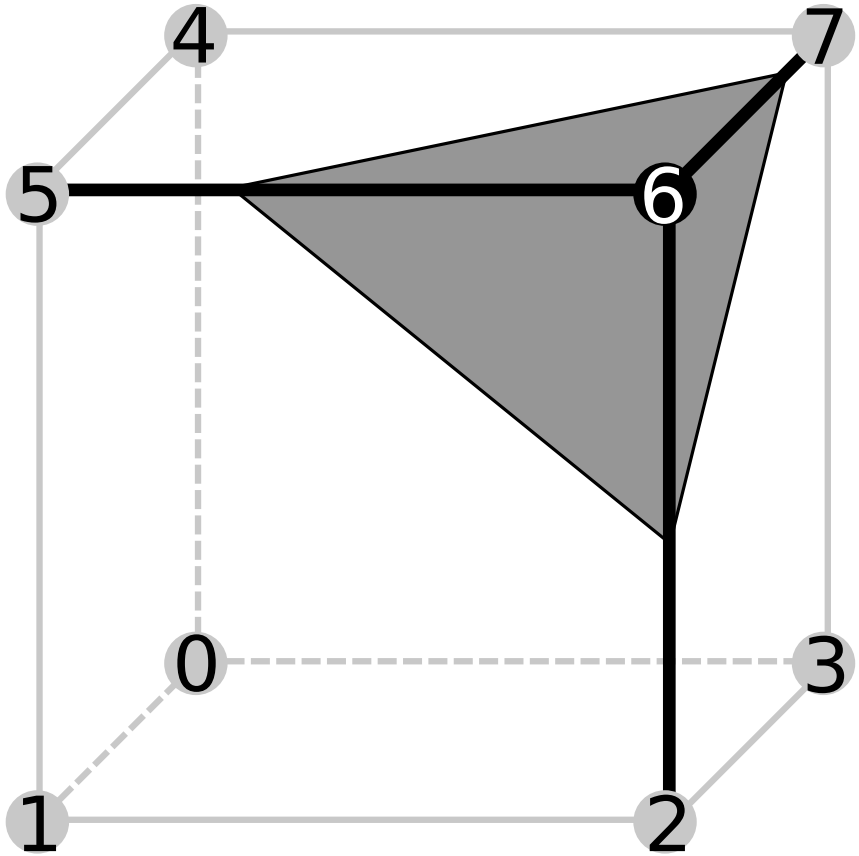
They use only the core, stable features of any programming language.

Easy to understand

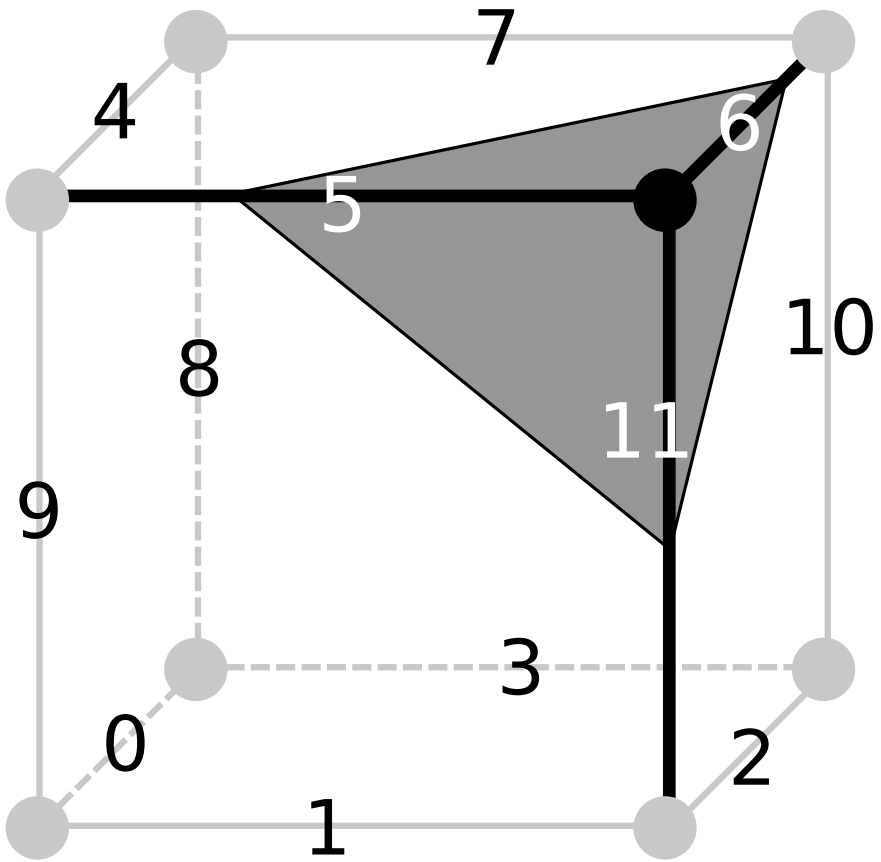
Users can easily apply them, modify them, or port them to different languages.

The Marching Cubes Miniapp

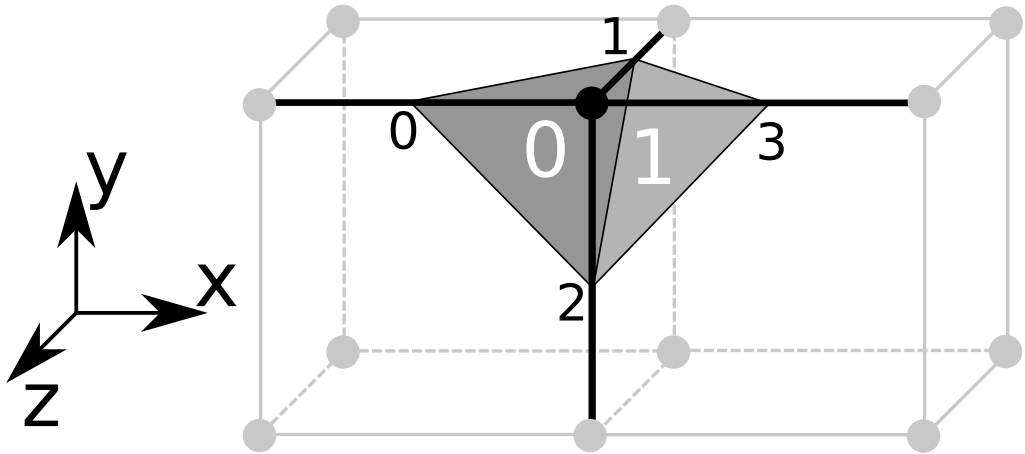
Algorithm



Algorithm



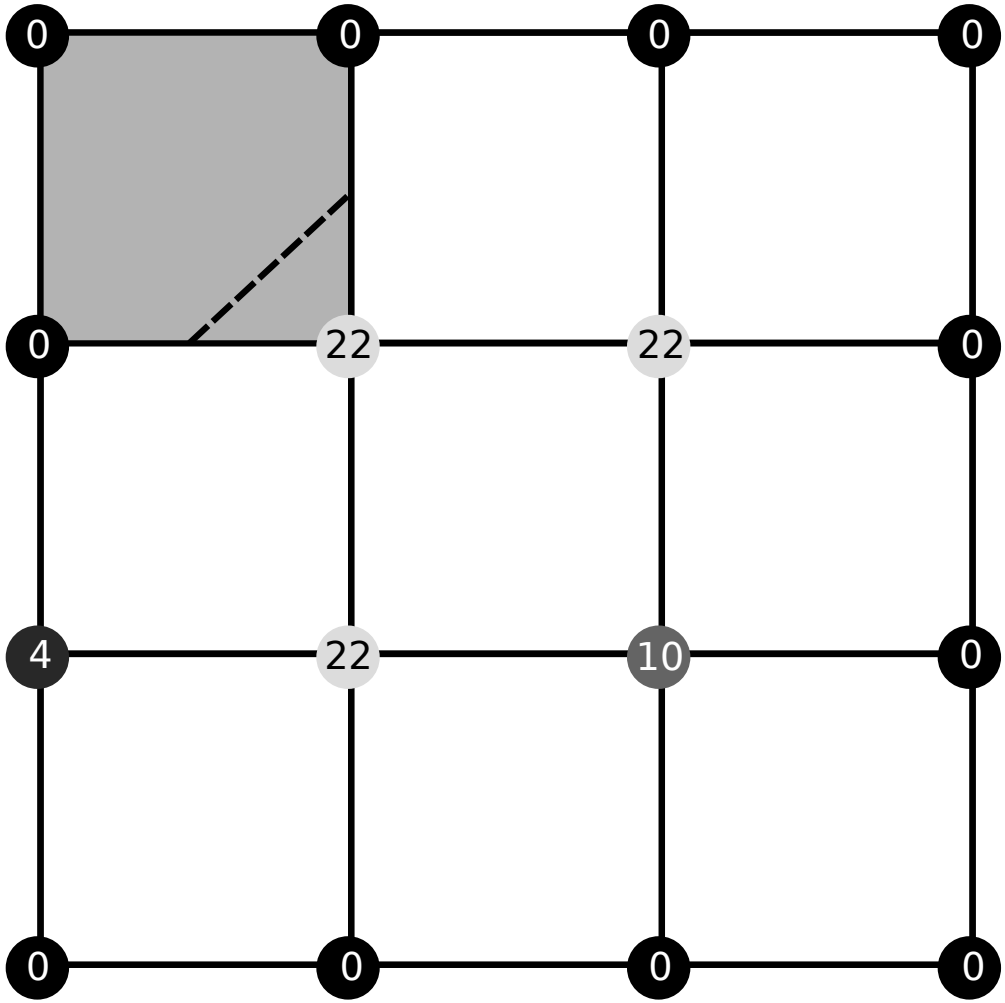
Algorithm



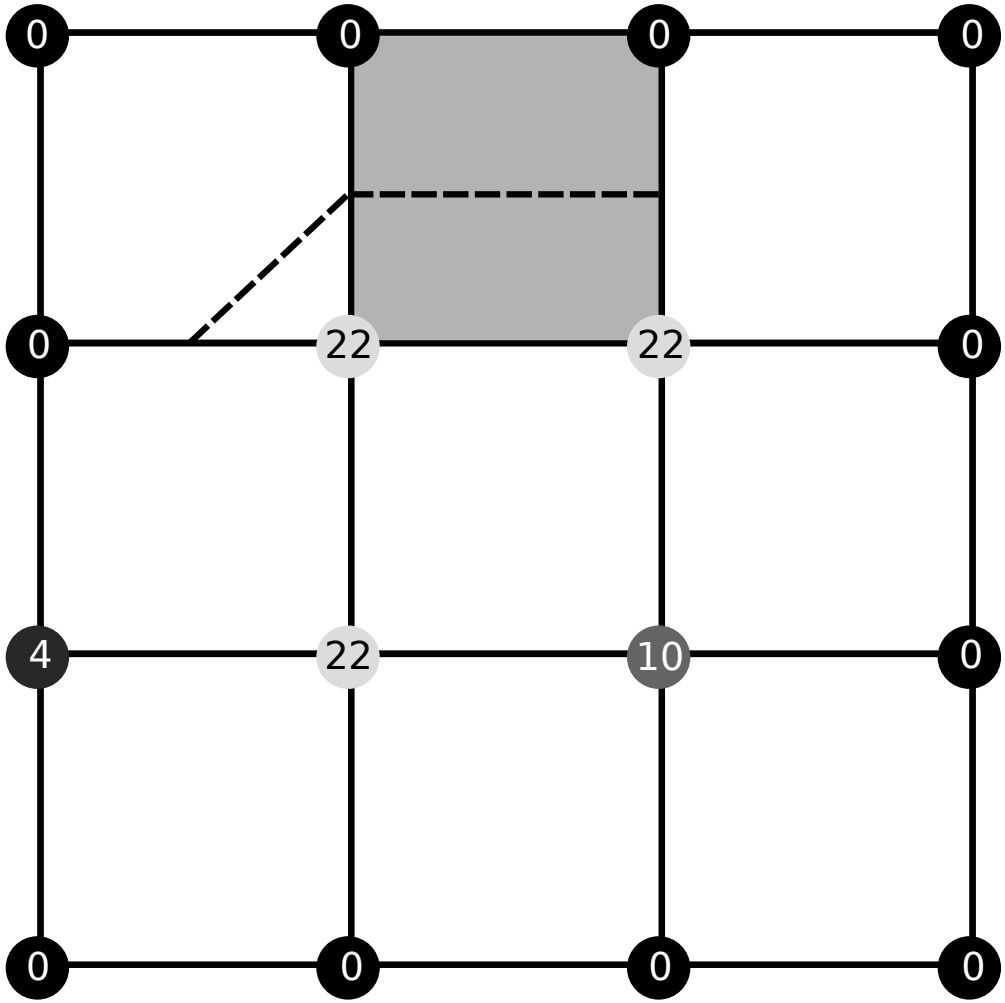
Mesh	triangle0	triangle1
	0	1
	1	2
	2	3

Point Index	0	1	2	3
x	0.5	1	1	1.5
y	1	1	0.5	1
z	1	0.5	1	1

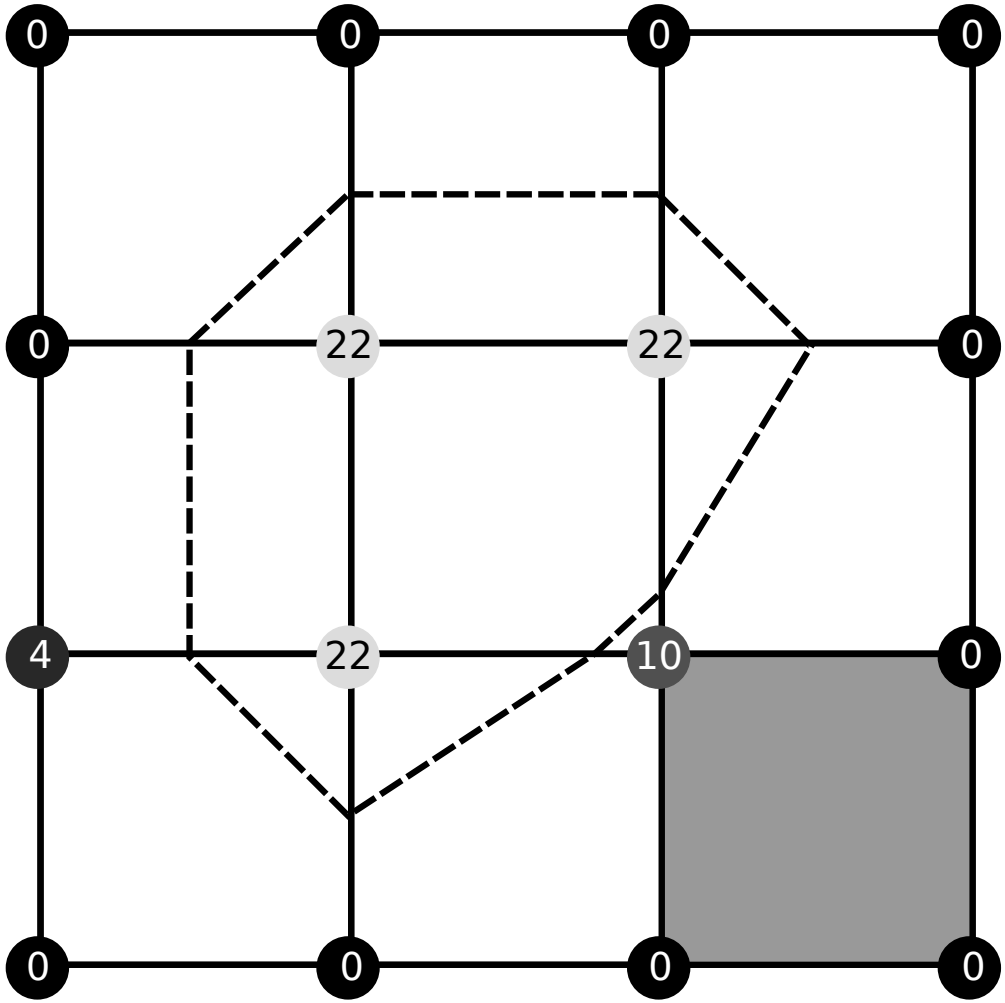
Marching Squares



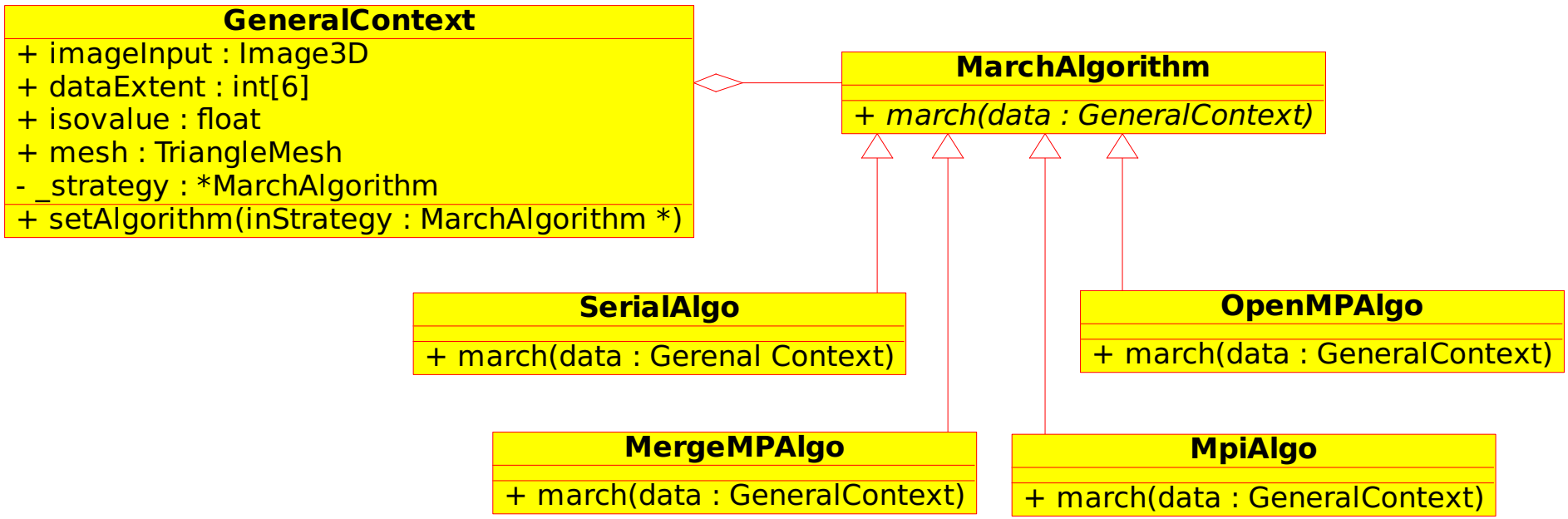
Marching Squares



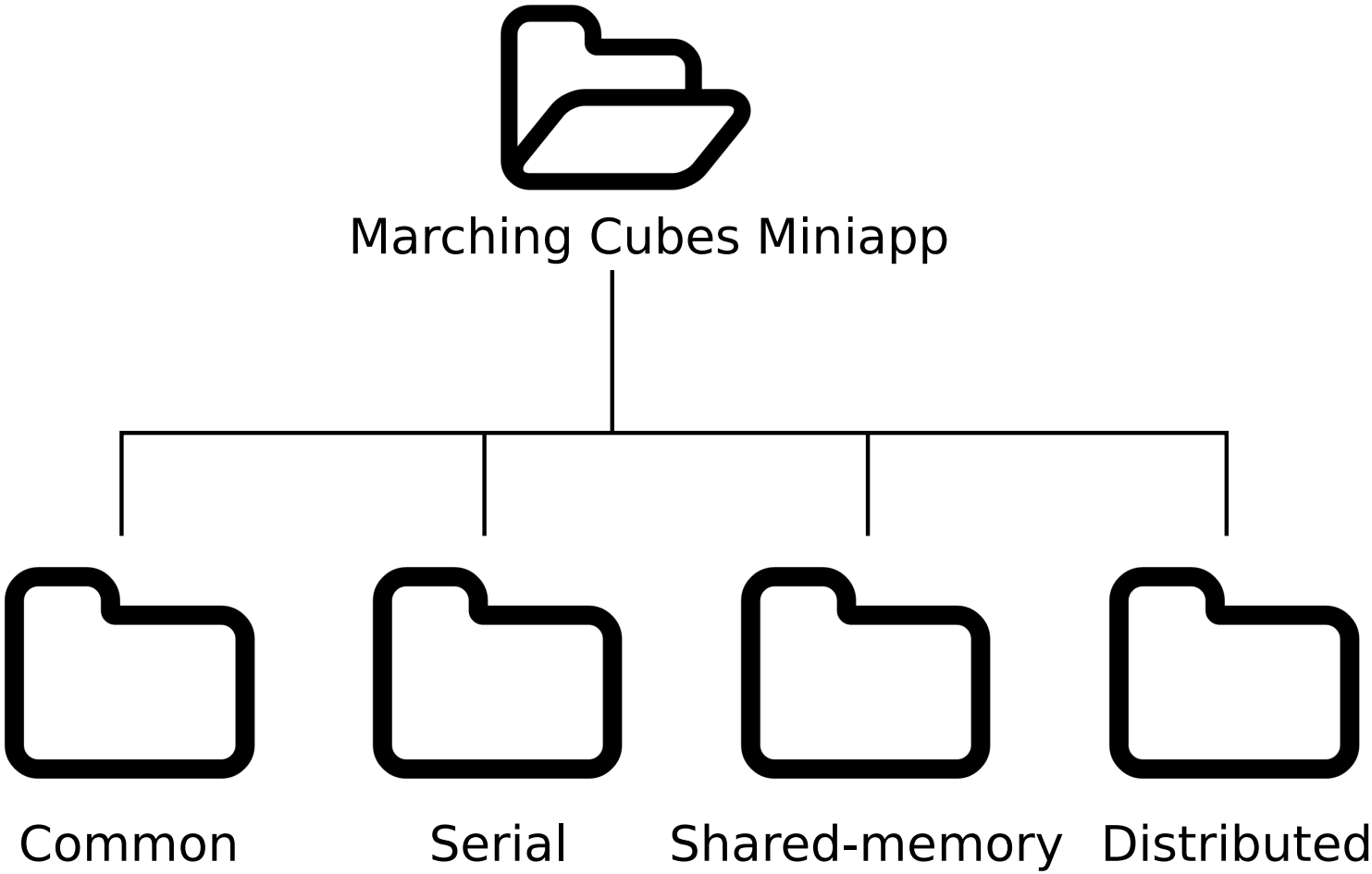
Marching Squares



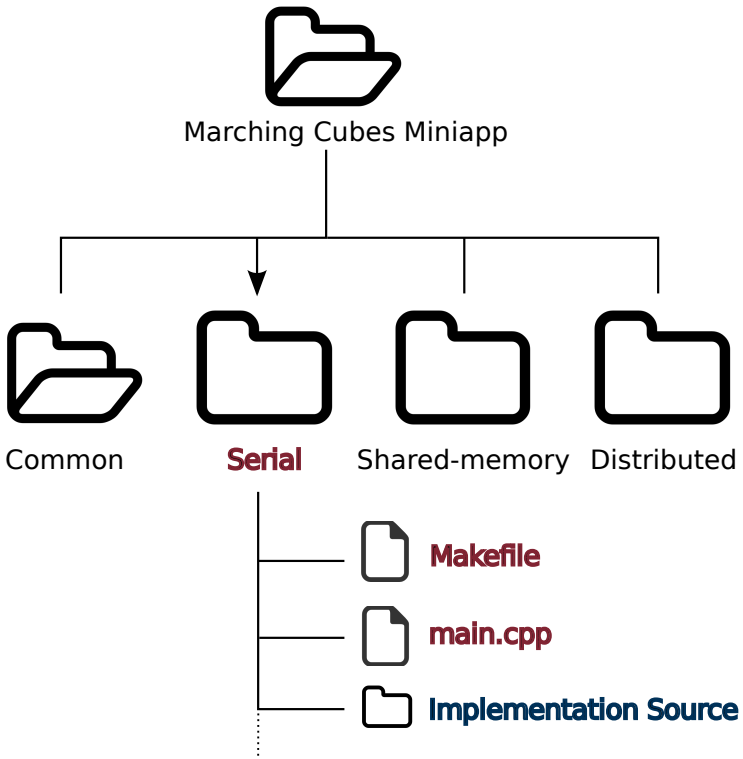
The Strategy Design Pattern



Seperate Implementations by Directory



Inside a Directory



Build Commands

Example (Build Commands)

```
cd common
make all
cd ../serial
make all
```

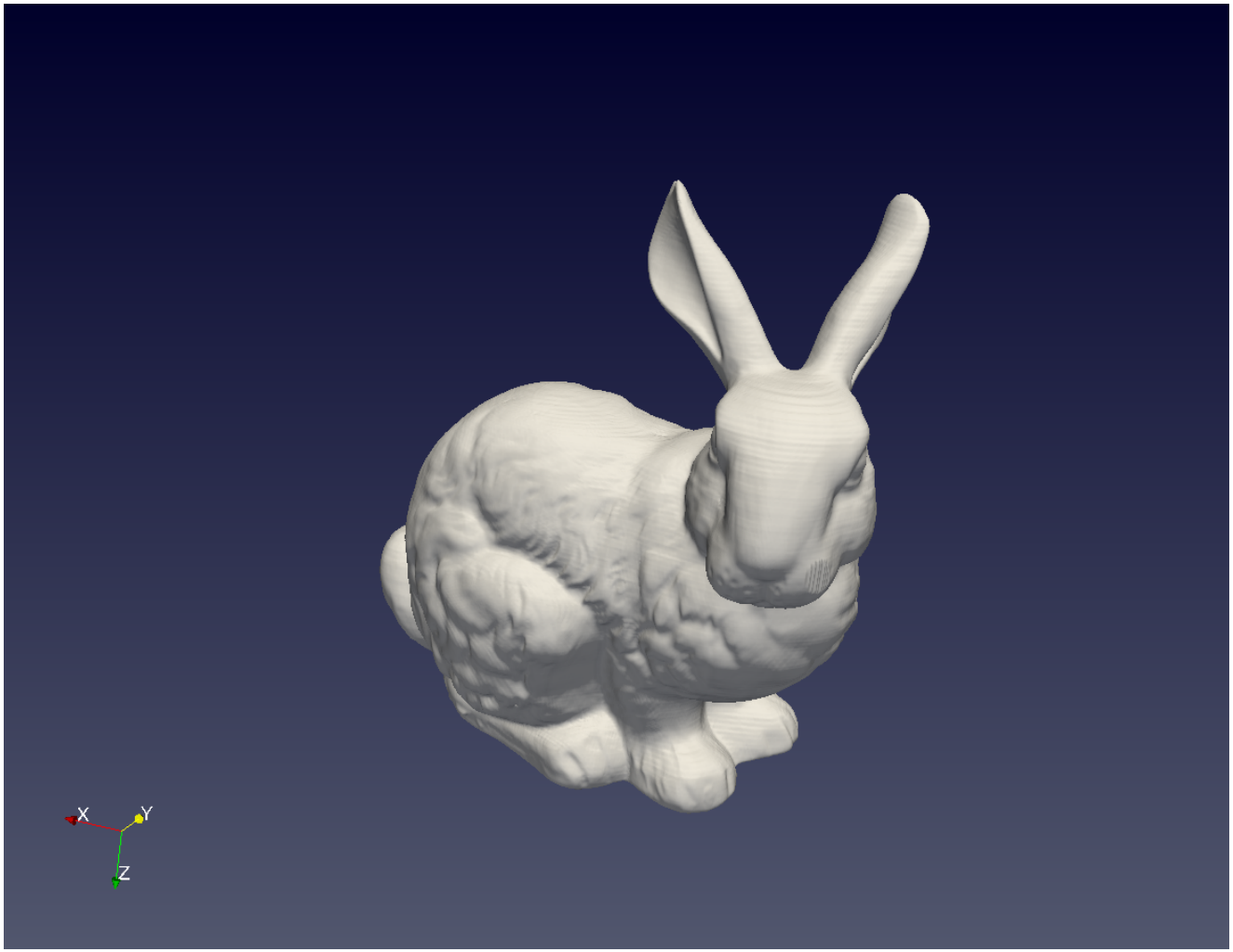
Start Marching

Run Commands

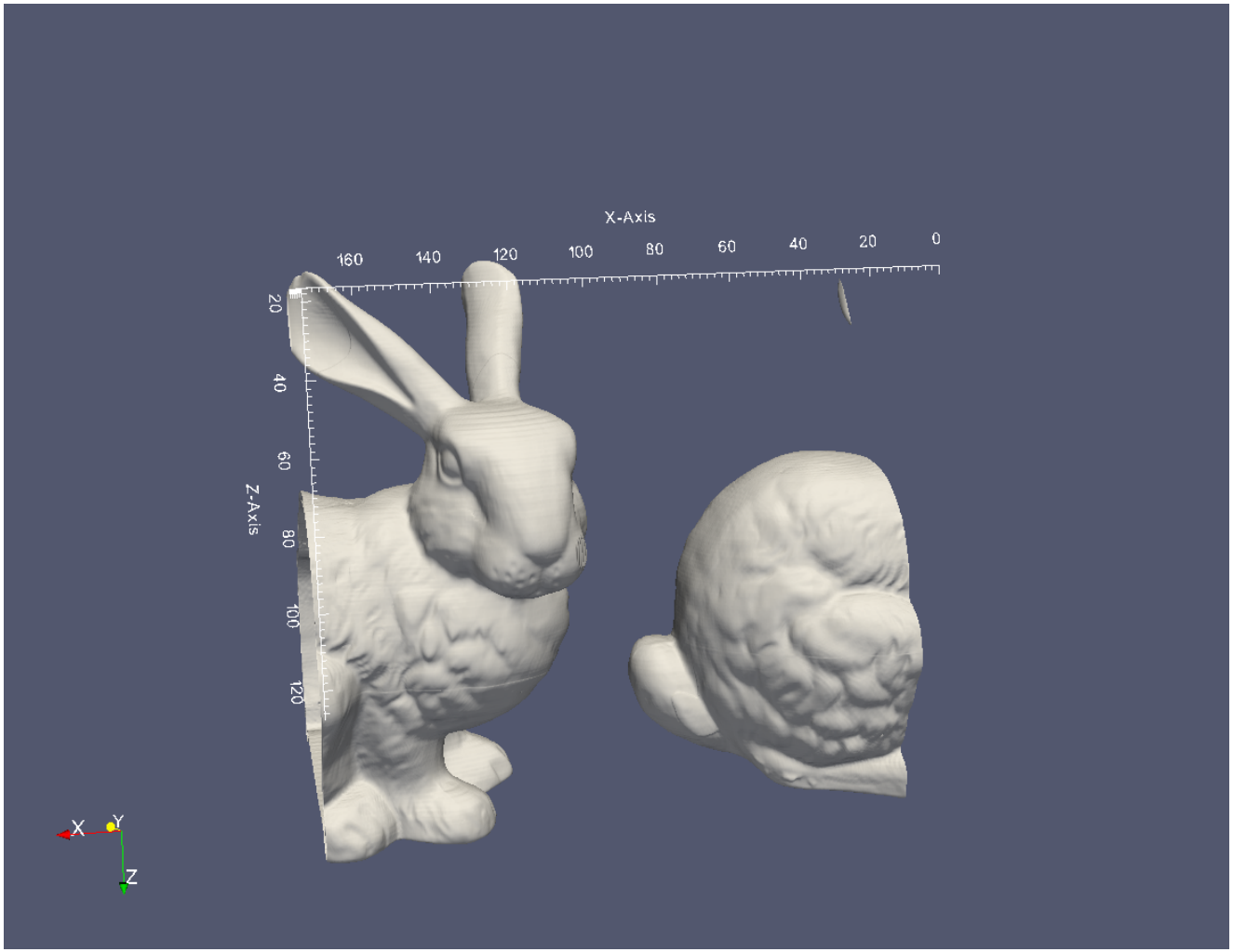
```
./MarchingCubes <in file> <out file> <isovalue>

./MarchingCubes <header file> <data file> <out file> <isovalue>
```

Miniapp Results

The Stanford Bunny CT scan



The Stanford Bunny CT scan



Serial Implementation

Example (YAML output)

```
Mini-Application Name: Marching Cubes
Mini-Application Version: 0.1
Volume image data file path: ../Data/inHeader.vtk
File x-dimension: 512
File y-dimension: 512
File z-dimension: 361
Number of points in image volume: 94633984
Marching cubes algorithm: SERIAL
Total Program CPU Time (clicks): 1.79e+06
Total Program CPU Time (seconds): 1.79
Total Program WALL Time (seconds): 1.78487
```

Implementation Performances

Implementation Algorithm	Wall Time	CPU Time
Serial	1.79	1.79
OpenMP	0.72	2.29
MergeMP	0.89	2.94

Table: Implementation runtimes on a workstation

Supernova Data

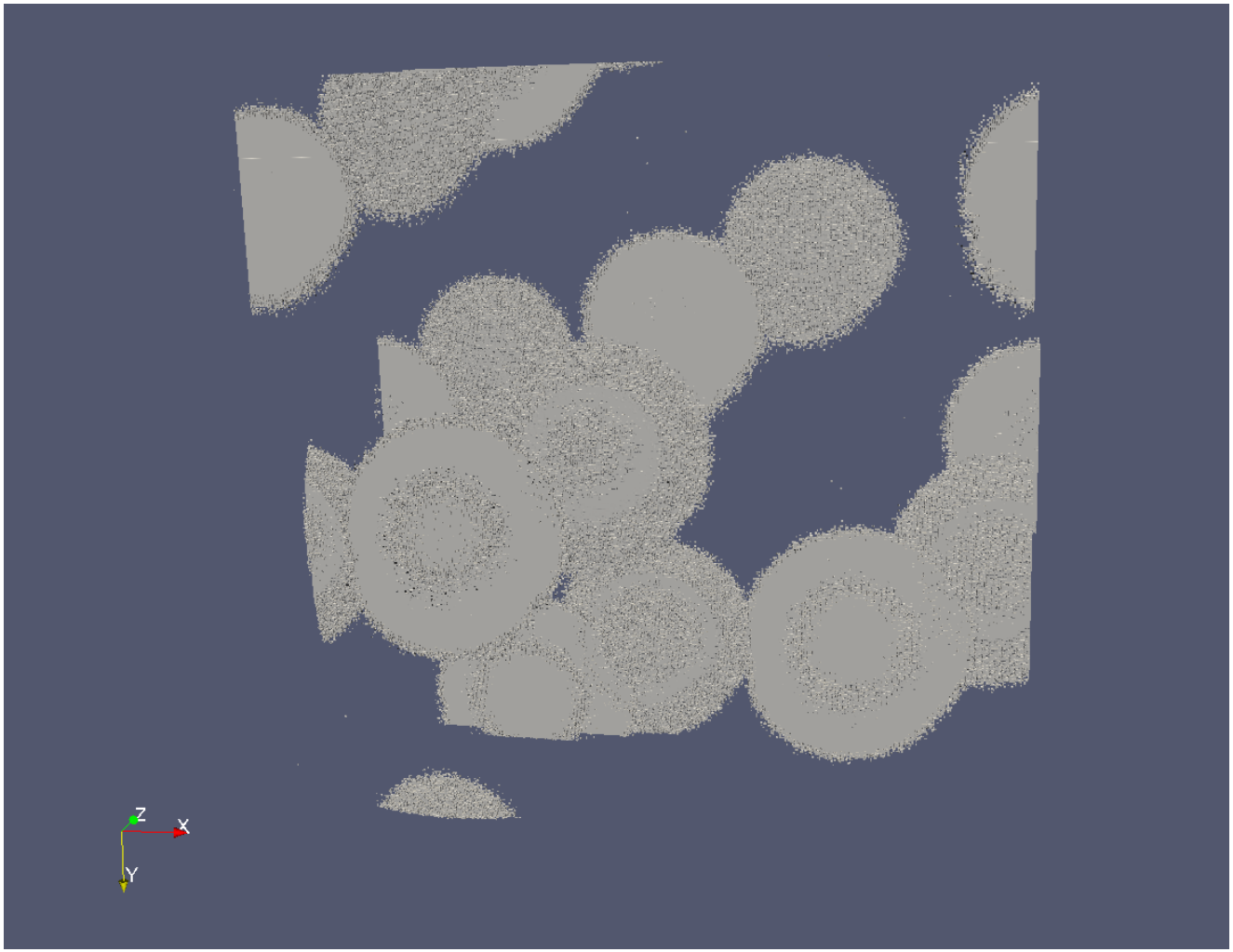
HPC-scale Input

A 150 GB image volume of a supernova.

Be Aware

The code still has major issues running at this scale. Results are to be taken with a grain of salt.

A Collection of Nova Data Slices



Large-scale Perfomance

Platform	Wall Time	CPU Time
Shannon	45.5	775
Shepard	297	2704

Table: Latest runtimes with current settings. This is not a fair comparison.

Funding Acklowledgement

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000.