Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 00 000 000	O O O	Future Work
			000		

Scaling Hierarchical *N*-Body Simulations on GPU Clusters

Pritish Jetley Lukasz Wesolowski Filippo Gioachin Laxmikant V. Kalé Thomas R. Quinn

April 29, 2010

æ

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 00 000 000	Performance ○ ○	Future Work

Clusters of GPUs provide immense computational power

æ

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 00 000 000	Performance ○ ○	Future Work

Clusters of GPUs provide immense computational power

Suitable for well-structured data parallel operations

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 00 000 000	Performance ○ ○	Future Work

Clusters of GPUs provide immense computational power

▲ @ ▶ ▲ ∃ ▶

- Suitable for well-structured data parallel operations
- Algorithms with high flop intensity do well

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 00 000 000	Performance ○ ○	Future Work

- Clusters of GPUs provide immense computational power
- Suitable for well-structured data parallel operations
- Algorithms with high flop intensity do well
- What about complex, asynchronous applications with medium grain size?

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 000 000 000 000	Performance ○ ○	Future Work

- Clusters of GPUs provide immense computational power
- Suitable for well-structured data parallel operations
- Algorithms with high flop intensity do well
- What about complex, asynchronous applications with medium grain size?
 - How can we optimize kernel performance?

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000 00 000 000		

- Clusters of GPUs provide immense computational power
- Suitable for well-structured data parallel operations
- Algorithms with high flop intensity do well
- What about complex, asynchronous applications with medium grain size?
 - How can we optimize kernel performance?
 - What are the obstacles to scaling on clusters of GPUs?

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 00 000 000 000	O O O	Future Work

æ

ChaNGa

Barnes-Hut simulator

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance O O	Future Work
			000 000		

・ロト ・日 ・ ・ ヨ ・ ・

표 문 문

ChaNGa

- Barnes-Hut simulator
 - Tree traversal
 - Force computation

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation Cha	aNGa (GPU Manager	ChaNGa on the GPU	Performance o o	Future Work
			000 000		

э

ChaNGa

- Barnes-Hut simulator
 - Tree traversal
 - Force computation
- Several production-quality techniques
 - Ewald summation
 - SPH
 - Gravitational softening
 - Quadrupole moments

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 00 000	Performance O O	Future Work
			000		

ChaNGa

- Barnes-Hut simulator
 - Tree traversal
 - Force computation
- Several production-quality techniques
 - Ewald summation
 - SPH
 - Gravitational softening
 - Quadrupole moments
- Multiple timestepping

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 00 000	Performance O O	Future Work
			000		

ChaNGa

- Barnes-Hut simulator
 - Tree traversal
 - Force computation
- Several production-quality techniques
 - Ewald summation
 - SPH
 - Gravitational softening
 - Quadrupole moments
- Multiple timestepping
- Optimized for parallel performance
 - Particle cache
 - Prefetching
 - Overlap of fetch latency with useful work
 - Scales up to 32K cores

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000 00 000 000		

(A) (□) (A) (□) (A)

æ

GPU Manager

Work-request (WR) abstraction

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 00 000	O ○ ○	Future Work
			000		



- Work-request (WR) abstraction
- Asynchronous invocation-callback paradigm

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			000		



- Work-request (WR) abstraction
- Asynchronous invocation-callback paradigm
- Asynchronous memory transfer and kernel invocation

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn Scaling Hierarchical *N*-Body Simulations on GPU Clusters

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			00		
			000		
			000		

<ロ> (日) (日) (日) (日) (日)

æ

Adapting ChaNGa to the GPU

Work accrual framework

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			000		
			000		

æ

Adapting ChaNGa to the GPU

- Work accrual framework
- Kernel structure

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			000		
			000		

æ

Adapting ChaNGa to the GPU

- Work accrual framework
- Kernel structure
- Balance CPU and GPU work

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			000		
			000		

æ

Adapting ChaNGa to the GPU

- Work accrual framework
- Kernel structure
- Balance CPU and GPU work
- Overlap tasks

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			000		
			000		

3

Adapting ChaNGa to the GPU

- Work accrual framework
- Kernel structure
- Balance CPU and GPU work
- Overlap tasks
- Reduce serial overheads

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ●00 ○0 ○00 ○00	Performance ○ ○	Future Work
Kernel Organiza	ation				

Threads per block

(日) (同) (三) (三)

æ



Figure: Organization of force computation kernels.

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ●00 ○0 ○00 ○00	Performance ○ ○	Future Work
Kernel Organiza	ation				



- Threads per block
 - ► More threads ⇒ more concurrency

- 4 同 6 4 日 6 4 日 6

æ

Figure: Organization of force computation kernels.

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ●00 ○0 ○00 ○00	Performance ○ ○	Future Work
Kornol Organiza	tion				



- Threads per block
 - More threads ⇒ more concurrency
 - More threads \Rightarrow fewer blocks/SM

æ

Figure: Organization of force computation kernels.

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ●oo ○○ ○○○ ○○○	Performance ○ ○	Future Work
Karnal Organiza	tion				



Threads per block

- ► More threads ⇒ more concurrency
- More threads \Rightarrow fewer blocks/SM

э

Optimal value?

Figure: Organization of force computation kernels.

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ●00 ○0 ○00 ○00	Performance ○ ○	Future Work
Kernel Organiza	ntion				



- Threads per block
 - ► More threads ⇒ more concurrency
 - More threads \Rightarrow fewer blocks/SM

э

- Optimal value?
- Block shape

Figure: Organization of force computation kernels.

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ●○○ ○○ ○○○ ○○○	Performance ○ ○	Future Work
Kernel Organiza	ation				



Figure: Organization of force computation kernels.

- Threads per block
 - ► More threads ⇒ more concurrency
 - More threads \Rightarrow fewer blocks/SM
 - Optimal value?
- Block shape
 - More particles \Rightarrow fewer loads

- 4 同 6 4 日 6 4 日 6

э

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ●00 ○0 ○00 ○00	Performance ○ ○	Future Work
Kanal Organia	ation.				



Figure: Organization of force computation kernels.

- Threads per block
 - ► More threads ⇒ more concurrency
 - More threads \Rightarrow fewer blocks/SM
 - Optimal value?
- Block shape
 - More particles \Rightarrow fewer loads
 - ► More particles ⇒ more shared memory/block

- 4 @ > - 4 @ > - 4 @ >

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ●00 ○0 ○00 ○00	Performance ○ ○	Future Work
Kanal Organia	ation.				



Figure: Organization of force computation kernels.

- Threads per block
 - ► More threads ⇒ more concurrency
 - More threads \Rightarrow fewer blocks/SM
 - Optimal value?
- Block shape
 - More particles \Rightarrow fewer loads
 - ► More particles ⇒ more shared memory/block

Optimal shape?

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○●○ ○○ ○○○ ○○○	Performance ○ ○	Future Work
Kernel Organiza	ation				

Experimental Results



• Works best with T = 128, 16 particles, 8 nodes per block

Image: A mathematical states and a mathem

æ

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○● ○○○ ○○○	Performance ○ ○	Future Work
Kernel Organiza	tion				

Ewald Computation

Structured as two (real and Fourier space) kernels

- Fewer registers per thread
- More blocks per SM
- Constant memory used in Fourier-space
- Speedup of about 20 over CPU

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ○○ ○○○ ○○○	O O O	Future Work
Tree Traversal vs	5. Computation				

э

Balancing Tree Traversal and Computation

GPU is hungry for work

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ○○ ○○○ ○○○	O O O	Future Work
Tree Traversal vs	5. Computation				

э

Balancing Tree Traversal and Computation

- GPU is hungry for work
 - CPU shouldn't hold back GPU

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ●○ ○○○ ○○○	O O O	Future Work
Tree Traversal v	s. Computation				

- GPU is hungry for work
 - CPU shouldn't hold back GPU
 - Spend less time traversing tree, more time computing

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ●○ ○○○ ○○○	Performance ○ ○	Future Work
Tree Traversal v	s. Computation				

- GPU is hungry for work
 - CPU shouldn't hold back GPU
 - Spend less time traversing tree, more time computing

Increase average bucket size

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ●○ ○○○ ○○○	Performance ○ ○	Future Work
Tree Traversal v	s. Computation				

- GPU is hungry for work
 - CPU shouldn't hold back GPU
 - Spend less time traversing tree, more time computing

▲ @ ▶ ▲ ∃ ▶ ▲

- Increase average bucket size
 - Tree is shallower: less traversal time on CPU

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn Scaling Hierarchical N-Body Simulations on GPU Clusters

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ●○ ○○○ ○○○	Performance ○ ○	Future Work
Tree Traversal v	s. Computation				

- GPU is hungry for work
 - CPU shouldn't hold back GPU
 - Spend less time traversing tree, more time computing

▲ @ ▶ ▲ ≥ ▶ ▲

- Increase average bucket size
 - Tree is shallower: less traversal time on CPU
 - Generates more computation: GPU kept busy

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn Scaling Hierarchical *N*-Body Simulations on GPU Clusters

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ●○ ○○○ ○○○	Performance ○ ○	Future Work
Tree Traversal v	s. Computation				

- GPU is hungry for work
 - CPU shouldn't hold back GPU
 - Spend less time traversing tree, more time computing
- Increase average bucket size
 - Tree is shallower: less traversal time on CPU
 - Generates more computation: GPU kept busy
 - Too much computation work hinders performance

- 4 同 ト 4 ヨ ト 4 ヨ ト

Optimal bucket size?

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
Tree Trevensel					

Experimental Results



A⊒ ► < ∃

3

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			00		
			000		

Overlapping CPU and GPU Work



Figure: Traversals construct interaction lists on host. These are sent to the device as Work Requests (WRs) for computation. Overlap is possible between these activities.

イロト 不得 トイヨト イヨト

æ

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn Scaling Hierarchical N-Body Simulations on GPU Clusters

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ○●○ ○○○	Performance ○ ○	Future Work
CPU-GPU Over	rlap				

イロト イヨト イヨト イヨト

æ

Obtaining Optimal Overlap

• More WRs
$$\Rightarrow$$
 more overlap

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○ ○● ○●○ ○○○	Performance ○ ○	Future Work
CPU-GPU Over	lap				

3

Obtaining Optimal Overlap

- More WRs \Rightarrow more overlap
- ► More WRs ⇒ more offload overhead

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○ ○● ○●○ ○○○	Performance ○ ○	Future Work
CPU-GPU Over	lap				

・ロト ・聞ト ・ヨト ・ヨト

3

Obtaining Optimal Overlap

- More WRs \Rightarrow more overlap
- ► More WRs ⇒ more offload overhead
- Optimal overlap?

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ○○● ○○○	Performance ○ ○	Future Work

CPU-GPU Overlap

Experimental Results



Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			000		
Serial Overhead	c				

イロト イヨト イヨト イヨト

æ

A Lower Bound on Execution Time

How well can we do?

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ○○○ ●○○	Performance ○ ○	Future Work
Serial Overhead	s				

æ

A Lower Bound on Execution Time

► How well can we do?

$$T_{gpu} = max(T_{cpu}^{l}, T_{gpu}^{f}) + T_{cpu}^{ovhd}$$

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			000		
			000		
Serial Overhead					

æ

A Lower Bound on Execution Time

► How well can we do?

$$T_{gpu} = max(T_{cpu}^{I}, T_{gpu}^{f}) + T_{cpu}^{ovhd}$$

• Perfect overlap
$$\Rightarrow max(T_{cpu}^{l}, T_{gpu}^{f}) = T_{cpu}^{l}$$

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			000		
			000		
Serial Overhead					

æ

A Lower Bound on Execution Time

How well can we do?

$$T_{gpu} = max(T_{cpu}^{l}, T_{gpu}^{f}) + T_{cpu}^{ovhd}$$

• Perfect overlap
$$\Rightarrow max(T_{cpu}^{l}, T_{gpu}^{f}) = T_{cpu}^{l}$$

• Full efficiency
$$\Rightarrow T_{cpu}^{ovhd} = 0$$

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			00		
			000		
			000		

æ

A Lower Bound on Execution Time

How well can we do?

$$T_{gpu} = max(T_{cpu}^{l}, T_{gpu}^{f}) + T_{cpu}^{ovhd}$$

• Perfect overlap
$$\Rightarrow max(T_{cpu}^{l}, T_{gpu}^{f}) = T_{cpu}^{l}$$

• Full efficiency
$$\Rightarrow$$
 $T_{cpu}^{ovhd} = 0$

• Therefore,
$$T^*_{gpu} = T^I_{cpu}$$

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			000		
			000		
Sevial Overhead					

A Lower Bound on Execution Time

How well can we do?

$$T_{gpu} = max(T_{cpu}^{I}, T_{gpu}^{f}) + T_{cpu}^{ovhd}$$

• Perfect overlap $\Rightarrow max(T_{cpu}^{l}, T_{gpu}^{f}) = T_{cpu}^{l}$

• Full efficiency
$$\Rightarrow$$
 $T_{cpu}^{ovhd} = 0$

• Therefore,
$$T^*_{gpu} = T^I_{cpu}$$

► And, $T_{cpu}^{ovhd} = T_{gpu} - T_{gpu}^* = T_{gpu} - T_{cpu}^I$

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU	Performance	Future Work
			000		
			000		
			000		
	-				

Unexpected Serial Overhead



- CUDA memory allocation/free calls block CPU
- Repeated memory pinning costs

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn Scaling Hierarchical N-Body Simulations on GPU Clusters

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU ○○○ ○○○ ○○●	Performance O O	Future Work
Serial Overhead	s				

Experimental Results



3

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn



Scaling Performance on Lincoln



A (1) > 4

3

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation Cha	aNGa GI	PU Manager 🛛 🔾	ChaNGa on the GPU	Performance	Future Work
			000 00 000 000	•	

Comparison

Comparison of CPU-only and CPU-GPU versions

Procs.		GPUs	3m		16m		80m		
			Sn	GFLOPS	Sn	GFLOPS	Sn	GFLOPS	
-	14	4	9.5	57.17					
	28	8	8.75	102.84	14.14	176.43			
	56	16	7.87	176.31	14.43	357.11			
	112	32	6.45	276.06	12.78	620.14	9.92	450.32	
	224	64	5.78	466.23	13.21	1262.96	10.07	888.79	
	448	128	3.18	537.96	9.82	1849.34	10.47	1794.06	
	896	256					-	3819.69	

Table: Speedups and computation rates with various data sets.

イロト イポト イヨト イヨト

2

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn

Motivation	ChaNGa	GPU Manager	ChaNGa on the GPU 000 00 000 000	Performance ○ ○	Future Work

Future Work

- Larger data sets, full machine runs
- Multistepped execution performance
- Load balancing issues with highly-clustered data sets
- (Single precision) hexadecapole moments
- Port SPH computation to GPU
- Fast multipole methods
- Pipelined tree traversal on the GPU
- Compare with other heterogeneous systems

Pritish Jetley, Lukasz Wesolowski, Filippo Gioachin, Laxmikant V. Kalé, Thomas R. Quinn