U.S. Army Research, Development and Engineering Command

Brown Deer Technology

High Performance Java

TECHNOLOGY DRIVEN. WARFIGHTER FOCUSED.

Jordan J. Ruloff, James A. Ross, David A. Richie,
Song J. Park, Dale R. Shires, Brian J. Henz

September 11, 2012

Schedule

- Introduction
- Context
- Java
- Aparapi
- Conclusion
- Questions

Software Goals:

Introduction

Software Goals:

- Compatibility

Introduction

Software Goals:

- Compatibility
- Reliability

Software Goals:

- Compatibility
- Reliability
- Fault-Tolerance

Introduction

Software Goals:

- Compatibility
- Reliability
- Fault-Tolerance
- Security

Software Goals:

- Compatibility
- Reliability
- Fault-Tolerance
- Security
- Reusability

Software Goals:

- Compatibility
- Reliability
- Fault-Tolerance
- Security
- Reusability
- Usability

Introduction

Introduction

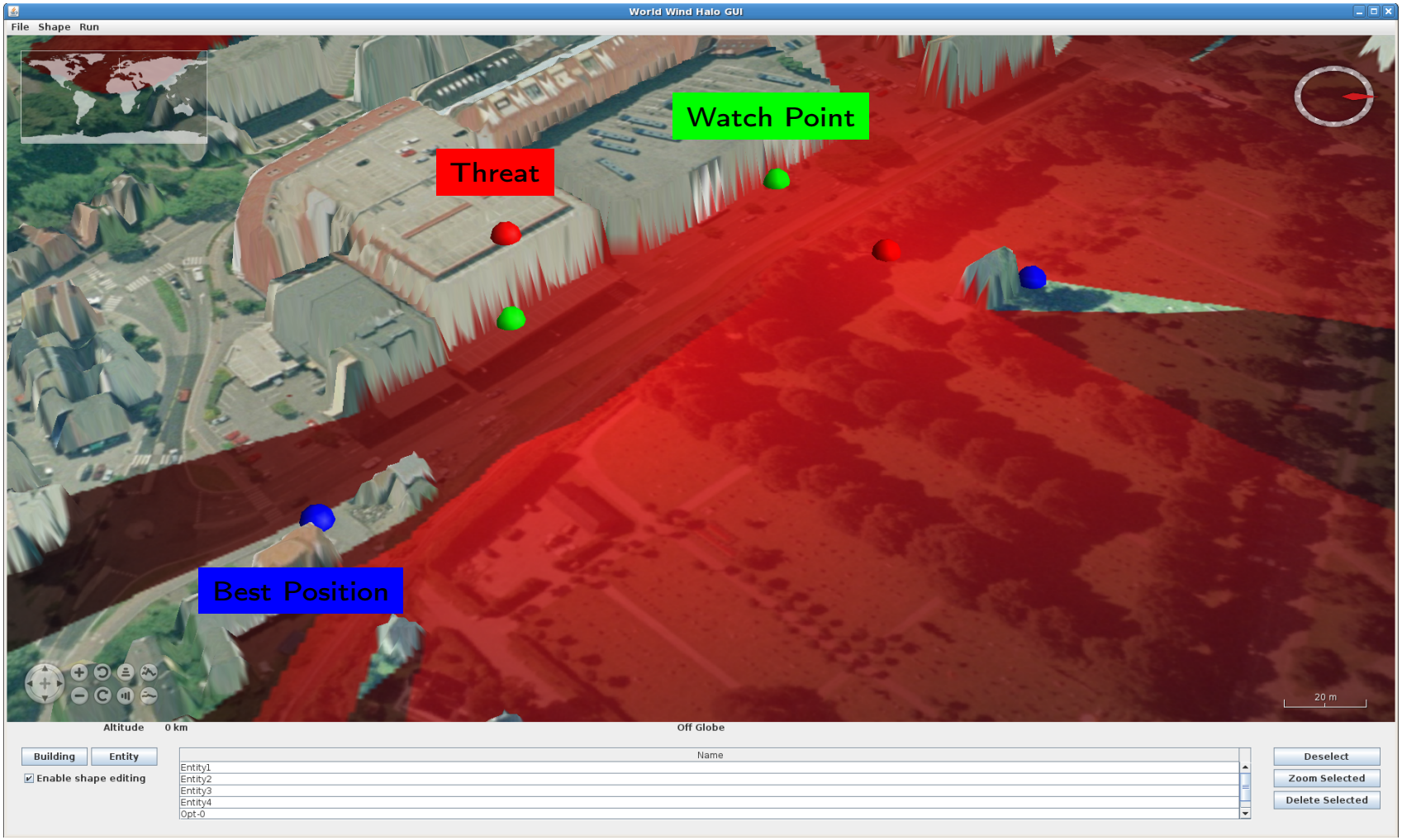
"For over a decade prophets have voiced the contention that the organization of a single computer has reached its limits and that truly significant advances can be made only by interconnection of a multiplicity of computers." - Gene Amdahl

Introduction

"For over a decade prophets have voiced the contention that the organization of a single computer has reached its limits and that truly significant advances can be made only by interconnection of a multiplicity of computers." - Gene Amdahl

"We stand at the threshold of a many core world. The hardware community is ready to cross this threshold. The parallel software community is not." - Tim Mattson

Context


World Wind Halo GUI
File Shape Run
Watch Point
Threat
Best Position
20 m
Altitude 0 km
Off Globe
Building
Entity
Enable shape editing
Name
Entity1
Entity2
Entity3
Entity4
Opt-0
Deselect
Zoom Selected
Delete Selected

Java Threading

- Thread
  - Java 5
  - 1 Runnable Task
- ThreadPoolExecutor
  - Java 5
  - 1 Concurrent Queue

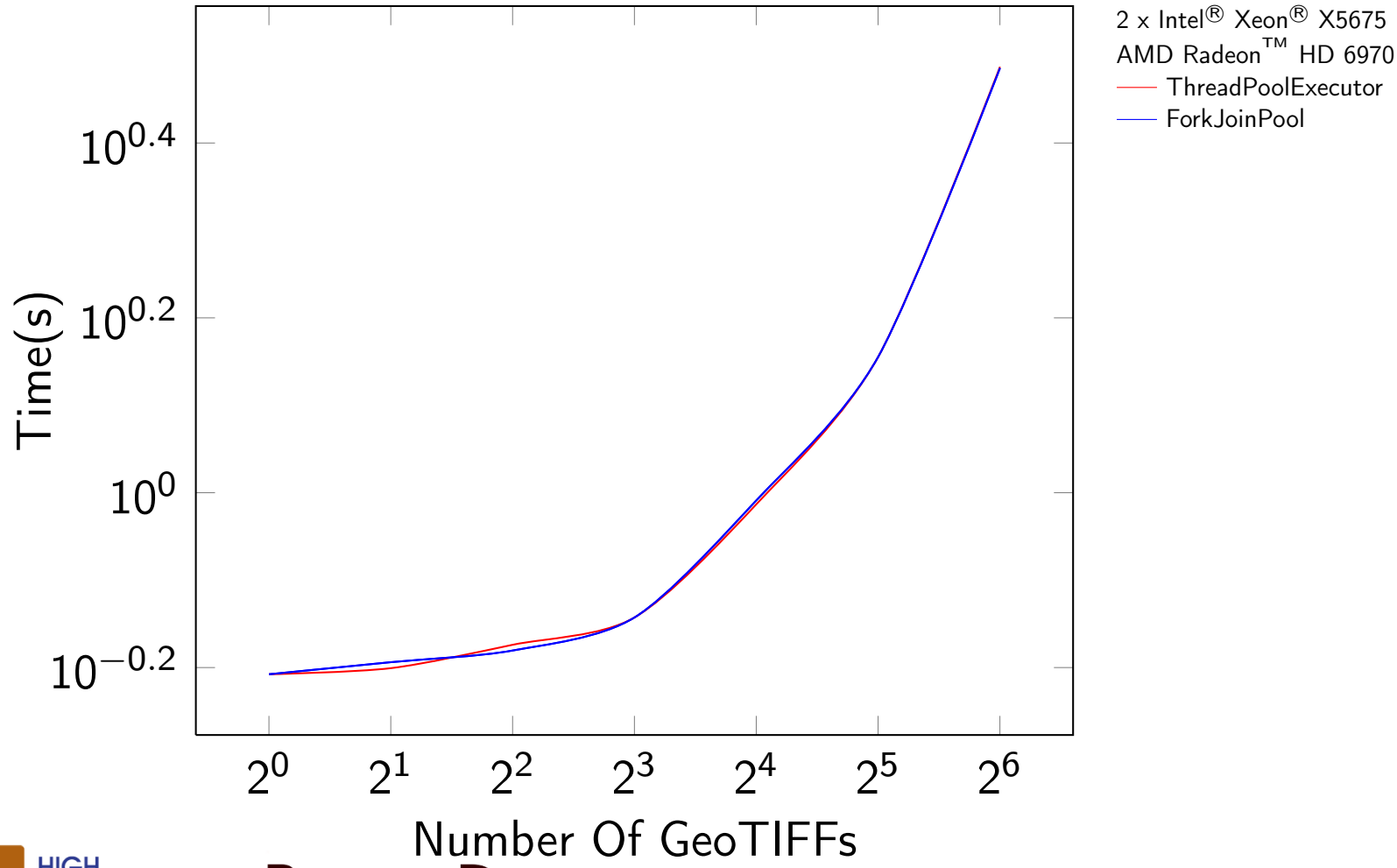
Java Threading

- Thread
  - Java 5
  - 1 Runnable Task
- ThreadPoolExecutor
  - Java 5
  - 1 Concurrent Queue
- ForkJoinPool
  - Java 7
  - N Concurrent Queues
  - Work-Stealing

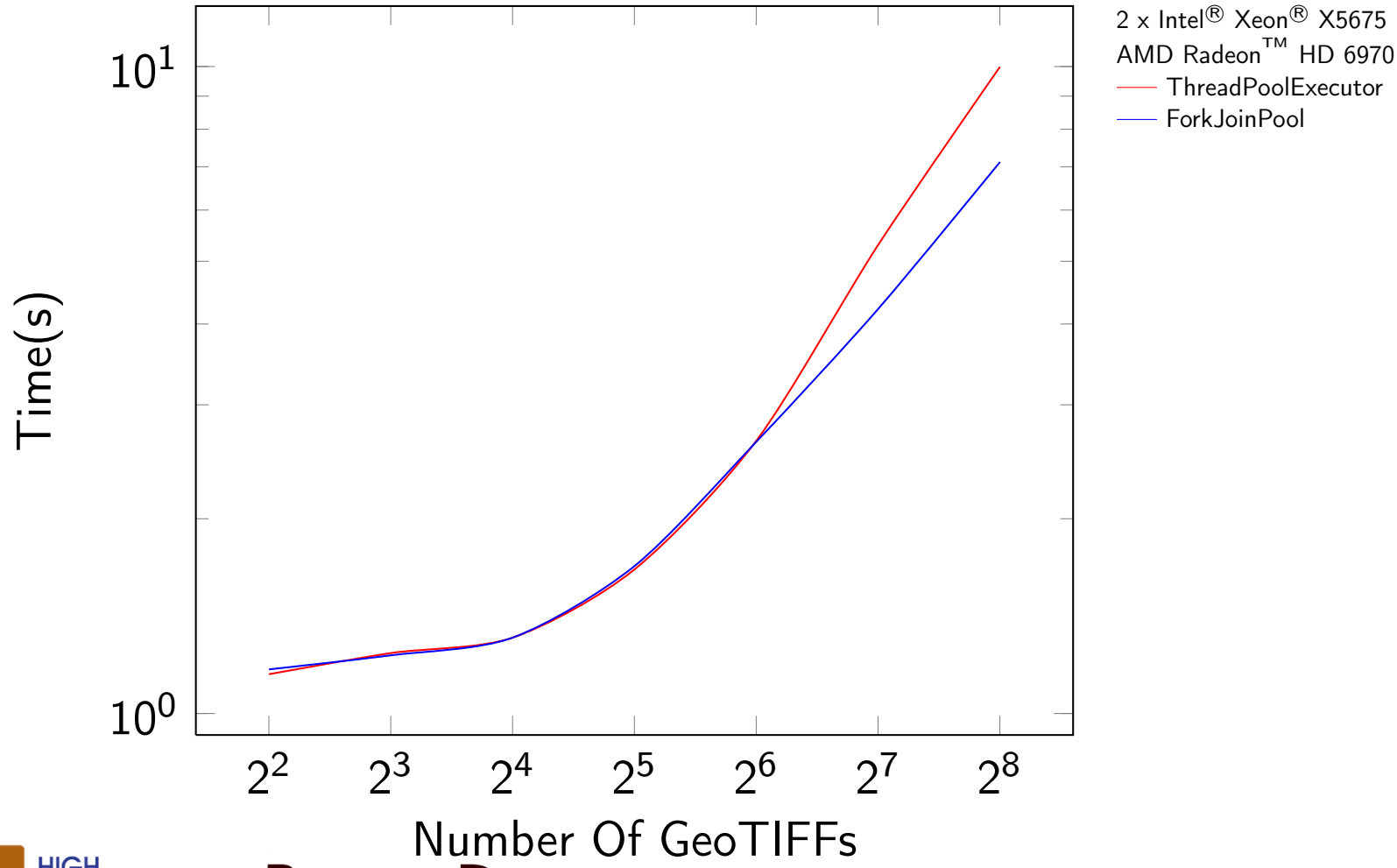
Java Benchmarks

- Load A Set Of GeoTIFFs Into A Usable Data Format
- NASA WorldWind Java Data Formats:
  - ElevationModel
  - SurfaceImage
- Performance Comparison Between ThreadPoolExecutor And ForkJoinPool

Execution Time For Loading A GeoTIFF Multiple Times



Execution Time For Loading Multiple GeoTIFFs Multiple Times



HIGH PERFORMANCE TECHNOLOGIES GROUP

Brown Deer Technology

TECHNOLOGY DRIVEN. WARFIGHTER FOCUSED.

Hardware

- Central Processing Units (CPUs)
- Graphics Processing Units (GPUs)
- Field-Programmable Gate Arrays (FPGAs)
- Digital Signal Processors (DSPs)
- Microcontrollers

Java Libraries

- JCuda
- jocl.org JOCL
- JogAmp.org JOCL
- JavaCL
- LWJGL
- Aparapi
- Rootbeer

Java Libraries

- JCuda
- jocl.org JOCL
- JogAmp.org JOCL
- JavaCL
- LWJGL
- Aparapi
- Rootbeer

Java Libraries

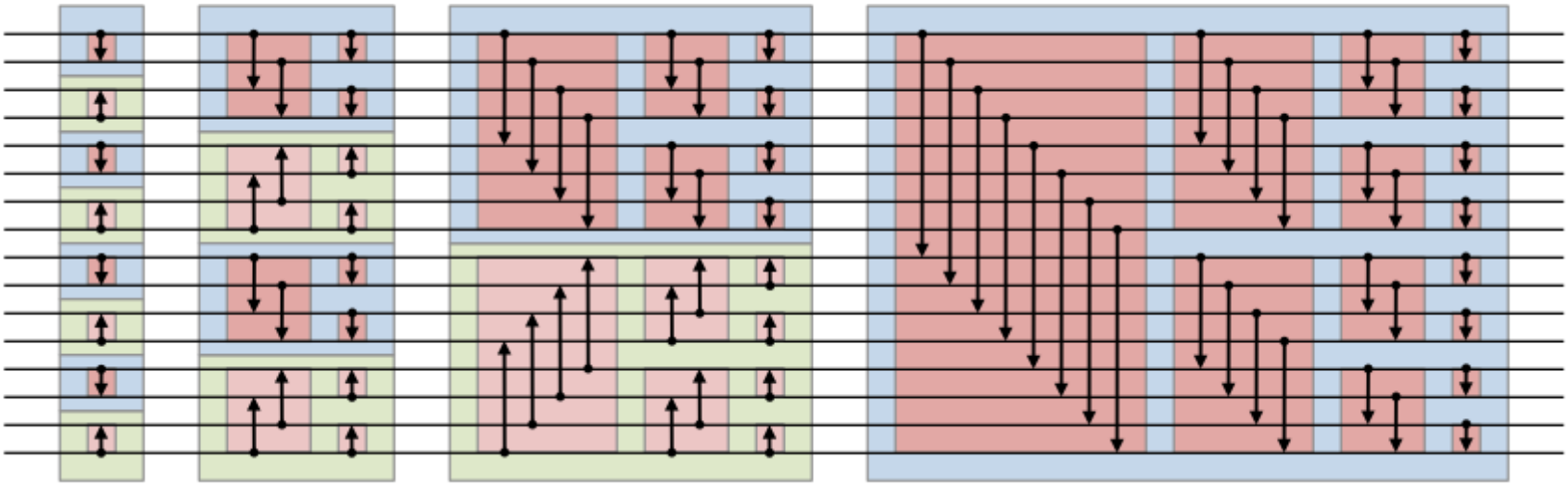
- JCuda
- jocl.org JOCL
- JogAmp.org JOCL
- JavaCL
- LWJGL
- Aparapi
- Rootbeer

Java Libraries

- JCuda
- jocl.org JOCL
- JogAmp.org JOCL
- JavaCL
- LWJGL
- Aparapi
- Rootbeer

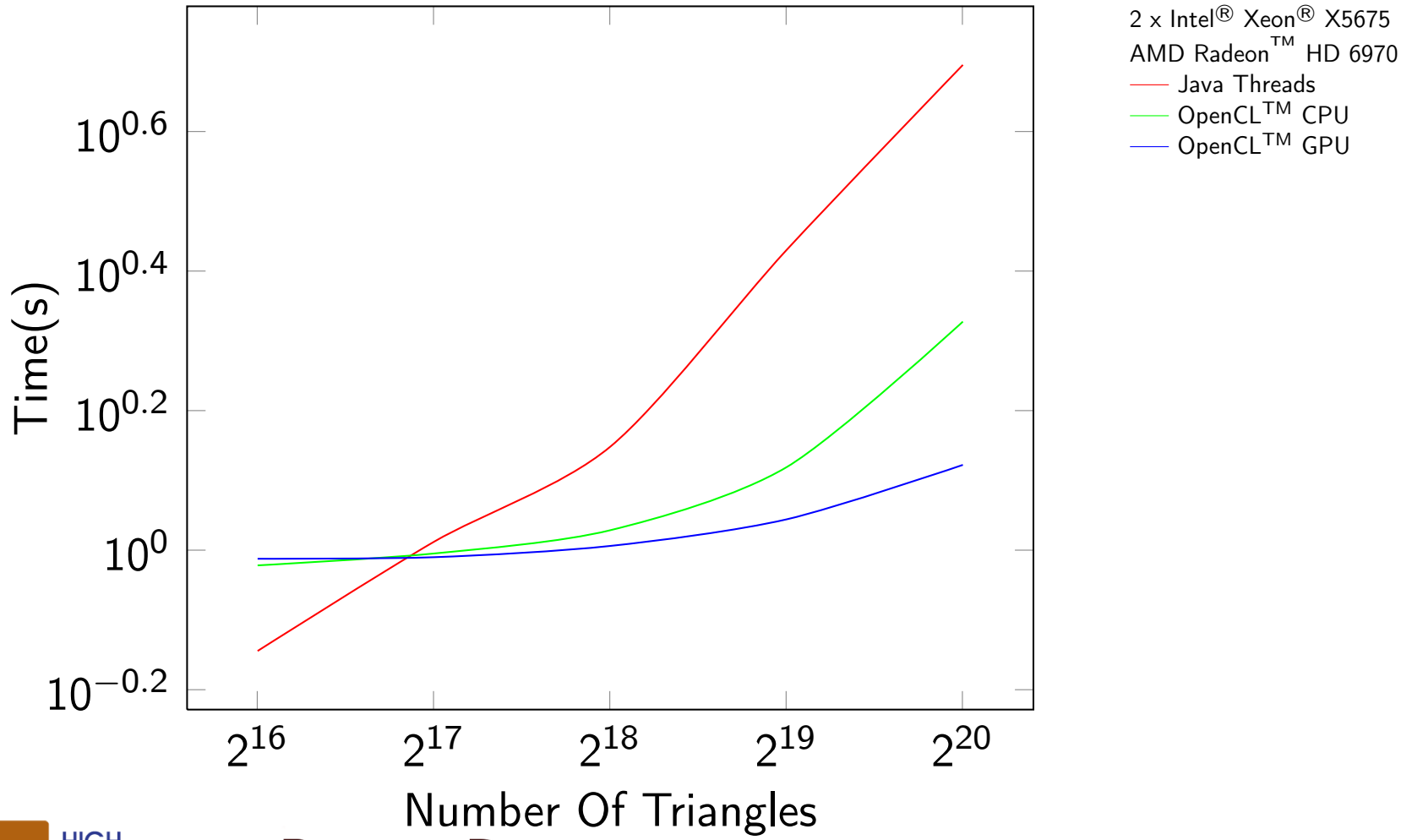
Preparation Kernels

- Bounding Box Construction
  - $O(1)$ Parallel Time
- Bitonic Sort
  - $O(log^2(n))$ Parallel Time



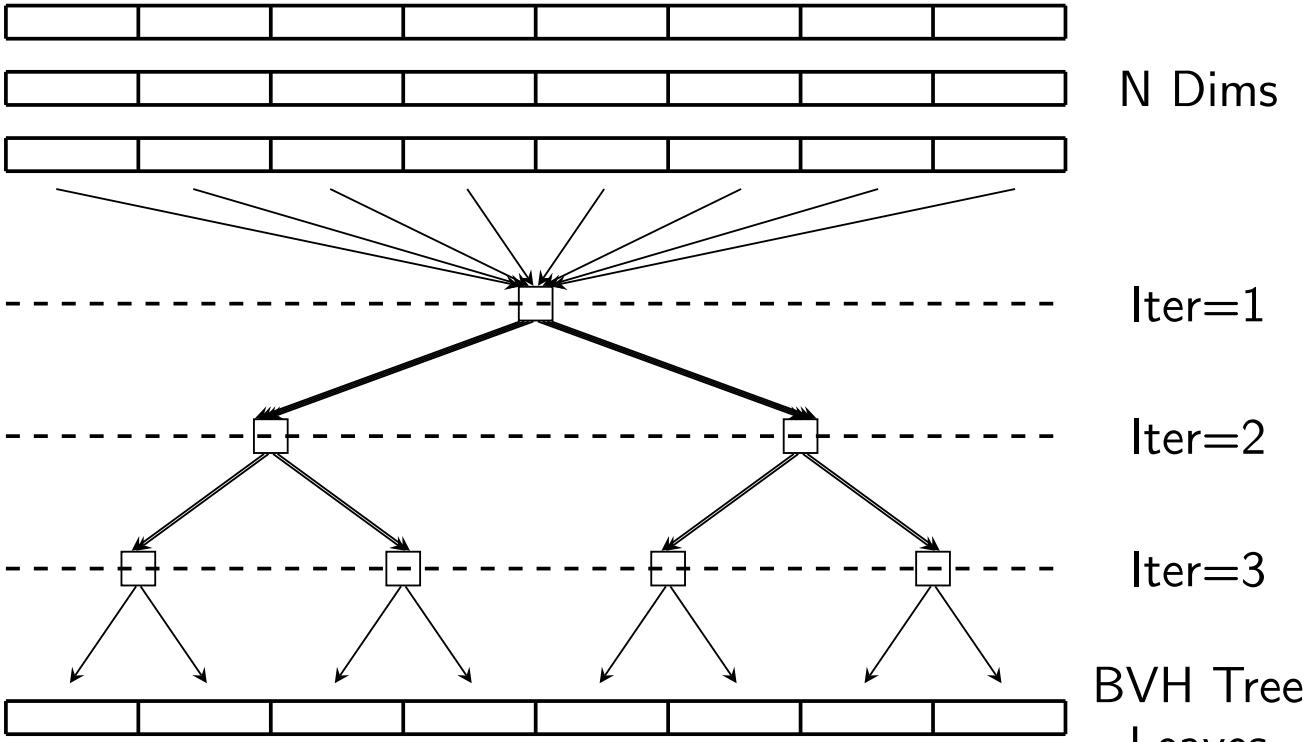
http://upload.wikimedia.org/wikipedia/en/b/bd/BitonicSort1.svg

Preparation Kernels Execution Time



BVH Tree Construction Kernel

- BVH Tree Construction
  - $O(log(n))$ Parallel Time
  - $O(2^{Iteration})$ Parallel Execution Units



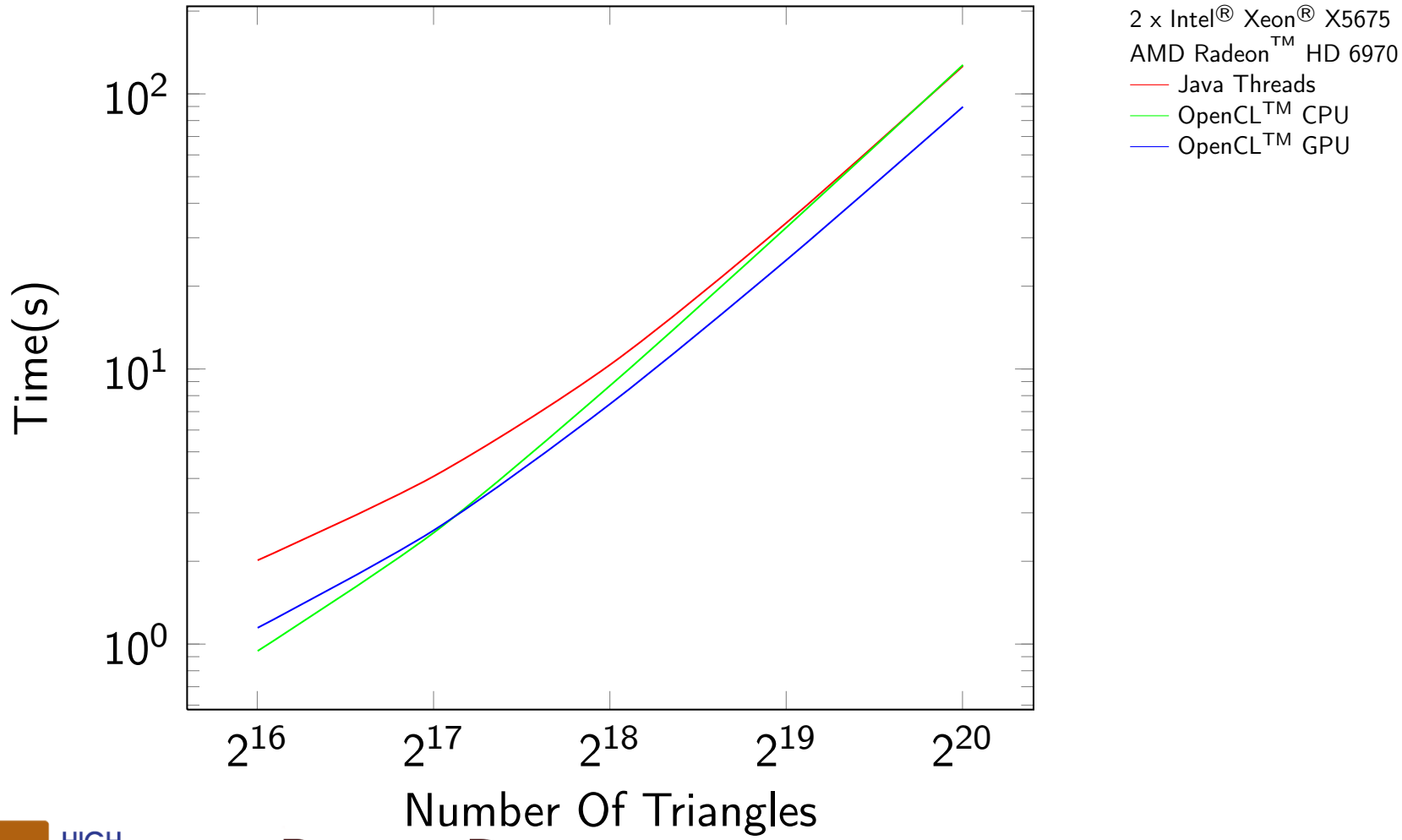
BVH Tree Construction Kernel Algorithm

```
 1: procedure BuildBVHTree
 2:     SetUpInitialNode
 3:     while NotFullySplit do
 4:         for all NodesWhichExist do
 5:             for all Dimension ∈ Dimensions do
 6:                 ChooseBestSplit
 7:             end for
 8:             SplitNode
 9:             SetupNodeForDataParallelPortion
10:             for all Dimension ∈ Dimensions do
11:                 if DimensionWasNotSplitDimension then
12:                     for all Indexes ∈ SortedIndexes do
13:                         CalculateNewPosition
14:                         Reorder
15:                     end for
16:                 end if
```

BVH Tree Construction Kernel Algorithm


```
17:                 end for
18:             end for
19:         end while
20:         for all Node ∈ Nodes do
21:             CalculateAxisAlignedBoundingBoxes
22:         end for
23:         for Values ∈ Sorted do
24:             SetToSortedIndexInFirstDimension
25:         end for
26: end procedure
```

BVH Tree Construction Kernel Execution Time



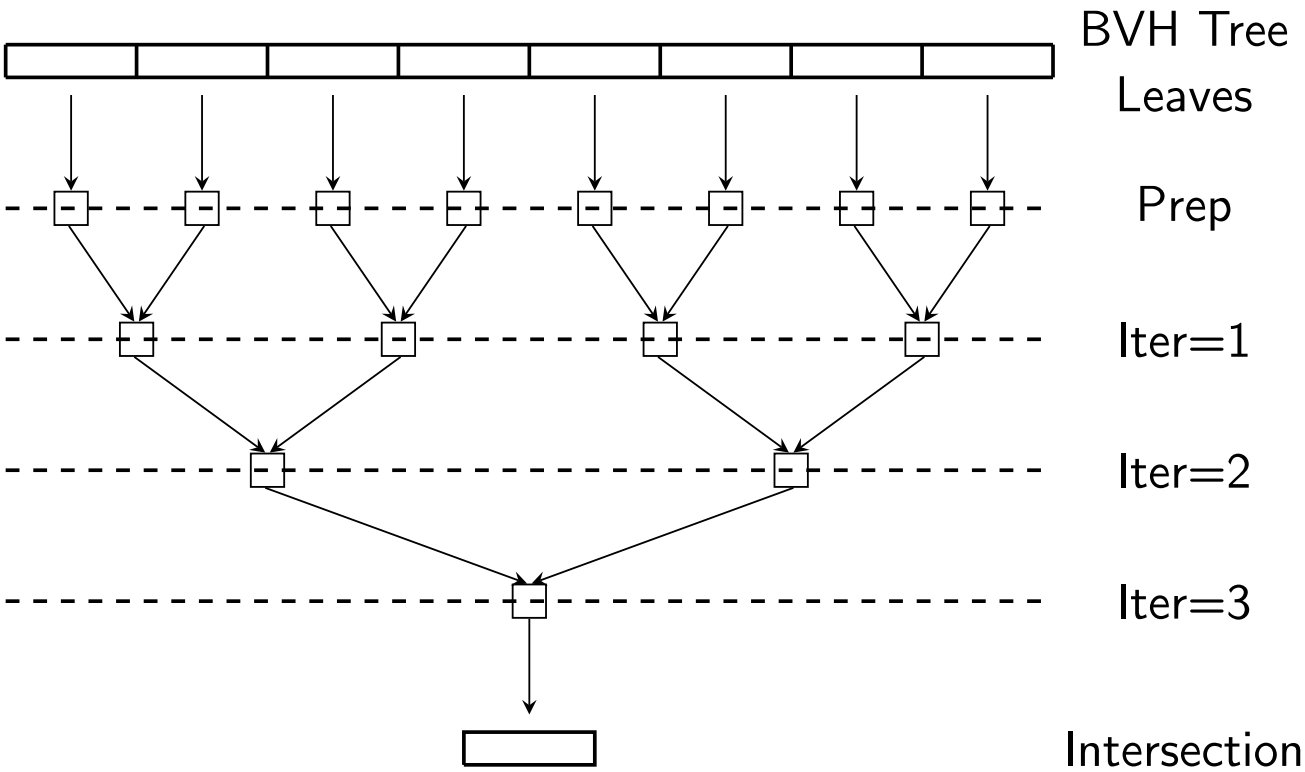
HIGH PERFORMANCE TECHNOLOGIES GROUP

Brown Deer Technology

TECHNOLOGY DRIVEN. WARFIGHTER FOCUSED.

Intersection Kernel

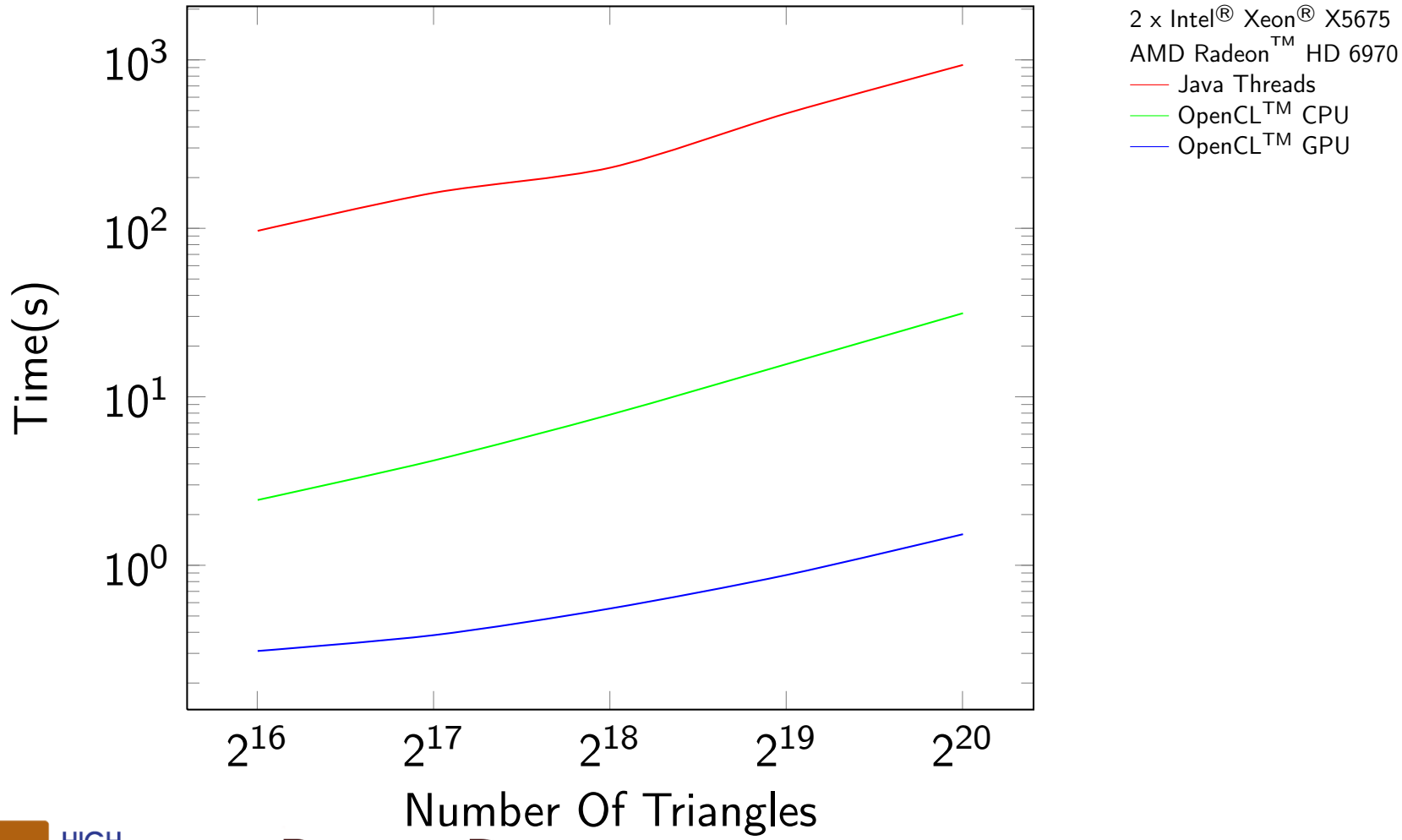
- Intersection Kernel
  - $O(log(n))$ Parallel Time
  - $O(2^{Iterations-Iteration})$ Parallel Execution Units



Intersection Kernel Algorithm


```
1:  procedure IntersectBVHTree
2:      for all Ray ∈ Rays do in Parallel
3:          for all Node ∈ LeafNodes do in Parallel
4:              if RayHitsBoundingBox then
5:                  FindClosestStructureHit
6:              end if
7:          end for
8:          FindClosestNodeHit
9:      end for
10: end procedure
```

Intersection Kernel Execution Time



Conclusion

Conclusion

- Java 7 ForkJoinPool Should Be Utilized Where It Is More Efficient

Conclusion

- Java 7 ForkJoinPool Should Be Utilized Where It Is More Efficient
- Aparapi
  - Pros:
    - Higher Performance
    - Utilize Additional Resources

Conclusion

- Java 7 ForkJoinPool Should Be Utilized Where It Is More Efficient
- Aparapi
  - Pros:
    - Higher Performance
    - Utilize Additional Resources
  - Cons:
    - Limited Usability

Questions?