ISC 2011

Producer-Consumer Model for Massively Parallel Zero-Sum Games on the GPU

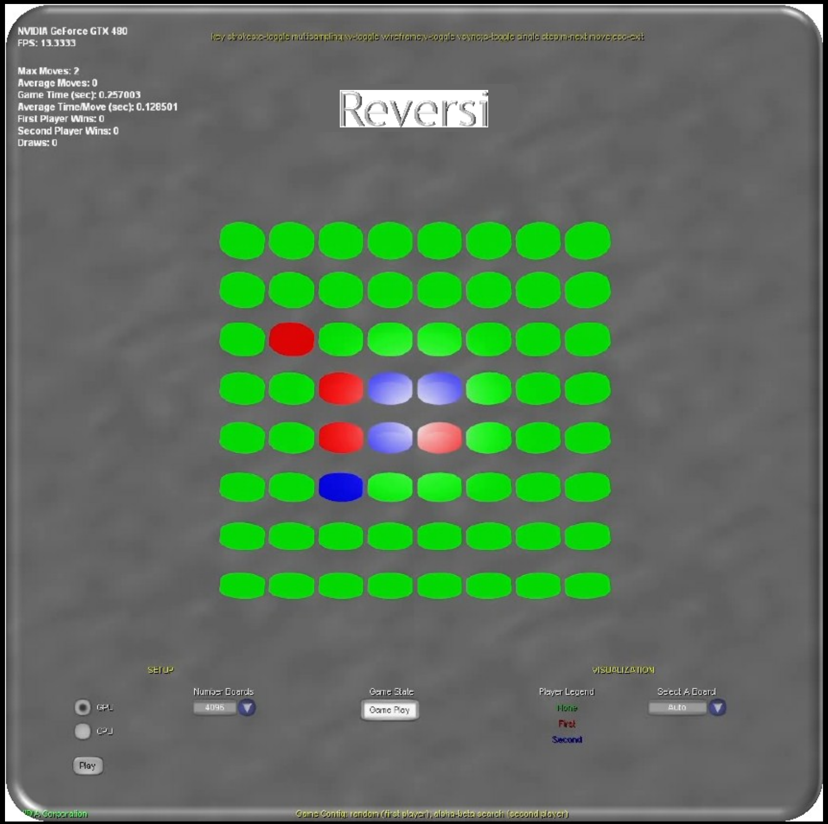
Avi Bleiweiss
NVIDIA Corporation

ISC 2011

Zero-Sum

- One's gain ⇒ other's loss
- Perfect info
- Multi player game
  - Simple and involved

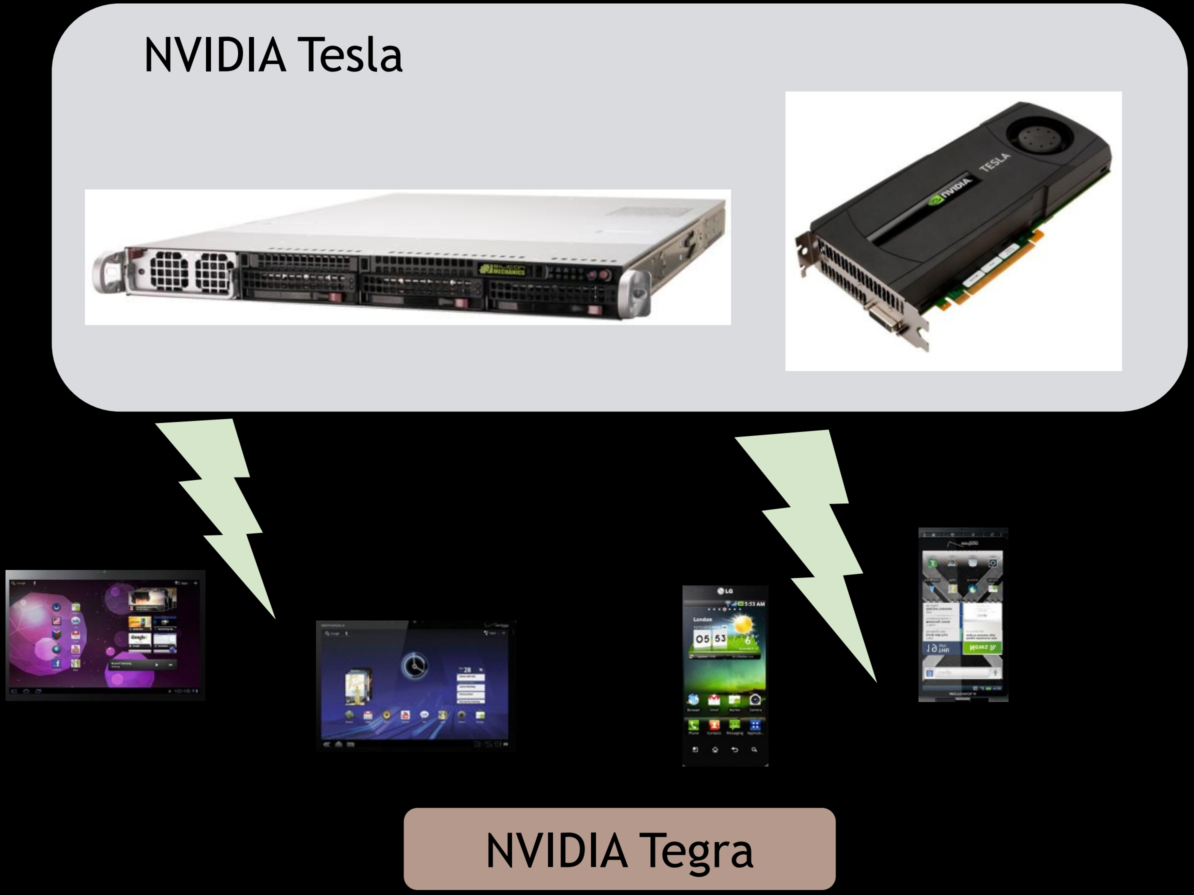
Matching Pennies	Head	Tail
Head	1,-1	-1,1
Tail	-1,1	1,-1



ISC 2011

Motivation

- Play as you go
  - Thousands players
- Game cloud
  - GPU computing
- Mobile computer
  - Commodity



Problem

Game

- Two player
- Maximize look ahead
- Rapid node expansion
  - 10^{25} for 4x4x4 Tic-Tac-Toe
  - 10^{120} for Chess

Search

- Efficient parallel
  - Space split
  - Statistical simulation
  - Simultaneous matches
- Many thousands threads

Parallelism

Principal Variation Split

- Strongly ordered tree
- Synchronization bound
- Load imbalance

Young Brothers Wait Concept

- Parallel at any node
- Processor owns node
- Up to 1024 processors

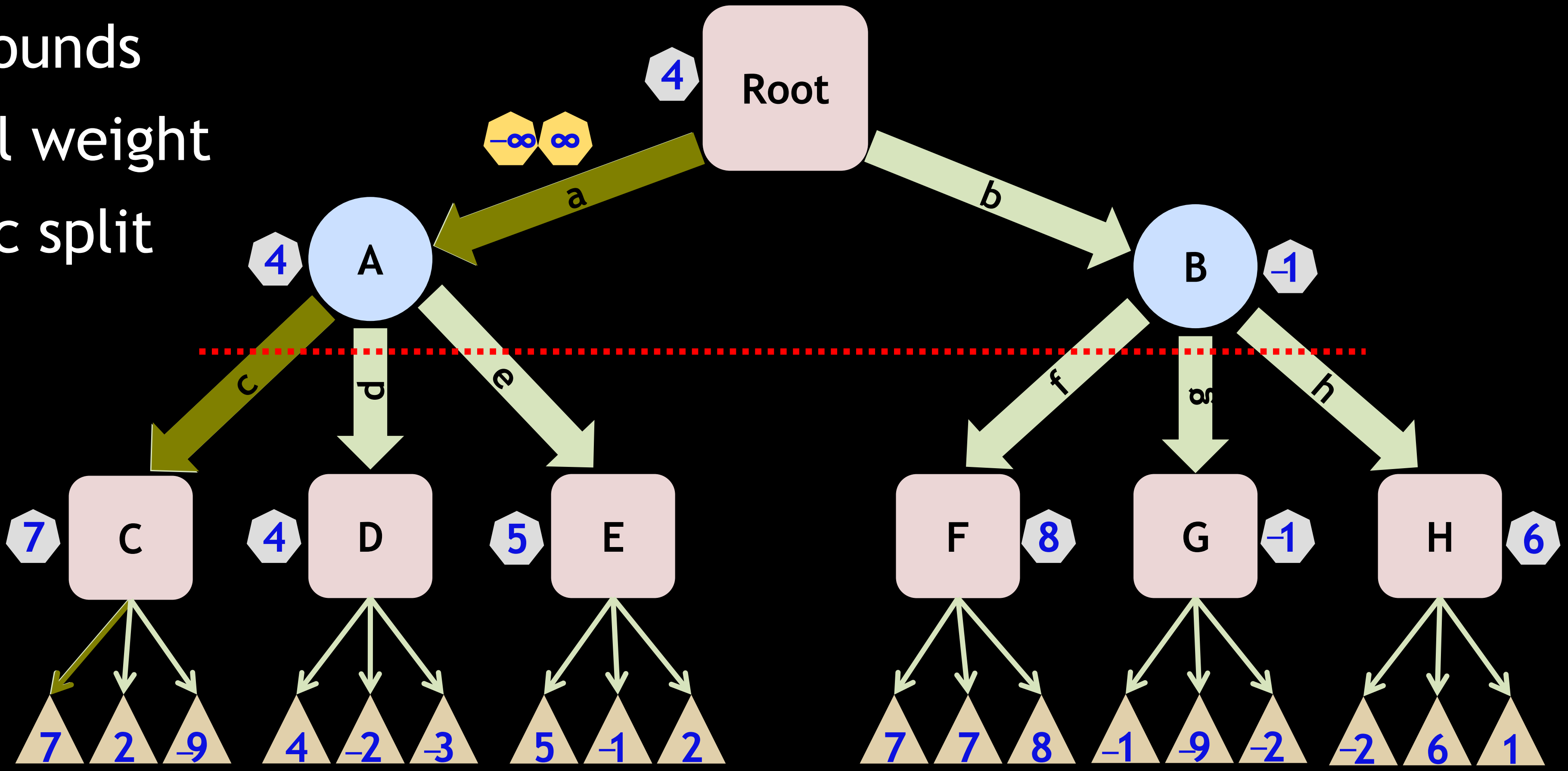
Dynamic Tree Splitting

- Processors share node
- Global job list
- Reasonable speedup

No massive parallel solution in shared memory settings!

Cut Nodes

- PV Bounds
- Equal weight
- Static split



Heuristic Search

$$
\begin{array}{l}
AlphaBeta(v, \alpha_l, \beta_l, \alpha_g, \beta_g) \\
\quad \alpha_l \leftarrow max(\alpha_l, \alpha_g) \\
\quad \beta_l \leftarrow min(\beta_l, \beta_g) \\
\quad \texttt{if}\ (\alpha_l > \beta_l)\ \texttt{return}\ rank \\
\quad \texttt{if}\ (s(v) = \emptyset)\ \texttt{return}\ f(v) \\
\quad \texttt{foreach}\ (v_j \ni s(v))\ \texttt{do} \\
\qquad value \leftarrow AlphaBeta(v_j, \alpha_l, \beta_l, \alpha_g, \beta_g) \\
\qquad \texttt{if}\ (p(v) == First)\ \texttt{then} \\
\qquad\quad rank \leftarrow max(rank, value) \\
\qquad\quad \alpha_l \leftarrow max(\alpha_l, rank) \\
\qquad\quad \texttt{if}\ (\alpha_l > \alpha_g)\ \texttt{then}\ \alpha_g \xleftarrow{atomic} \alpha_l \\
\qquad \texttt{else} \\
\qquad\quad rank \leftarrow min(rank, value) \\
\qquad\quad \beta_l \leftarrow min(\beta_l, rank) \\
\qquad\quad \texttt{if}\ (\beta_l < \beta_g)\ \texttt{then}\ \beta_g \xleftarrow{atomic} \beta_l \\
\quad \texttt{return}\ rank
\end{array}
$$

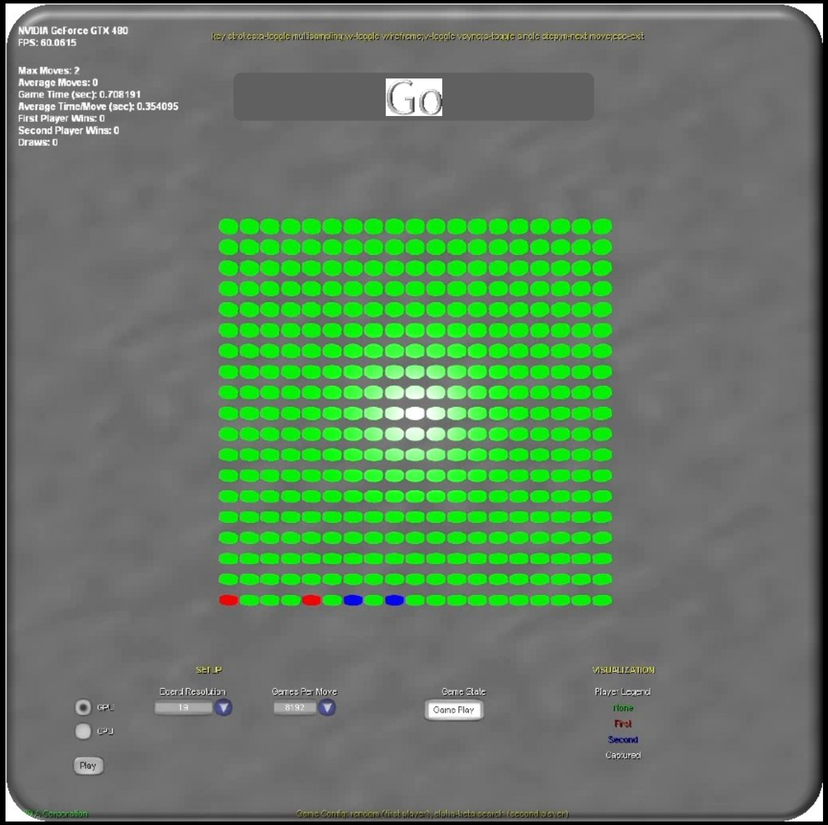
ISC 2011

Statistical Simulation

- Monte Carlo method
  - Annealing process
- k-simulation move error

$$\boxed{\Delta f(v) \sim \frac{1}{\sqrt{k}}}$$

- Many thousands games
  - for 19x19 Go game



PRESENTED BY NVIDIA.

Challenges

Deep recursion, limited stack

Divergent, irregular threads

Dynamic parallelism

Low arithmetic intensity

Implementation

- Kernel for each
  - Alpha-Beta, Monte Carlo
- Board C++ class
  - Rules specific
- Games

Heuristic Search	3D Tic-Tac-Toe	Connect-4	Reversi
Monte Carlo Simulation	Go		

ISC 2011

State Abstraction

Pitched 2D/3D

- Cells
  - Player
- Successors
  - Move

Player {none, 1^{st}, 2^{nd}}

```
struct Move {
    int3 position;
    int rank;
};
```

Rank {Win,Lose,Draw}

- Manipulate
  - Update
  - Undo
- Query
  - Winner
  - Full

Stack

- Recursion depth >1000
- Greedy allocation

$$\mathit{StackBytes/Thread} * \mathit{SM\,\#} * \mathit{MaxWarps/SM} * \mathit{Threads/Warp}$$

- Hybrid design

Runtime/Compiler

- Local variables
- Function parameters

Local Memory

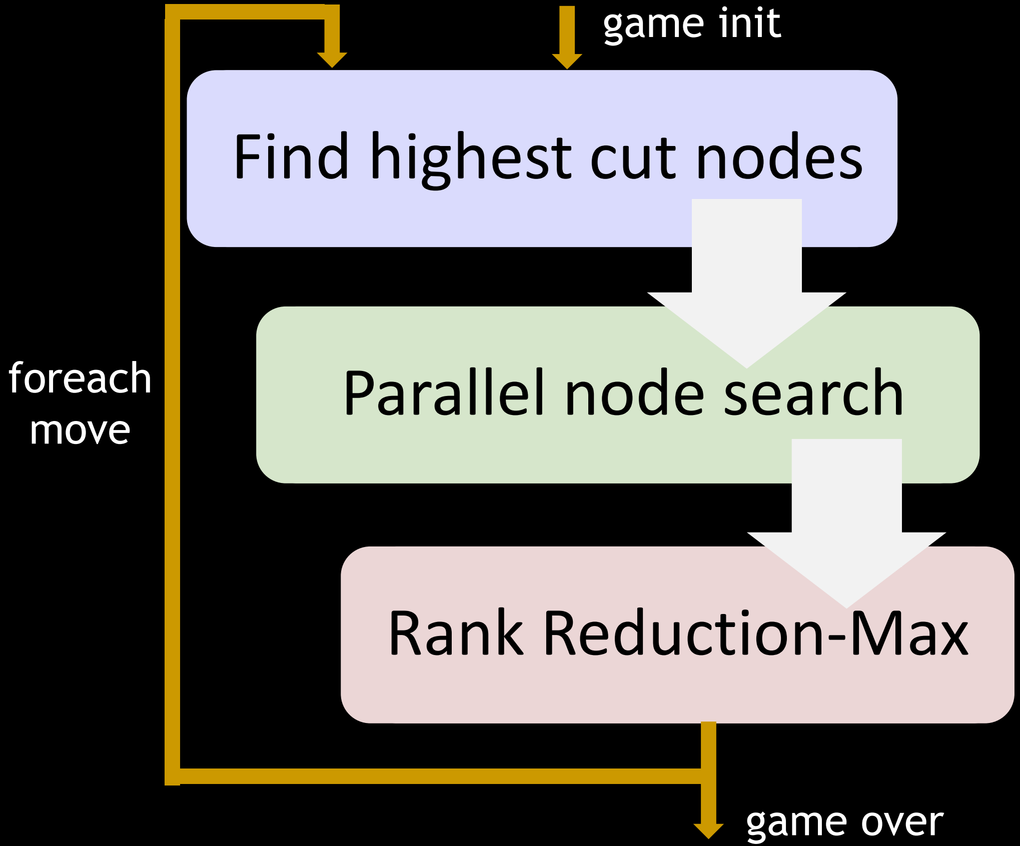
User

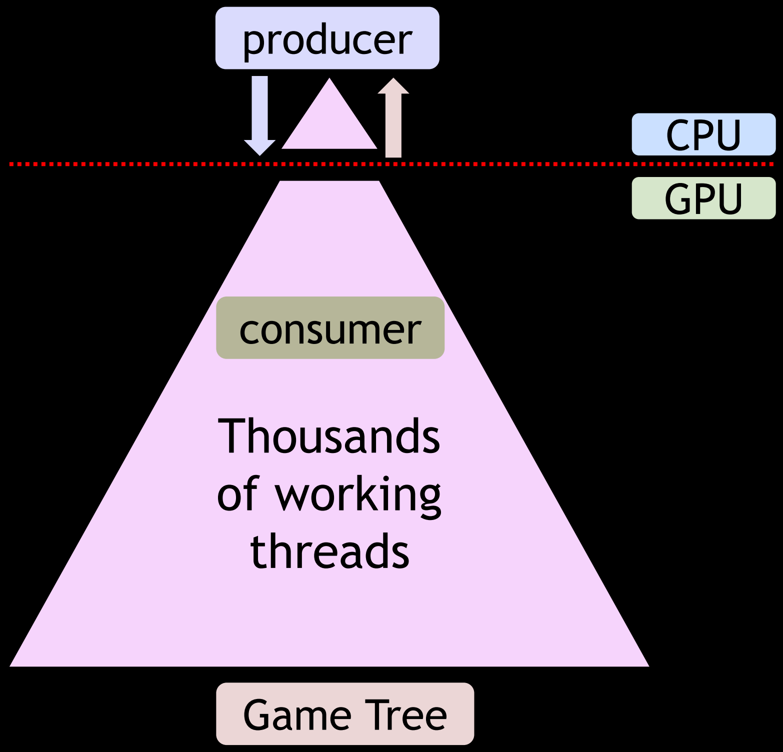
- Successors

Global Memory

ISC 2011

Producer-Consumer





Shared αβ

Kernel global scope

```
__device__ int *galpha, *gbeta;

__device__ void
resolve(int& alpha, int& beta)
{
  if(alpha ≤ *galpha) alpha = *galpha;
  else atomicMax(galpha, alpha);

  if(beta ≥ *gbeta) beta = *gbeta;
  else atomicMin(gbeta, beta);
}
```

Check local αβ

Global atomic update

Limitations

- Stack allocation
  - Bounds parallelism
- Static split constraints

Depth	Tic-Tac-Toe Threads		
	3x3x3	4x4x4	5x5x5
1	650	3906	15252
2	15600	238266	1860744

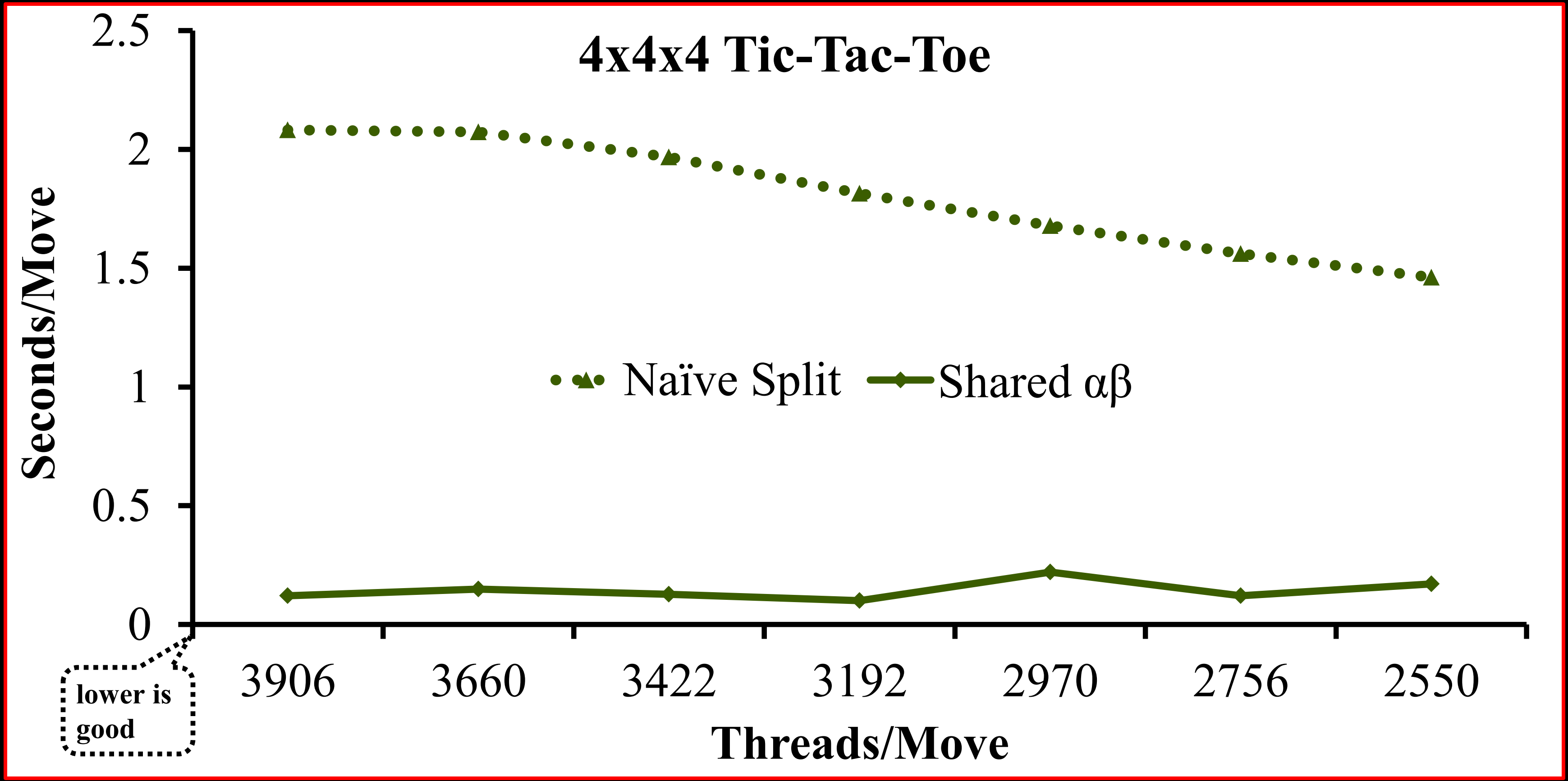
Methodology

- CUDA Toolkit 3.1, Windows
- Single processor

GPU	SMs	Warps/SM	Clocks(MHz)	L1/Shared (KB)	L2(KB)
GTX480	15	2	723/1446/1796	48/16	640

CPU	Cores	Clocks(MHz)	L1/L2 (KB)
I7-940	8	2942/(3*1066)	32/8192

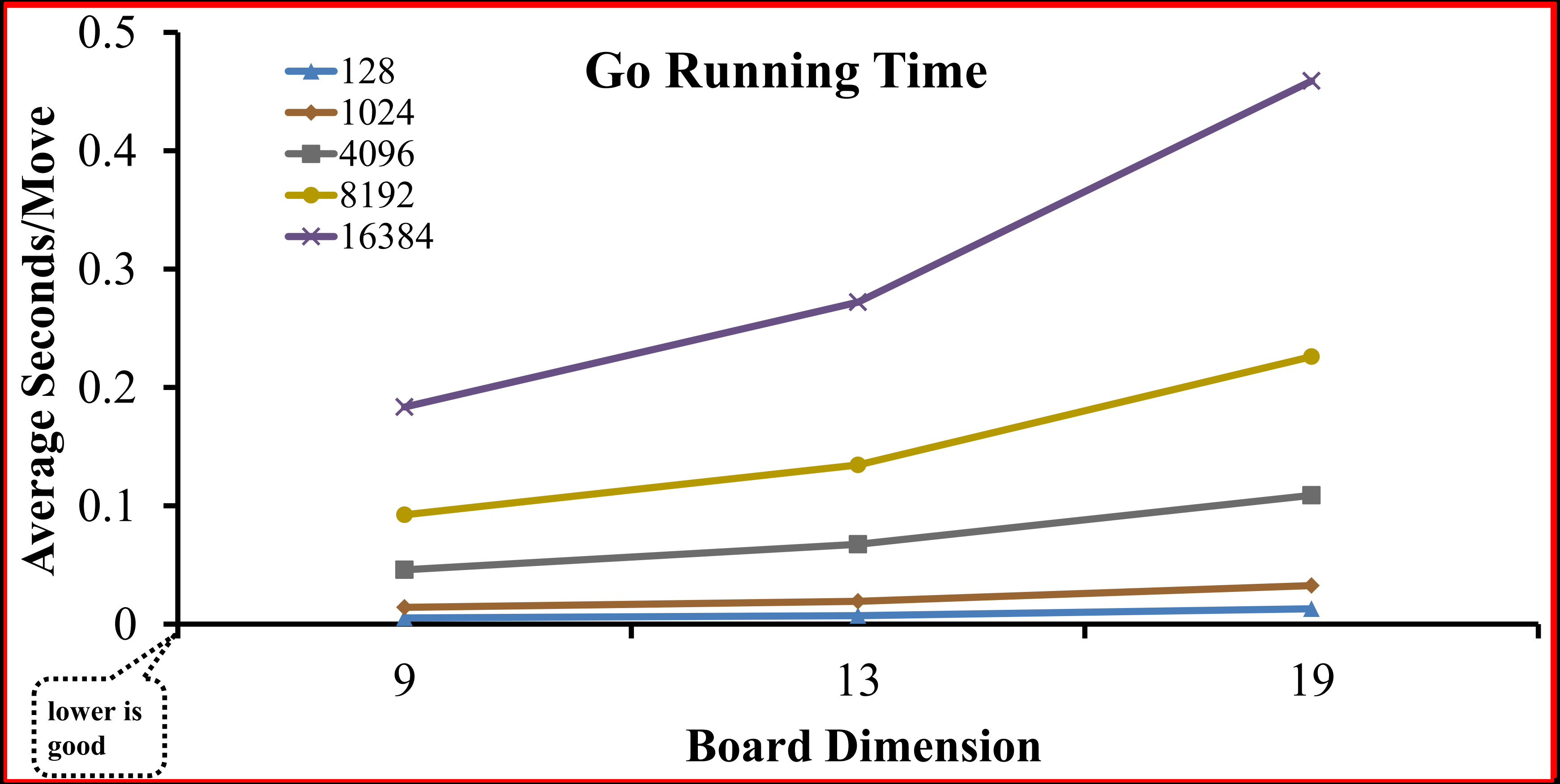
Space Split



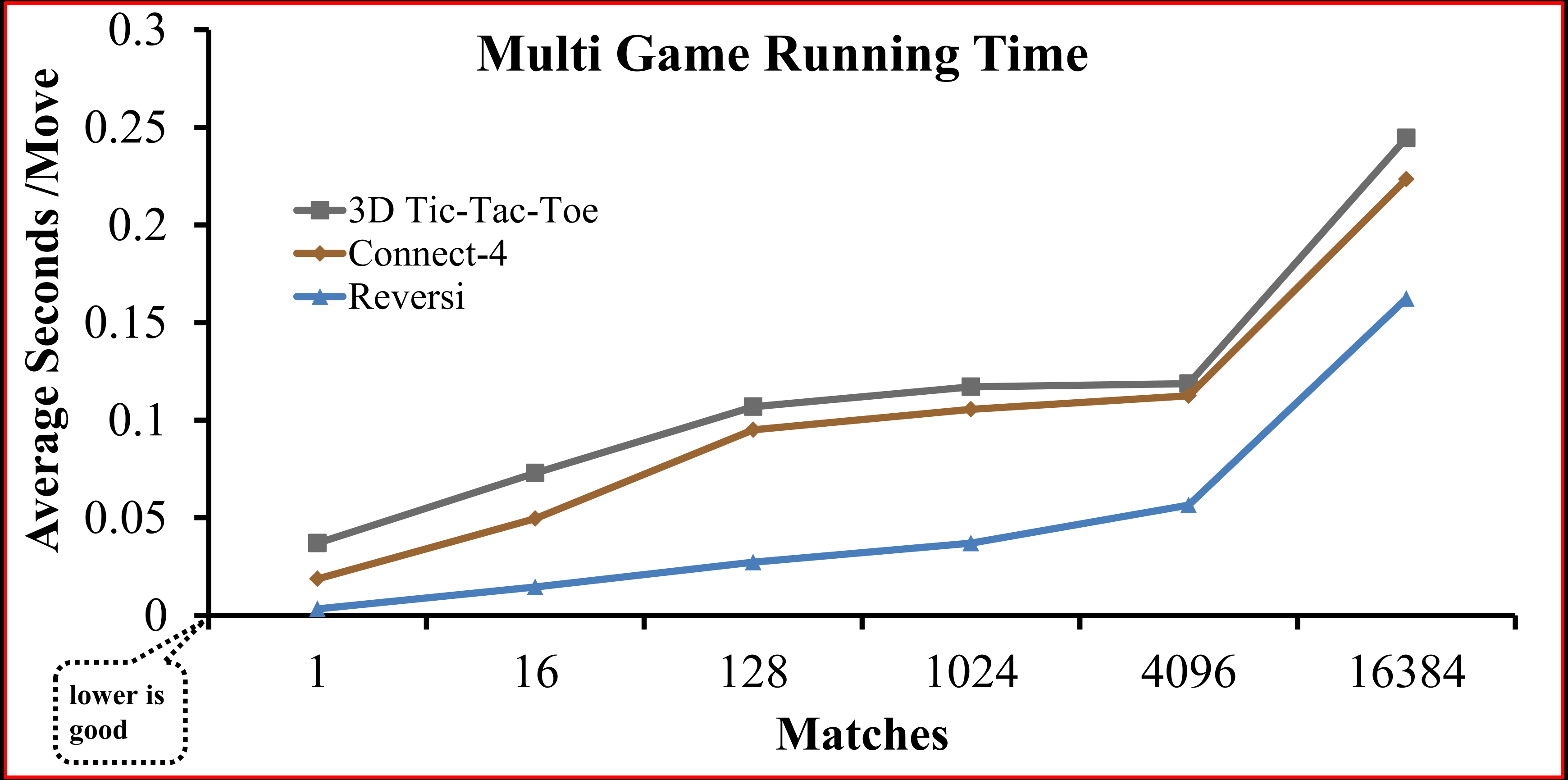
4x4x4 Tic-Tac-Toe
Seconds/Move
2.5
2
1.5
1
0.5
0
Naïve Split
Shared αβ
3906
3660
3422
3192
2970
2756
2550
Threads/Move
lower is good

ISC 2011

Monte Carlo



Simultaneous Matches



Future Work

- Data packing optimization
- Per-SM transposition table
- Predictable node ranks
- Dynamic space split

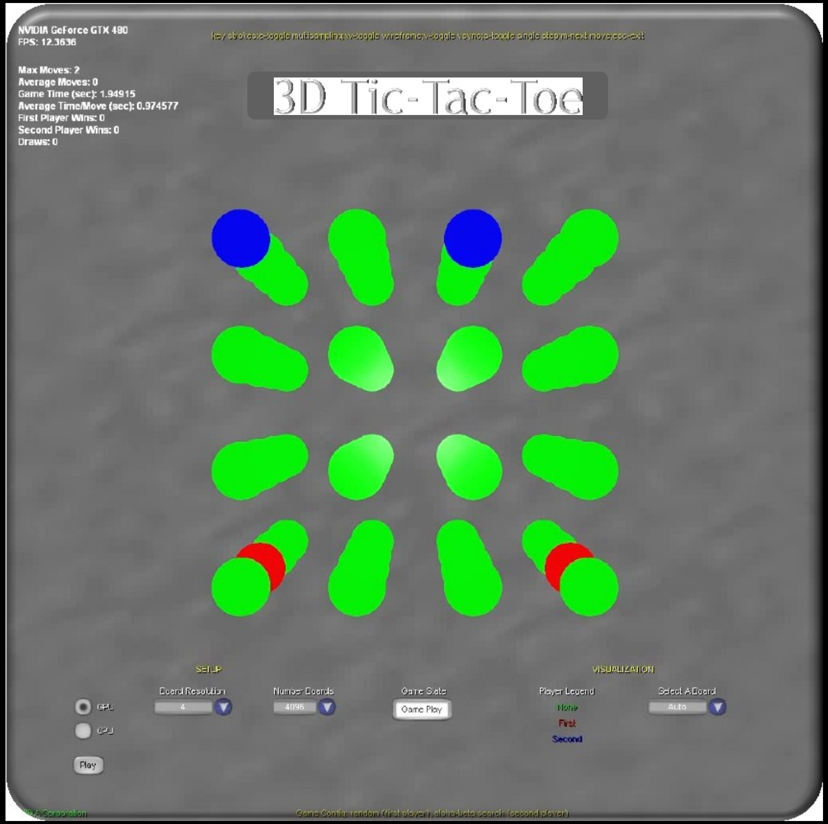
GPU Performance

Metric	Game	Dimension	Speedup
Shared αβ vs. Naïve Split	3D Tic-Tac-Toe	4x4x4	13.37X mean
Monte Carlo vs. CPU	Go	19x19	121.64X @16K

ISC 2011

Summary

- Efficient GPU based
  - Heuristic search
  - Statistical Simulation
- Economical solution
  - Mobile-Cloud



PRESENTED BY NVIDIA.

Thank You!

Questions?

Info

- Base
  - http://developer.nvidia.com
- GPU AI for Board Games
  - Technology Preview
- Toolkit
  - CUDA Zone
- Debugger
  - Parallel Nsight