Large-Scale Text Mining on GPU

Xiaohui Cui

Computational Sciences and Engineering Division

Oak Ridge National Laboratory

Oak Ridge National Laboratory ----- DOE largest science and energy laboratory

- $1.3B budget
- 4,350 employees
- 3,900 research guests annually
- $350 million invested in modernization
- World's most powerful open scientific computing facility
- Nation's largest concentration of open source materials research
- Nation's most diverse energy portfolio
- Operating the world's most intense pulsed neutron source
- Managing the billion-dollar U.S. ITER project

OAK RIDGE National Laboratory

Applied Software Engineering Research Group

- The Applied Software Engineering Research (ASER) Group at the ORNL conducts innovative computer science research into some of the most challenging problems the nation faces.
- The group consists of 8 PhD scientists, 5 engineers, 2 MBAs and several postdoctorals, Postmasters and Intern-students.
- 2007 Won R&D 100 Award
- Over the last 10 years we have developed a number of agent projects, and an agent framework called the Oak Ridge Mobile Agent Community (ORMAC).
- ASER have developed agent-based solutions in Intelligence Analysis, Cyber Security, Geospatial Analysis, Supply Chain Management, Lean Manufacturing, Scientific Data Management, Data Fusion, Distilled simulation and Semantic Web Applications with average $4M annual research budget.

Core Capabilities

Research:

- **Advanced Text Analysis**
  - Distributed clustering of massive data
  - Entity information extraction
- **High Performance Computing**
  - GPU Computing Programming Model
  - Cloud Computing
- **Collective Intelligence and Emergent Behaviors**
  - Non-traditional, nature-inspired techniques
  - Social Network Analysis & Mass collaboration model
- **Intelligent Software Agents**
  - Advanced software framework for multi-agent systems
  - Agent Based Model and Simulation

Engineering

- **Knowledge Discovery from Large Scale Dataset**
- **Social Media Tool for Knowledge Sharing and Collaboration**
- **Cyber security monitoring, protecting and count measuring**
- **Energy efficient high performance computing**

CUDA/GPU Has Been Used To Accelerate…

- **Scientific computation;**
- **Physics and molecular dynamics simulation;**
- **Image processing;**
- **Games;**
- **Digital signal processing;**
- **Finance;**
- **Data streaming;**
- **Text Mining ?**

Outline

- ❑ **Text Analysis/Mining**

- ❑ **GPU Computing**

- ❑ **Text Analysis/Mining on GPU Cluster**
  - ❑ **Text Encoding**
  - ❑ **Dimension Reduction**
  - ❑ **Document Clustering**

The GPU Enhanced Computer for Large-Scale Text Mining

Credit to ORNL Intern-Students:

- Jesse St. Charles, Carnegie Mellon University
- Joseph Cavanagh, University of Minnesota
- Yongpeng Zheng, North Carolina State University

LOCKHEED MARTIN
We never forget who we're working for™

Office of Science
U.S. DEPARTMENT OF ENERGY

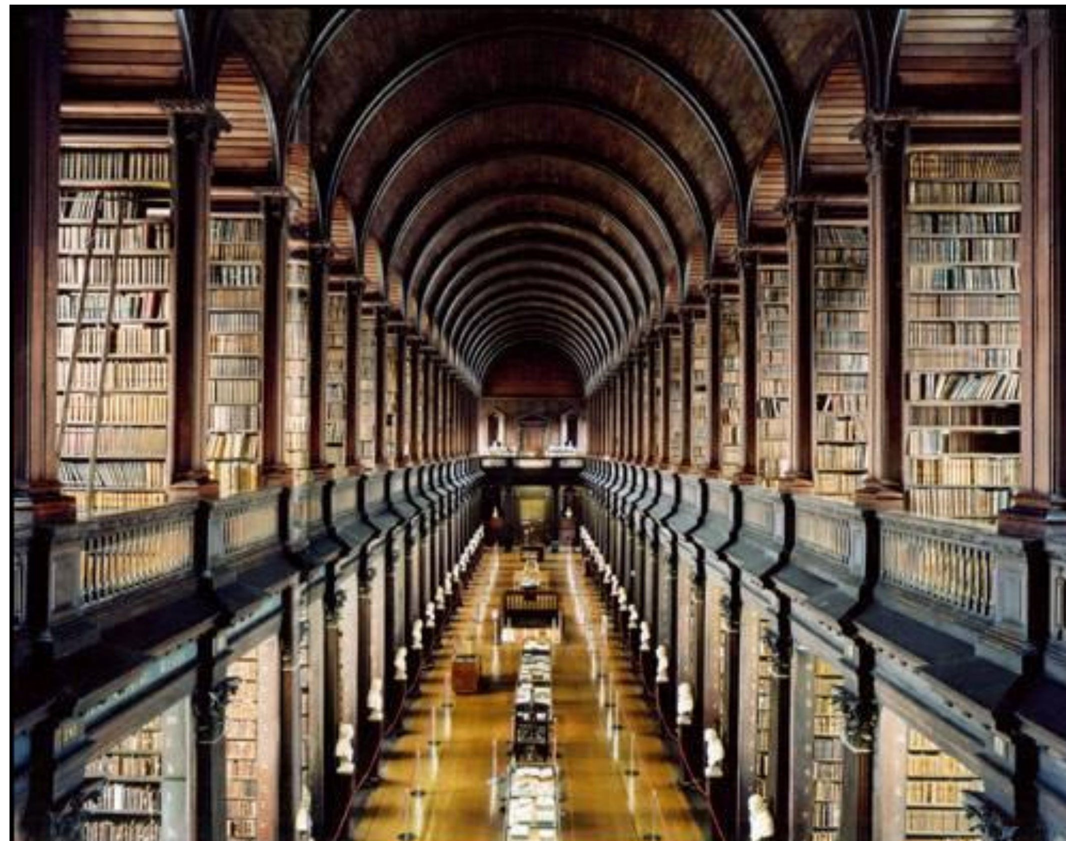
SULI

Text Analysis/Mining

We can read a newspaper, but not a library …


THE VICTORIA GAZETTE.

VOL. I. VICTORIA, VANCOUVER ISLAND, FRIDAY, JUNE 25, 1858. NO. 1.



Overview of Text Analysis

- **Keyword Methods – Very fast, good for millions of documents**
  - **Search/Query**
    - **“253-43-6834”**
    - **Good if you know what you are looking for and there are a small number of hits**
  - **Unsupervised Classification**
    - **“What is on this hard drive?”**
    - **Good to get a general overview of a set of data**
  - **Supervised Classification**
    - **“Who is engaged in arms trafficking in the Middle East?”**
    - **Good for finding topics of interest**

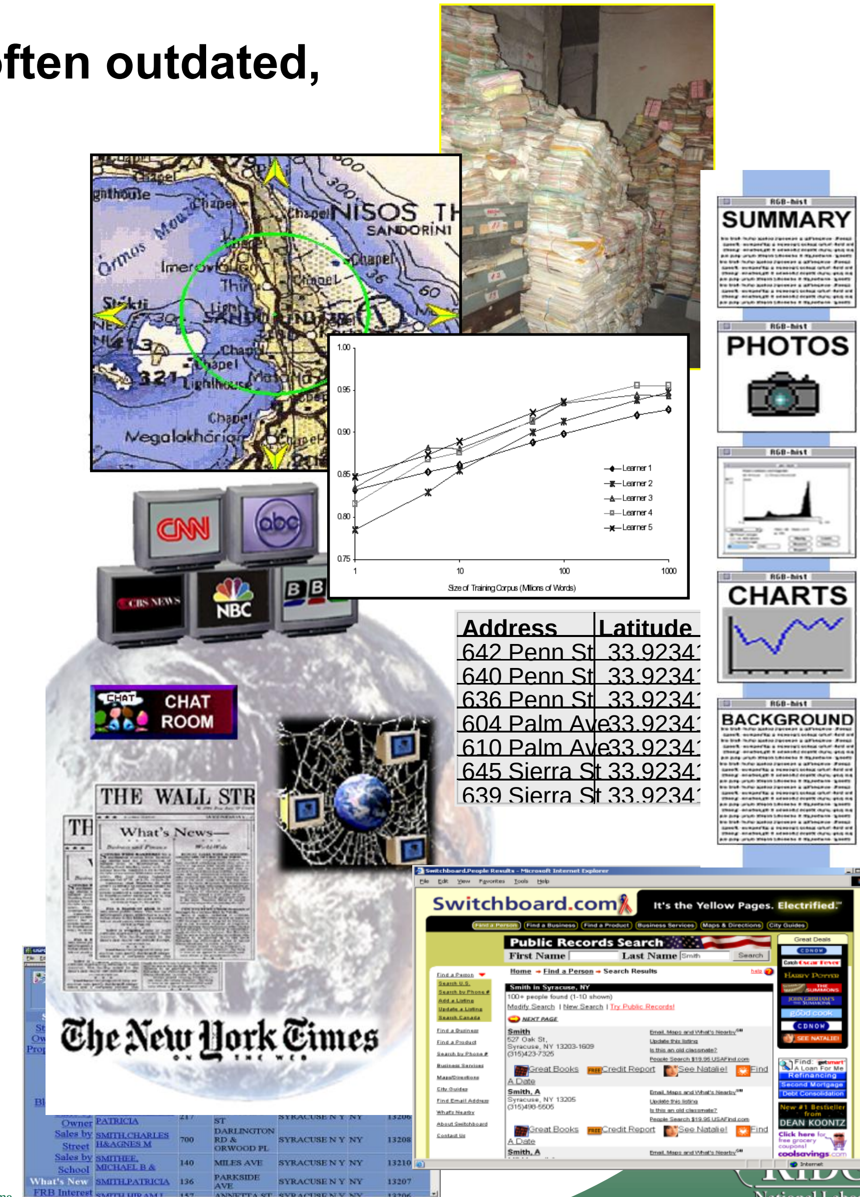
The Knowledge Base of the Digital World…

- We live in the infosphere, Data everywhere
- …but it's unstructured, inconsistent, often outdated,
  …in other words…a mess!

Giga Google

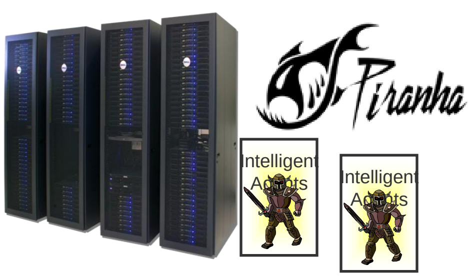
Indexing billions of web pages

Google Data Center



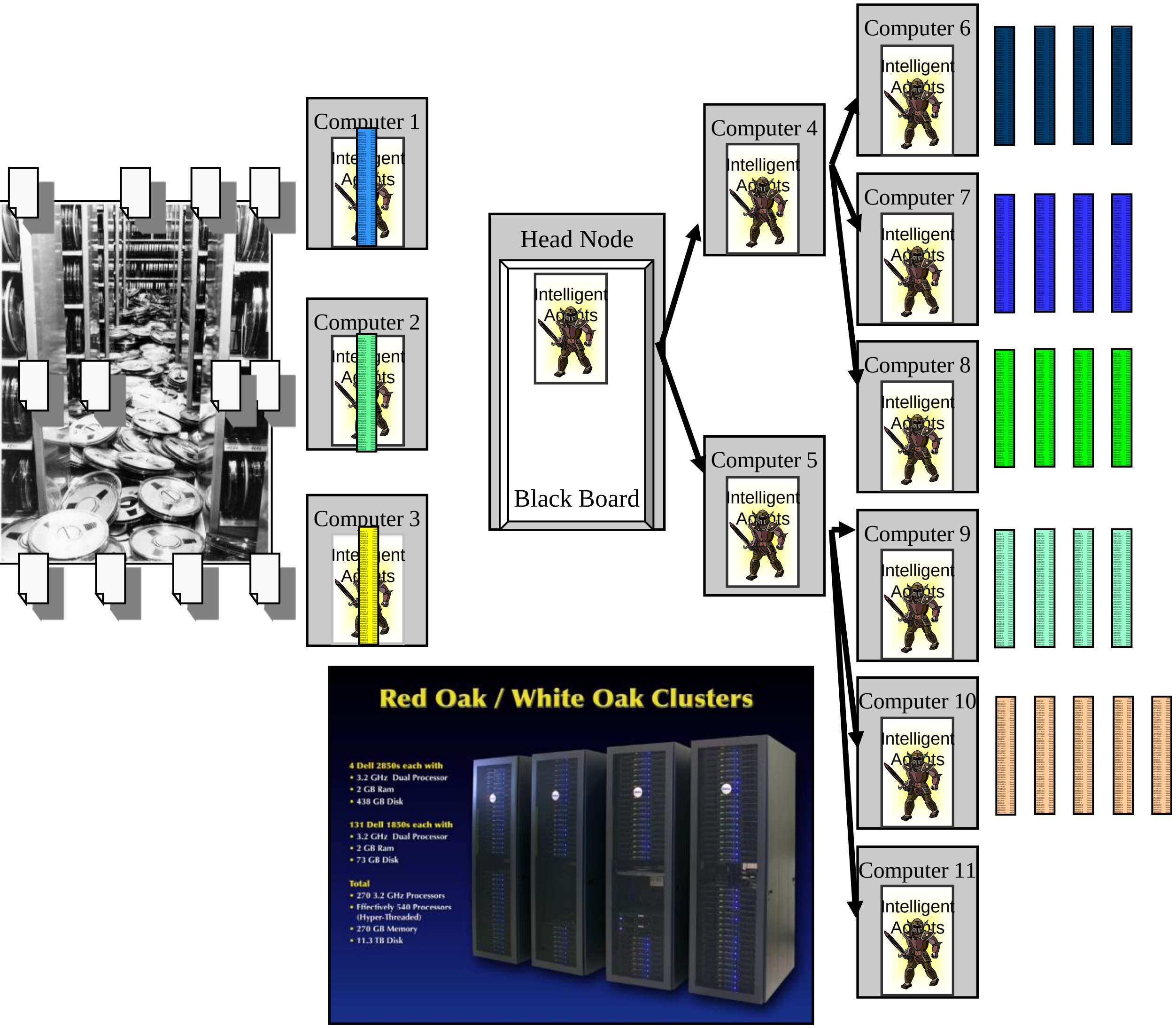
- What is in there?
- Are there any threats?
- What am I missing?

Piranha: Agent Based Text Analysis

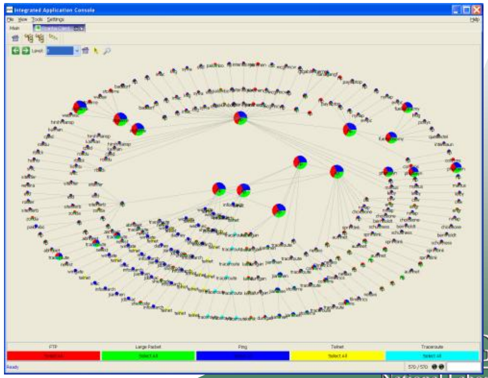


ORNL Red/White Oak Clusters
135 Dell Computers,1.7 TFLOPS

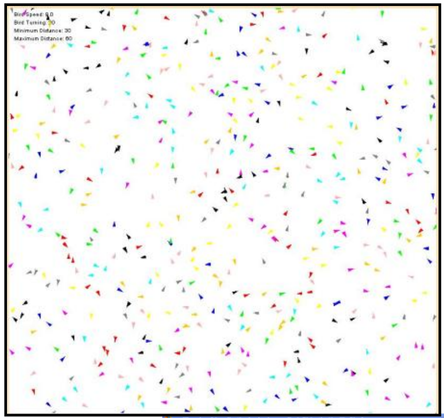
Agent Based Document Analysis System

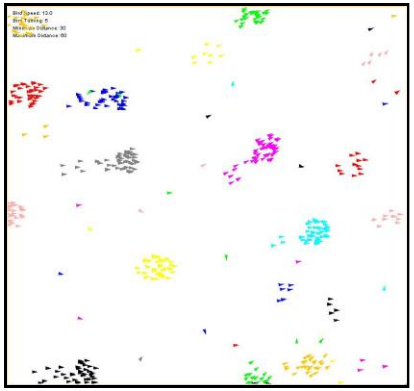


- Standard Approach
  - 11.5 Days (Single Machine)
- Agent approach
  - <u>8 minutes 24 Seconds!</u>
  - 2000 times faster
  - With no loss of accuracy



Web Based Document Clustering Project

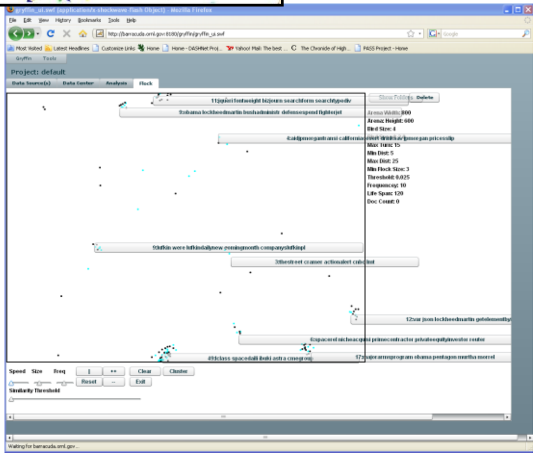


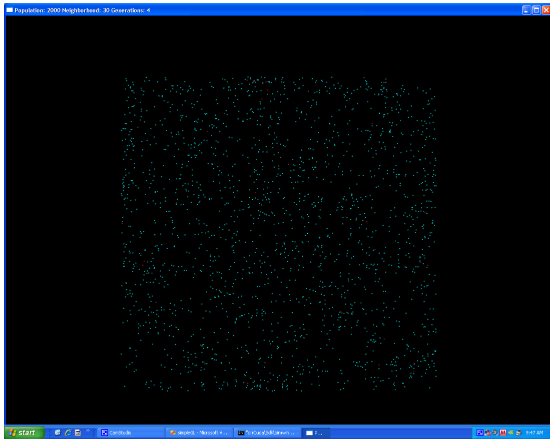


Self organizing property in flocking model for documents clustering

Each document is projected as a bird in a 2D virtual space. The document birds that have similar features will automatically group together and became a flock.

Other birds that have different document vector features will stay away from this flock.



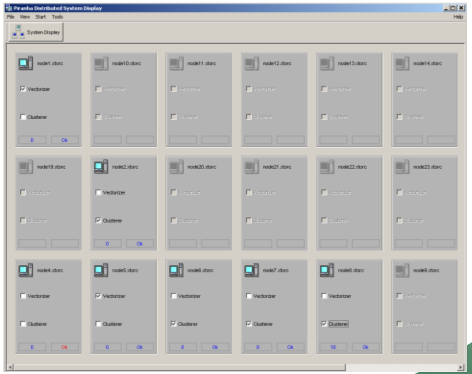


Piranha Server

4 Dell 2850s each with
- **3.2 GHz Dual Processor**
- **2 GB Ram**
- **438 GB Disk**

131 Dell 1850s each with
- **3.2 GHz Dual Processor**
- **2 GB Ram**
- **73 GB Disk**

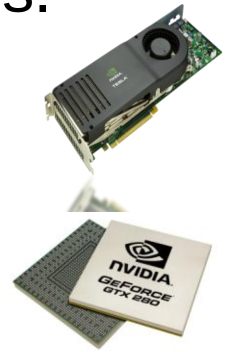
Total
- **1.7 TFLOPS**
- **270 GB Memory**
- **11.3 TB Disk**

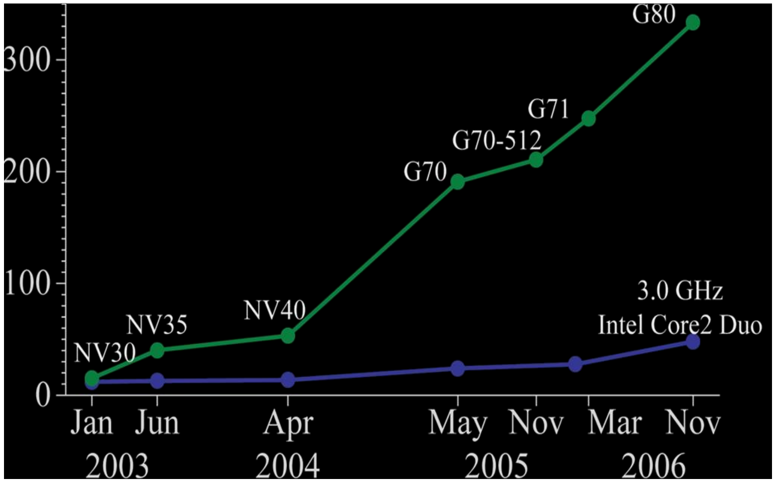


Faster, Smaller and Cost Efficient…

- We are quickly reaching an age in which a capability is needed for text mining (TM) of terabyte-scale unstructured text corpora for prompt decision-making.

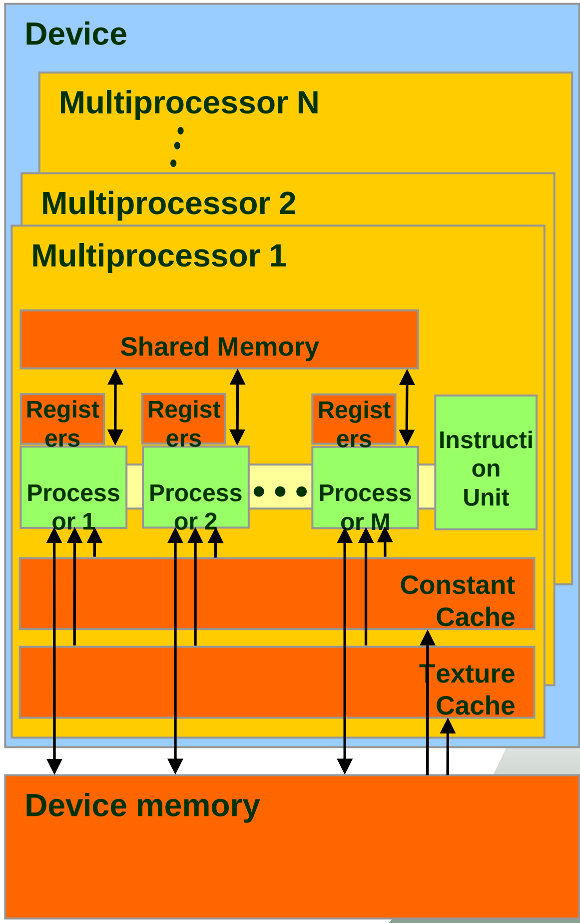
- DHS, DoD, and the intelligence community, who have armies of analysts searching text on a daily basis.





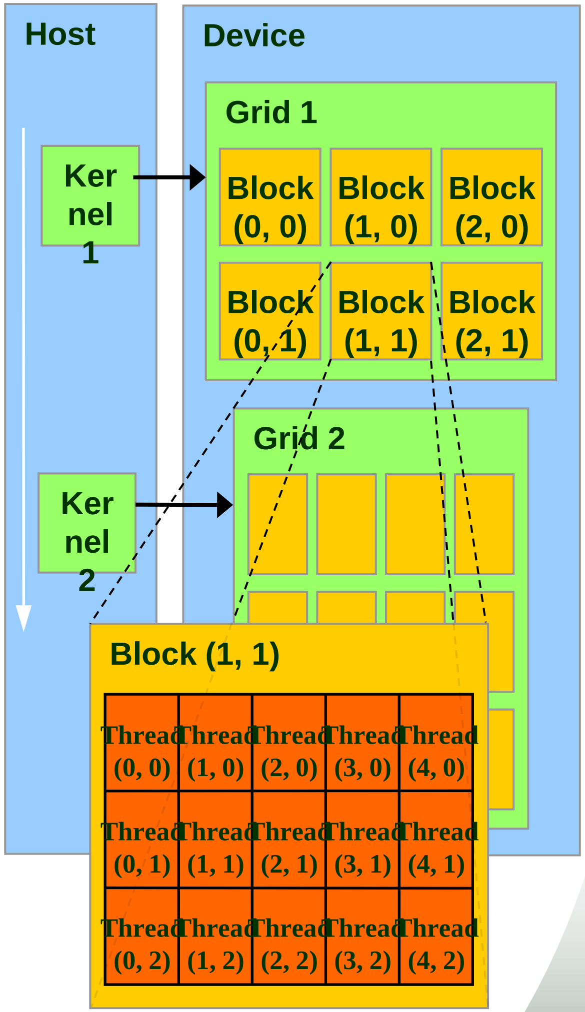
GPU Computing

- A new computing approach whereby hundreds of streaming processors (SP) on a GPU chip simultaneously communicate and cooperate to solve complex computing problems
- GPU as a coprocessor that
  - Has its own DRAM memory
  - Communicate with host (CPU) through bus (PCIe)
  - Runs many threads in *parallel*
- GPU threads
  - GPU threads are extremely lightweight (almost no cost for creation/context switch)
  - GPU needs at least several thousands threads for full efficiency
- NVIDIA CUDA provides C like program language to support the approach.



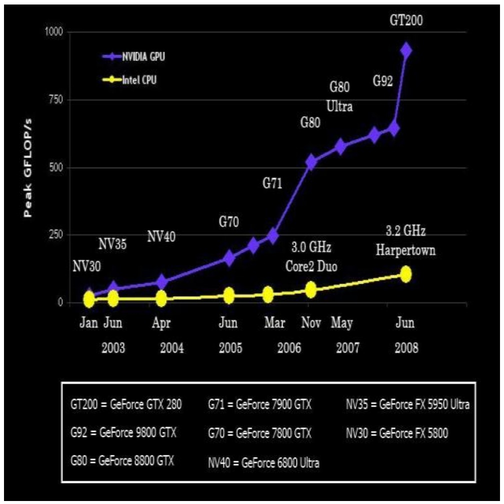
NVIDIA CUDA

- C-like programming language developed by NVIDIA™ in order to ease programming on NVIDIA G80 or above GPU core.
- Higher Abstraction allowing programmers to focus on the algorithm, not the hardware
- Portability between different CUDA enabled NVIDIA™ GPUs
- Reduced learning curve over alternative options for programming on a GPU
- These features make using CUDA highly attractive over other methods used for GPU programming



GPU Computing (Heterogeneous HPC)

A new computing approach whereby hundreds of streaming processors (SP) on a GPU chip simultaneously communicate and cooperate to solve complex computing problems



High performance and massively parallel:
CPU : ~4 cores (Quad Core) (30~40 GFLOPS)
GPU : ~240 cores (Nvidia GTX 280) (933 GFLOPS)
Cost Efficient:
GPUs: $400~$700
Quad Core CPU: $1000)
High memory bandwidth: CPU: 21 GB/s; GPU: 142 GB/s
Energy Efficient:
Simple architecture optimized for compute intensive task and energy efficiency.
GPU: 0.21w/GFLOPS; CPU: 1.43w/GFLOPS

$8,000

$1,000,000

GPU Based Large Scale Text Mining

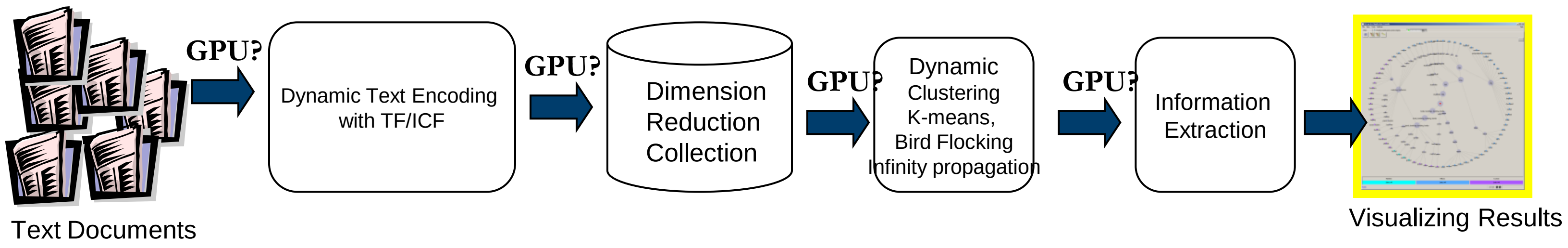
GPU computing provides a capability for text mining of terabyte-scale unstructured text corpora for prompt decision-making.

Challenges

- Bottleneck for GPU communication (PCIe) between host computer and GPU board. Synchronous direct memory access (DMA) transfers between the GPU and CPU.
- Hard to precisely anticipate ahead of time the potential speedup gains by porting TM functions on GPU
- Don’t know if there are any potentially insurmountable difficulties in exploiting the GPU for TM problems

GPU for Four Major Text Mining Functions

- **Research Questions:**
  - **Will the single precision of GPU computing impact the precision of the TM results?**
  - **Are existing text mining algorithms suitable for porting to the GPU?**
  - **How significant is the speedup when using GPU?**
- **Research Tasks:**
  - Text encoding and Dimension Reduction**:**
    - **Proof TFICF GPU implementation can speedup encoding and maintain the similarity calculation precision.**
  - Document Clustering**:**
    - **Can GPU's faster computation capability speedup massive document matrix dimension reduction for Clustering?**
  - Information Extraction**:**
    - **Prove the feasibility of Porting Information Extraction Algorithms to the GPU for speedup the process**



Text Encoding: Convert Text to Vectors

- The Vector Space Model
  - The Vector Space Model (VSM) is a way of representing natural language documents through the words that they contain
  - Developed by Gerard Salton in the early 1960s
  - It is a standard technique in Information Retrieval
  - One of the most widely used model

- Text Representation
  - Documents and queries are represented in a high-dimensional space
  - Each dimension corresponds to a term in the document
  - A vocabulary is built from all the words in all documents in the collection
  - Documents and queries are both vectors

Standard Information Retrieval

Document 1

The Army needs senor technology to help find improvised explosive devices

Terms

Army
Sensor
Technology
Help
Find
Improvise
Explosive
device

Document 2

ORNL has developed sensor technology for homeland defense

ORNL
develop
sensor
technology
homeland
defense

Document 3

Mitre has won a contract to develop homeland defense sensors for explosive devices

Mitre
won
contract
develop
homeland
defense
sensor
explosive
devices

Term List

Army
Sensor
Technology
Help
Find
Improvise
Explosive
Device
ORNL
develop
homeland
Defense
Mitre
won
contract

Vector Space Model

	Doc 1	Doc 2	Doc 3
Army	1	0	0
Sensor	1	1	1
Technology	1	1	0
Help	1	0	0
Find	1	0	0
Improvise	1	0	0
Explosive	1	0	1
Device	1	0	1
ORNL	0	1	0
develop	0	1	1
homeland	0	1	1
Defense	0	1	1
Mitre	0	0	1
won	0	0	1
contract	0	0	1

Standard Information Retrieval

Vector Space Model

	Doc 1	Doc 2	Doc 3
Army	1	0	0
Sensor	1	1	1
Technology	1	1	0
Help	1	0	0
Find	1	0	0
Improvise	1	0	0
Explosive	1	0	1
Device	1	0	1
ORNL	0	1	0
develop	0	1	1
homeland	0	1	1
Defense	0	1	1
Mitre	0	0	1
won	0	0	1
contract	0	0	1

TFIDF

$$W_{ij} = \log_2 \left(f_{ij} + 1\right) * \log_2 \left(\frac{N}{n}\right)$$

Doc 1	Doc 2	Doc 3
0.1436	0	0
0	0	0
0.053	0.053	0
0.1436	0	0
0.1436	0	0
0.1436	0	0
0.053	0	0.053
0.053	0	0.053
0	0.1436	0
0	0.053	0.053
0	0.1436	0
0	0.1436	0
0	0	0.1436
0	0	0.1436
0	0	0.1436
0	0	0.1436

Example

Document 1
Document 2
Document 3
Document 4
Document 5
Document 6
Document 7
Document 8

Self-similarity of a document list

0.0039	8.6098	7.1703	10.0571	12.3431	17.466	18.1869	9.2297
8.6098	0.0028	7.5595	7.2158	12.4792	16.0304	15.5062	9.6361
7.1703	7.5595	0.0014	8.6879	11.4787	16.8117	17.8836	8.5396
10.0571	7.2158	8.6879	0	13.5047	16.7089	16.0021	10.5474
12.3431	12.4792	11.4787	13.5047	0.0096	19.5937	19.9289	12.7648
17.466	16.0304	16.8117	16.7089	19.5937	0.0175	22.1899	17.7616
18.1869	15.5062	17.8836	16.0021	19.9289	22.1899	0.0078	18.3482
9.2297	9.6361	8.5396	10.5474	12.7648	17.7616	18.3482	0.0028

Issues

- Every added or removed document from the set requires recalculation of the entire VSM

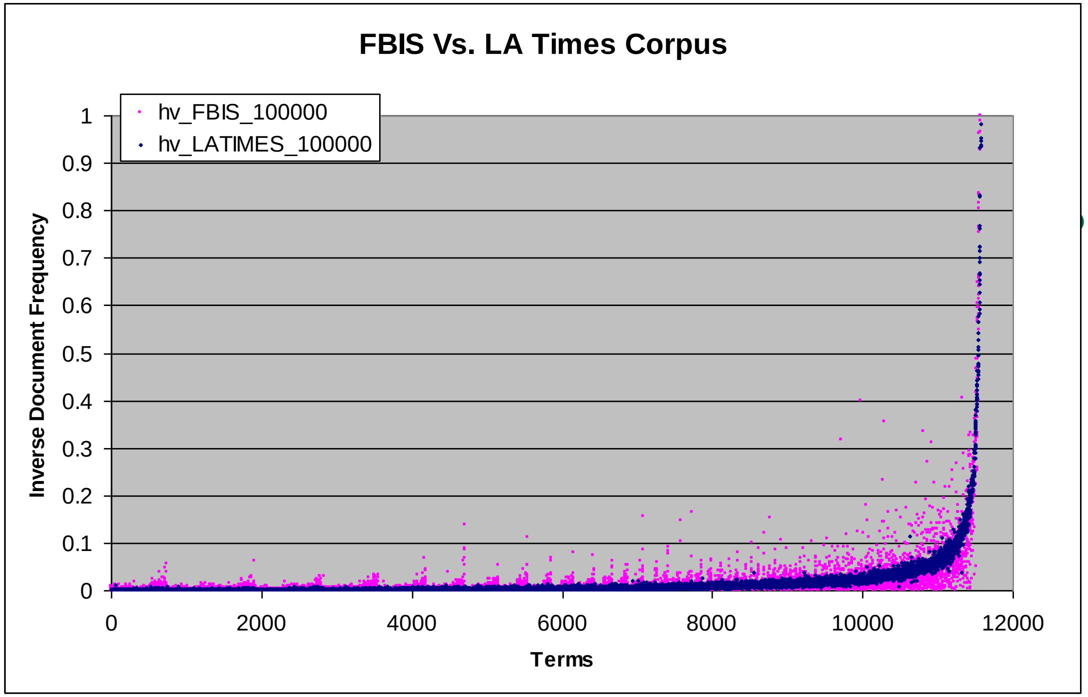
$$W_{ij} = \log_2\left(f_{ij} + 1\right) * \log_2\left(\frac{N}{n}\right)$$

Document Set must be known before VSM can be calculated

  - TFIDF not practical for dynamic data
  - Requires sequential processing
  - Not good for a distributed agent approach

- The limitation of GPU restrict the applying GPU for TFIDF calculation
  - Need copy all required data into GPU memory before computing
  - Maximum GPU memory (1GB), not enough space for coping whole document collection into GPU memory

Number-Frequency Relationship of words[1]



- **Chi Square Test shows the two distributions are same**
- **Samples from the distributions are the same**

[1] Estoup, *Games Stenographies* (1916), Joos, *Language XII* (1936), Zipf, *Human Behavior and the Principle of Least Effort* (1949)

Replace IDF with reference corpus distribution

$$W_{ij} = \log_2 \left(f_{ij} + 1 \right) * \log_2 \left(\frac{C+1}{c+1} \right)$$

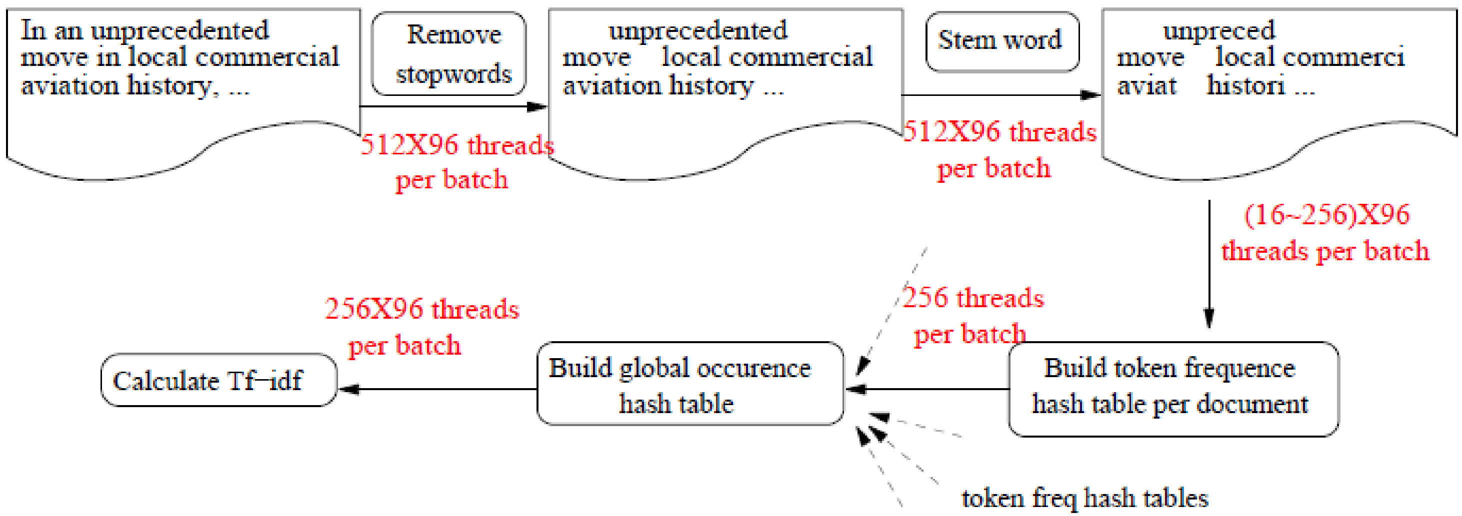
C is the number of documents our reference corpus, and c is the number of documents in the reference corpus where Tj occurs at least once.

- **The reference corpus contains 239,864 unique terms from 255,749 documents of the TREC Text Research Collection Vol. 5.**
- **Allows us to create a vector from an individually streamed document**

Why this matters

- We can now generate an accurate vector directly from a text document
- That vector can be generated where ever the document resides
- The TFICF is suitable to be implemented on GPU
  - There are no need to copy the whole document collection into GPU memory in one time
  - GPU memory access four times faster than CPU memory access
  - The GPU memory is large enough for store the whole ICF database

Methods



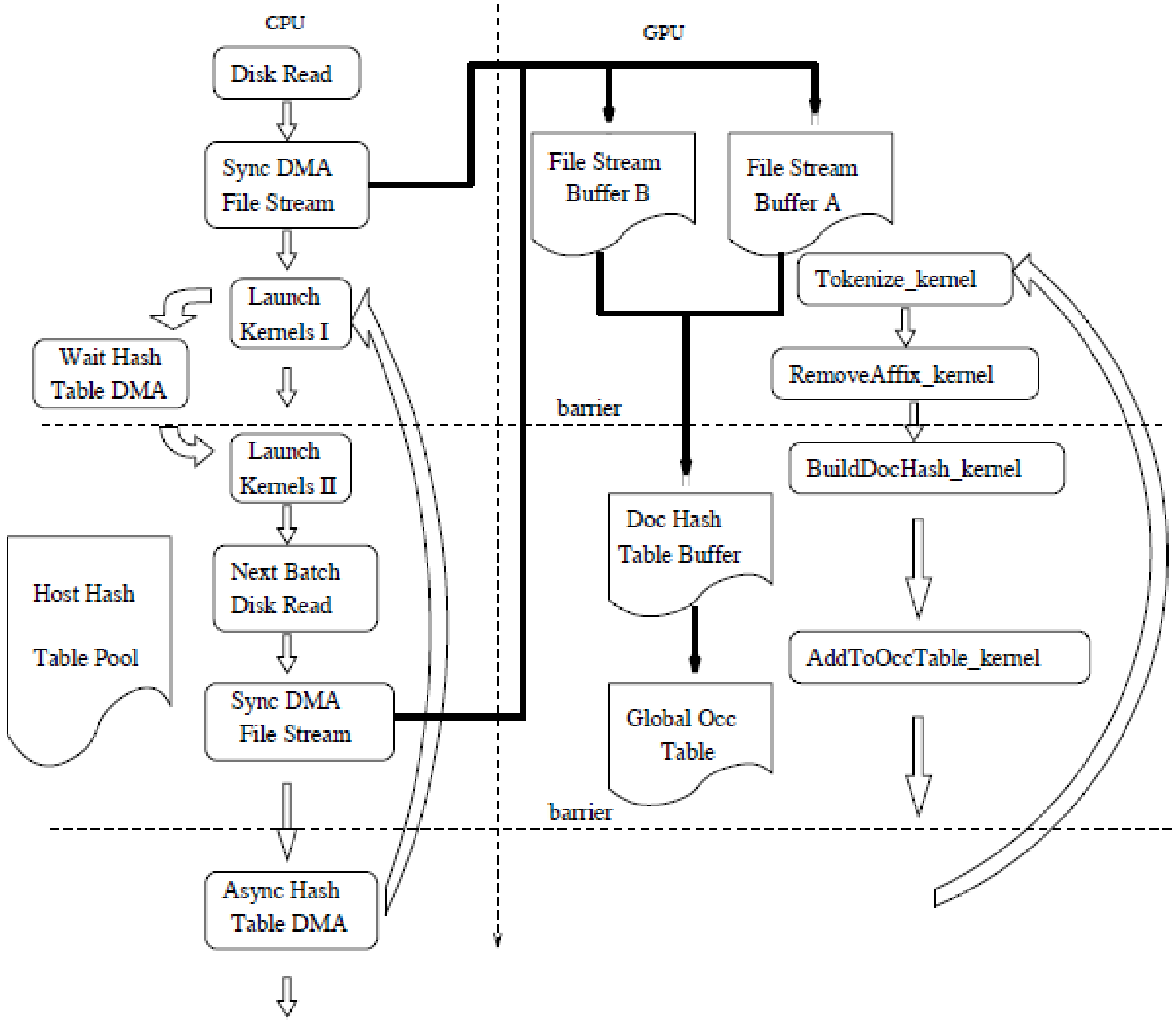
1. Remove stop words(noise words), such as “I”, “the”, “and”;
2. Strip affixes, such as prefixes(“pre”, “kilo”) and suffixes(“tion”, “ize”);
3. Hash table per document;
4. All document hash tables reduce to one global hash table;
5. Walk through the document hash table, calculate each term’s tfidf.

The key is to build the two hash tables.

Computer Sci
NC STATE UN

OAK RIDGE National Laboratory

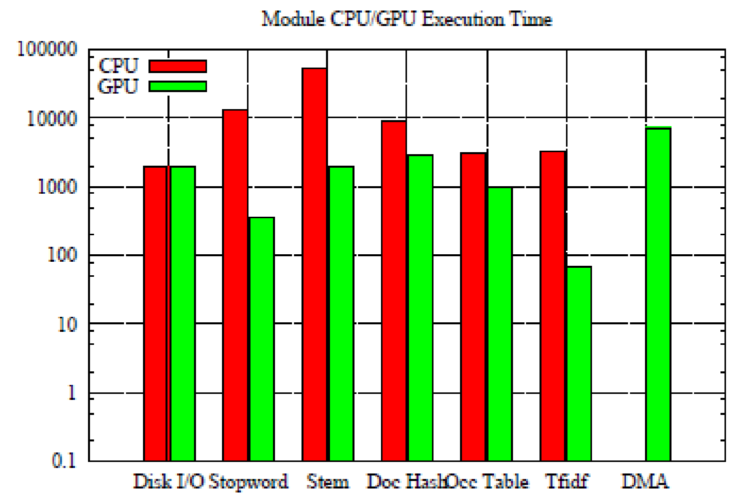
Double Buffer Framework

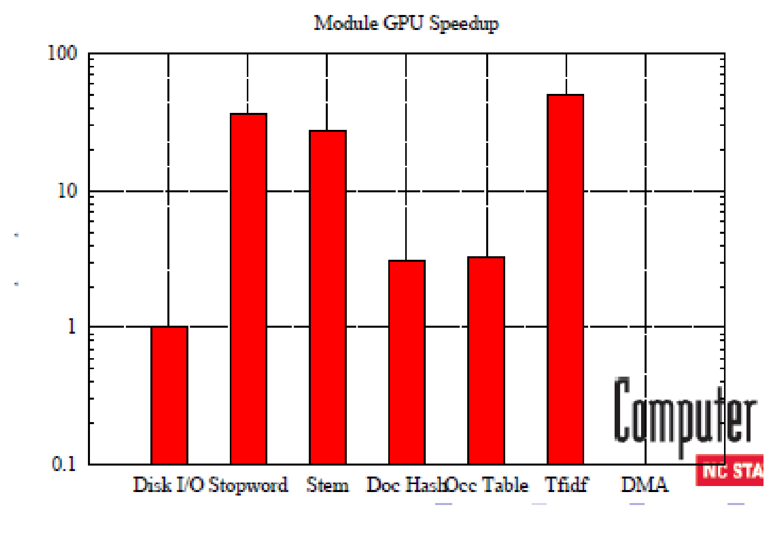


OAK RIDGE National Laboratory

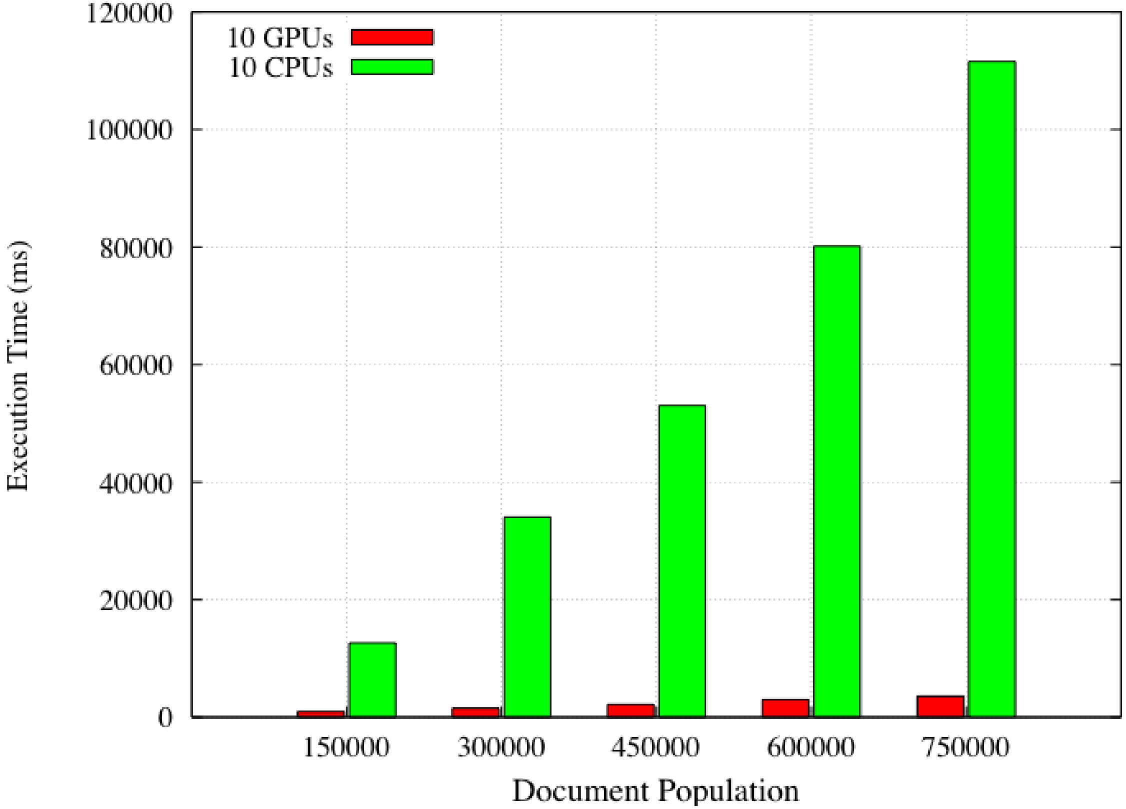
Individual Speedup (Single Batch, No Double Buffer)

- CPU:
  - Single-threaded C++;
  - 2.0G AMD Athlon Dual-core;
  - 2G Memory;
- GPU:
  - Geforce GTX 280;
  - 240 SPs, 1G global memory;
  - PCI Express x16;
- Speedups
  - Disk I/O: 1X
  - Stopword: 36X
  - Stem Token: 27X
  - Doc Hash: 3.1X
  - Occ Table: 3.2X
  - Tfidf: 50X
  - Overall: 6X





Document Processing -- TFICF

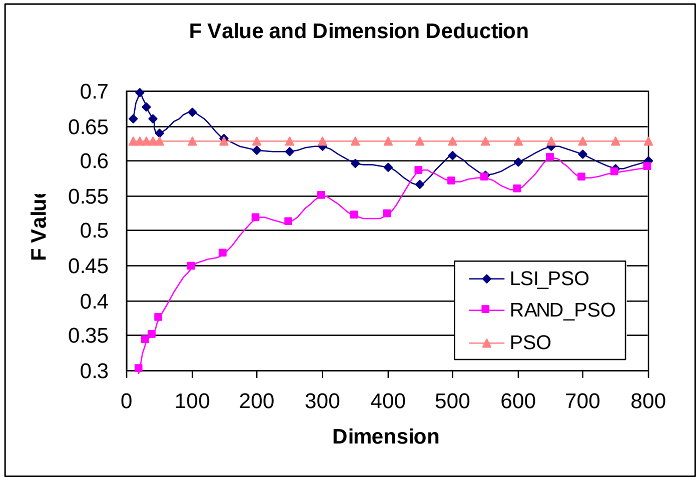


- **Convert documents to TF-ICF vectors : part of information retrieval process**
- **30X overall speedup at 10 GPUs**

Document Dimension Reduction on GPU

- **Document dimension reduction can reduce the size of document dataset.**

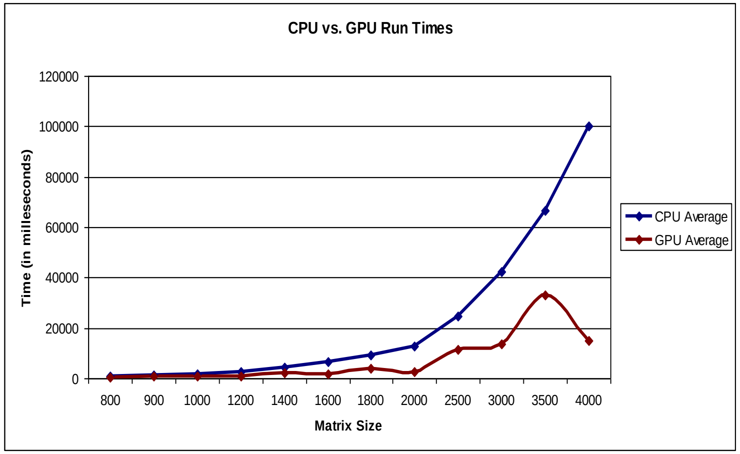
  Latent Semantic Indexing (LSI) for Document Dimension Reduction
  - **a mathematical method used for finding relationships between text within a collection of documents.**
  - **LSI can improve clustering result when reduced document vector to small dimensionality.**
  - **Major drawback: is high computational cost. $O(m^2n)$ for n×m matrix**
- **Random Projection**
  - **Time complexity O(mkn)**



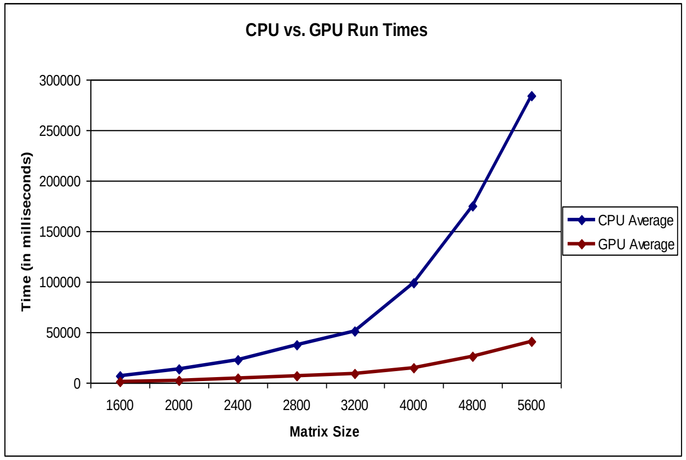
Document Clustering Precision after Dimension Reduction

Document Dimension Reduction on GPU

- Aim to decrease the time of Singular Value Decomposition. SVD is the computationally expensive portion of LSA.
- Increases the GPU one time clustering capability nearly 5,000 times (Assume 10k dimension document collection)



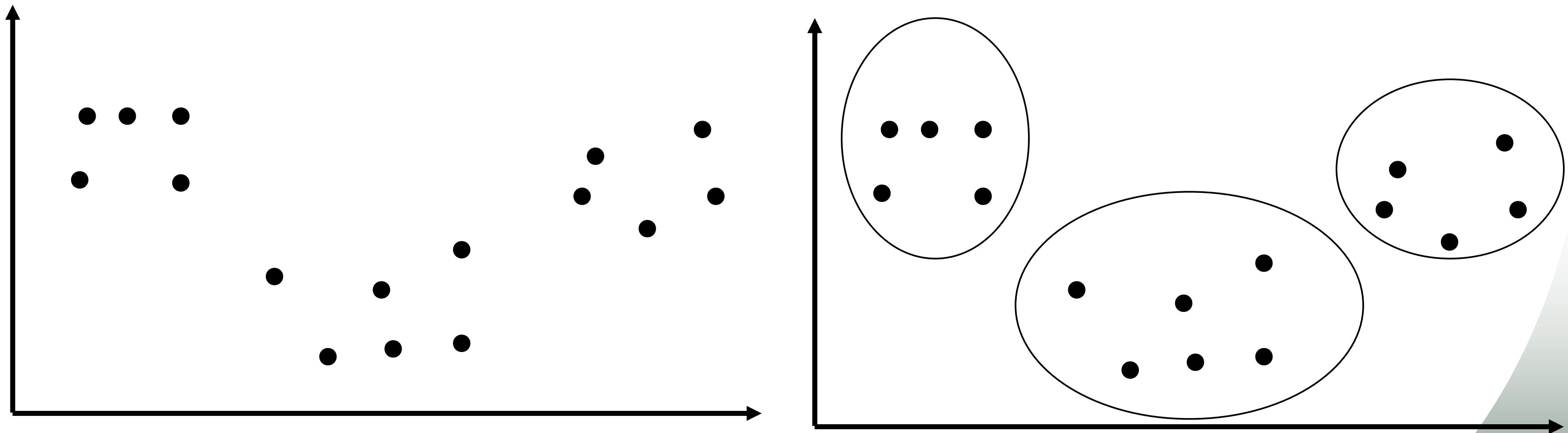
Test results for increasingly large matrices, up to 4000 x 4000



Test results for increasingly large matrices with dimension divisible by 16, up to 5600 x 5600

Clustering

- **Partition unlabeled individual examples into disjoint clusters, such that:**
  - **Examples within a cluster are very similar**
  - **Examples in different clusters are very different**



Standard Textual Clustering

Vector Space Model

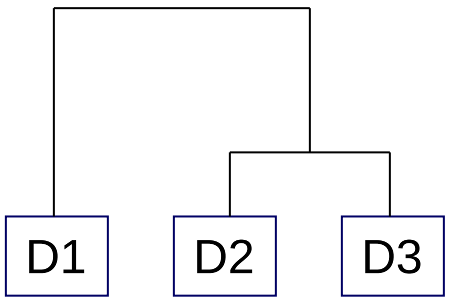
	Doc 1	Doc 2	Doc 3
Army	1	0	0
Sensor	1	1	1
Technology	1	1	0
Help	1	0	0
Find	1	0	0
Improvise	1	0	0
Explosive	1	0	1
Device	1	0	1
ORNL	0	1	0
develop	0	1	1
homeland	0	1	1
Defense	0	1	1
Mitre	0	0	1
won	0	0	1
contract	0	0	1

Dissimilarity Matrix

	Doc 1	Doc 2	Doc 3
Doc 1	100%	17%	21%
Doc 2		100%	36%
Doc 3			100%

Documents to Documents

Cluster Analysis



Most similar documents

TFIDF

$$W_{ij} = \log_2 (f_{ij} + 1) * \log_2 \left(\frac{N}{n} \right)$$

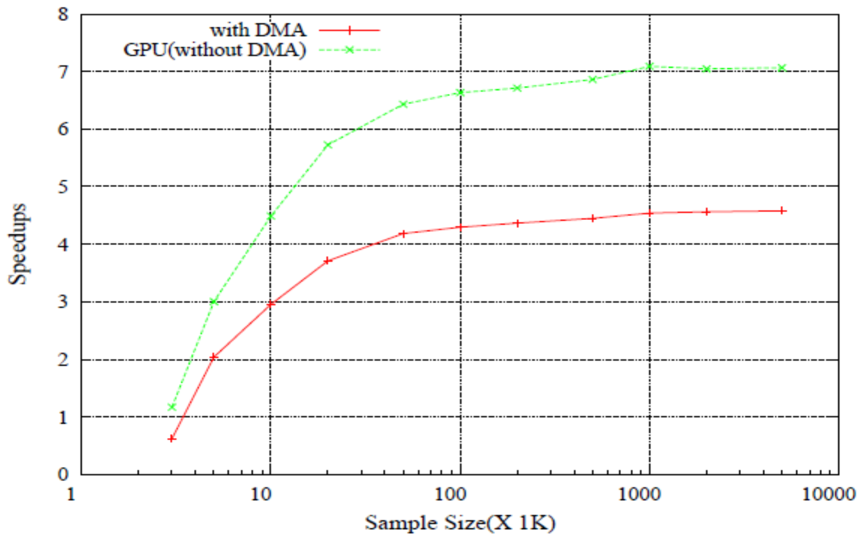
Euclidean distance

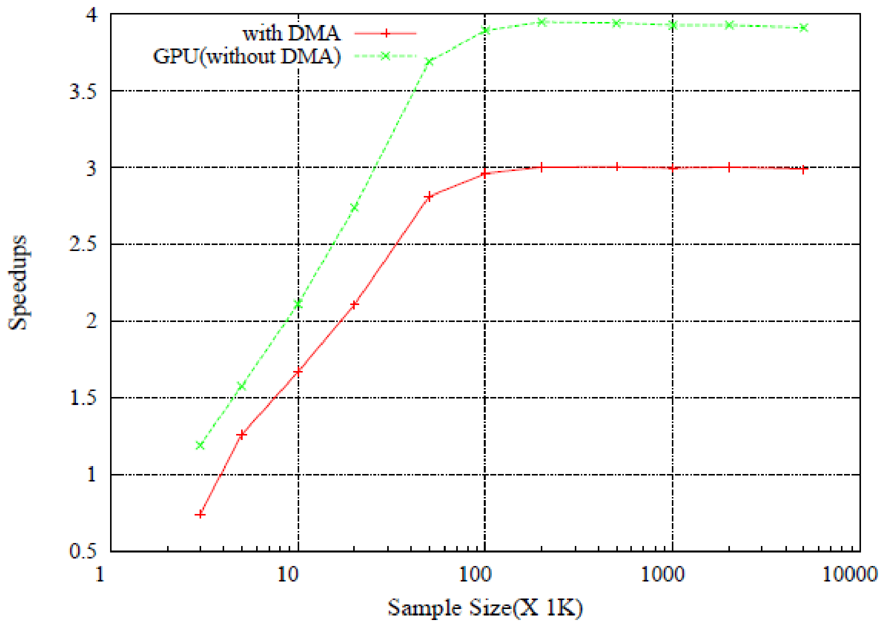
$$d_2(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2 \right)^{1/2}$$

Time Complexity

$$O(n^2 \mathrm{Log}\ n)$$

Standard Textual Clustering on GPU



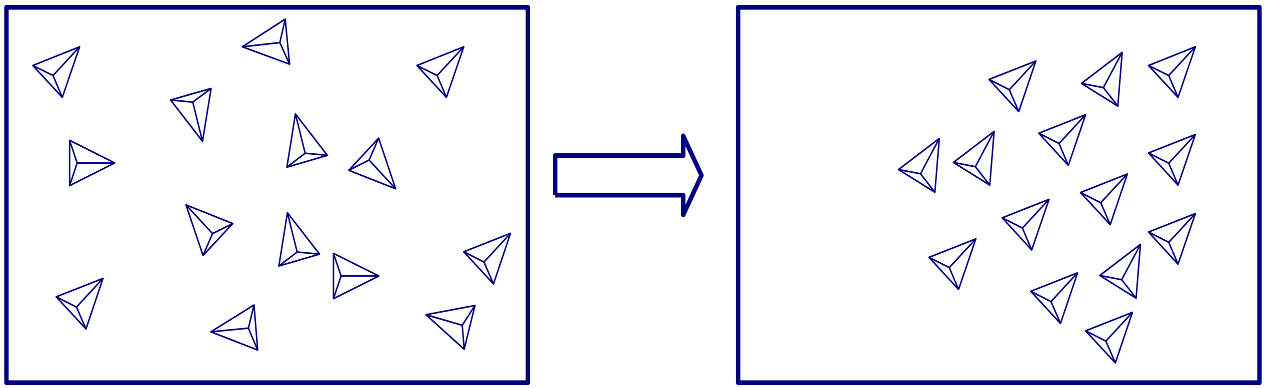


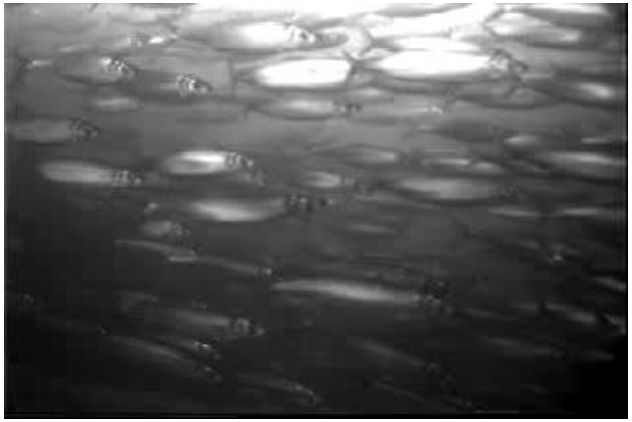
OAK RIDGE
National Laboratory

A New Clustering Algorithm Based on Bird Flock Collective Behavior

Trivial Behavior

Emergent behavior = flocking





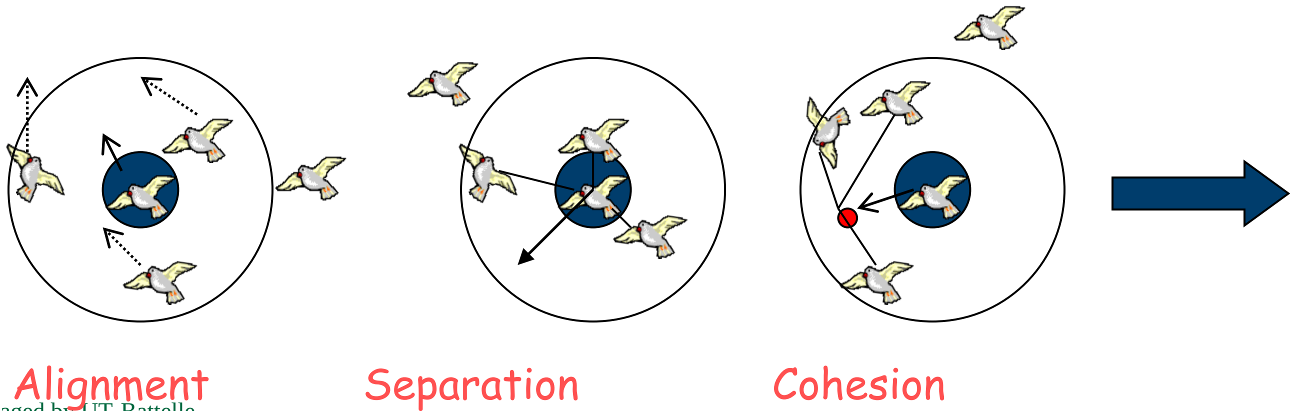
Flocking Model

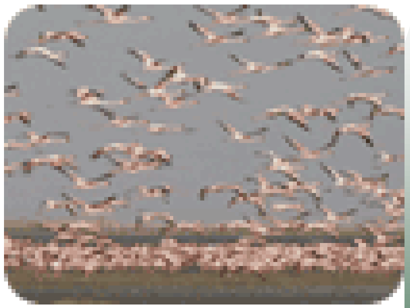
Flocking model, one of the first bio-inspired computational collective behavior models, was first proposed by Craig Reynolds in 1987.

Alignment : steer towards the average heading of the local flock mates

Separation : steer to avoid crowding flock mates

Cohesion : steer towards the average position of local flock mates





Mathematical Flocking Model

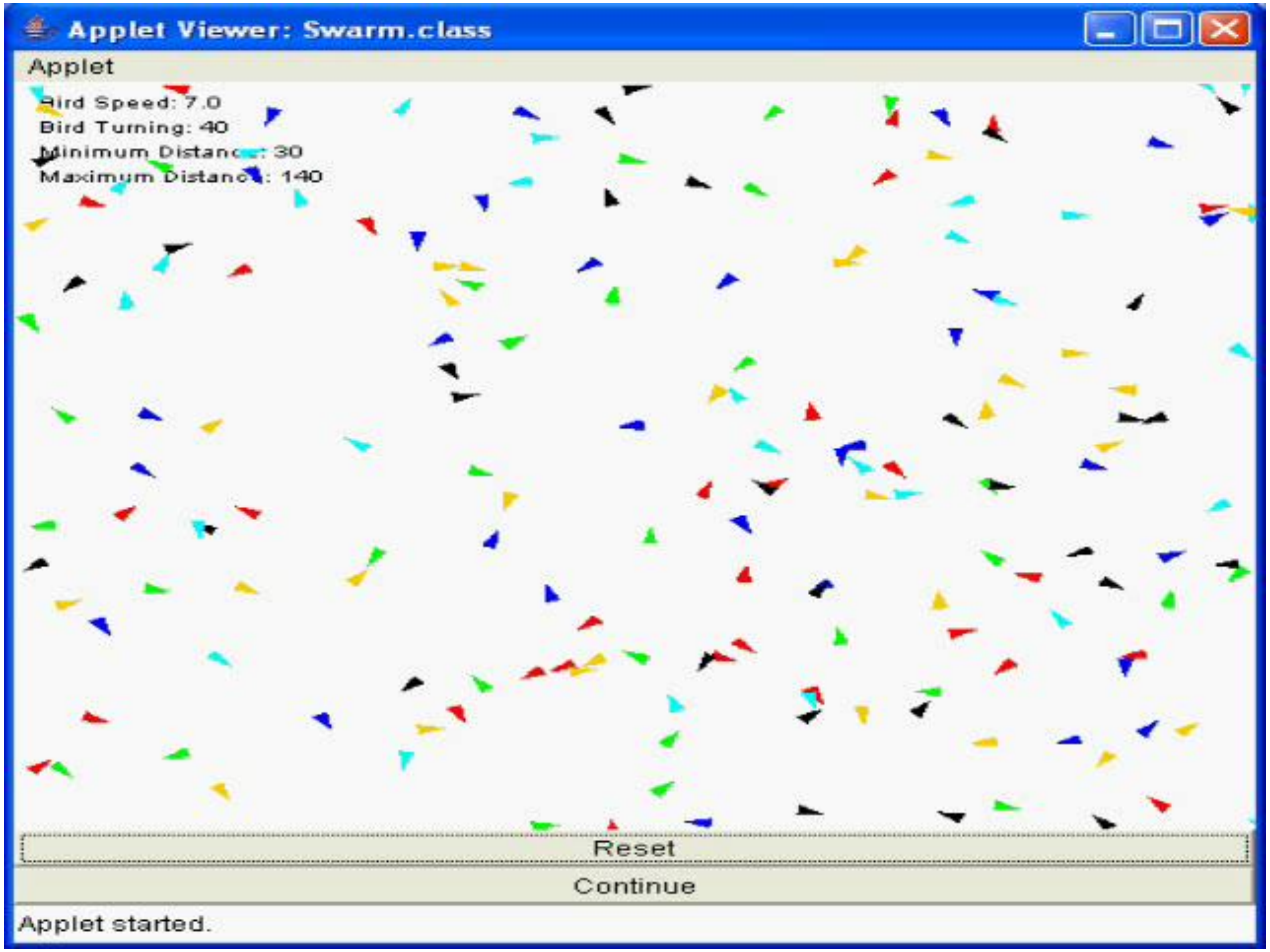
Alignment Rule: $$d(P_x, P_b) \le d_1 \cap (P_x, P_b) \ge d_2 \Rightarrow \vec{v}_{ar} = \frac{1}{n}\sum_{x}^{n} \vec{v}_x$$

Separation Rule: $$d(P_x, P_b) \le d_2 \Rightarrow \vec{v}_{sr} = \sum_{x}^{n} \frac{\overline{\vec{v}_x + \vec{v}_b}}{d(P_x, P_b)}$$

Cohesion Rule: $$d(P_x, P_b) \le d_1 \cap (P_x, P_b) \ge d_2 \Rightarrow \vec{v}_{cr} = \sum_{x}^{n} \overrightarrow{(P_x - P_b)}$$

Multiple Species Flocking (MSF) Model

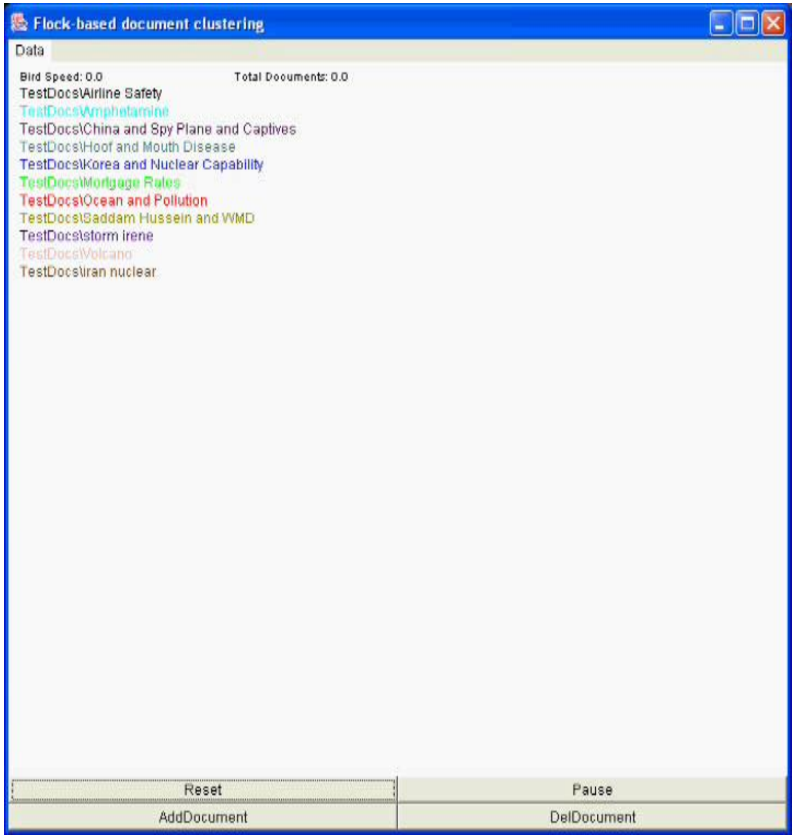
feature similarity rule: Steer away from other birds that have dissimilar features and stay close to these birds that have similar features.



Multiple Species Flocking Algorithm Swarm Intelligence Document Clustering

	Category/Topi c	Number of articles
1	**Airline Safety**	**10**
2	**China and Spy Plane and Captives**	**4**
3	**Hoof and Mouth Disease**	**9**
4	**Amphetamin e**	**10**
5	**Iran Nuclear**	**16**
6	**N. Korea and Nuclear Capability**	**5**
7	**Mortgage Rates**	**8**
8	**Ocean and Pollution**	**10**
9	**Saddam Hussein and WMD**	**10**
1 0	**Storm Irene**	**22**
1 1	**Volcano**	**8**

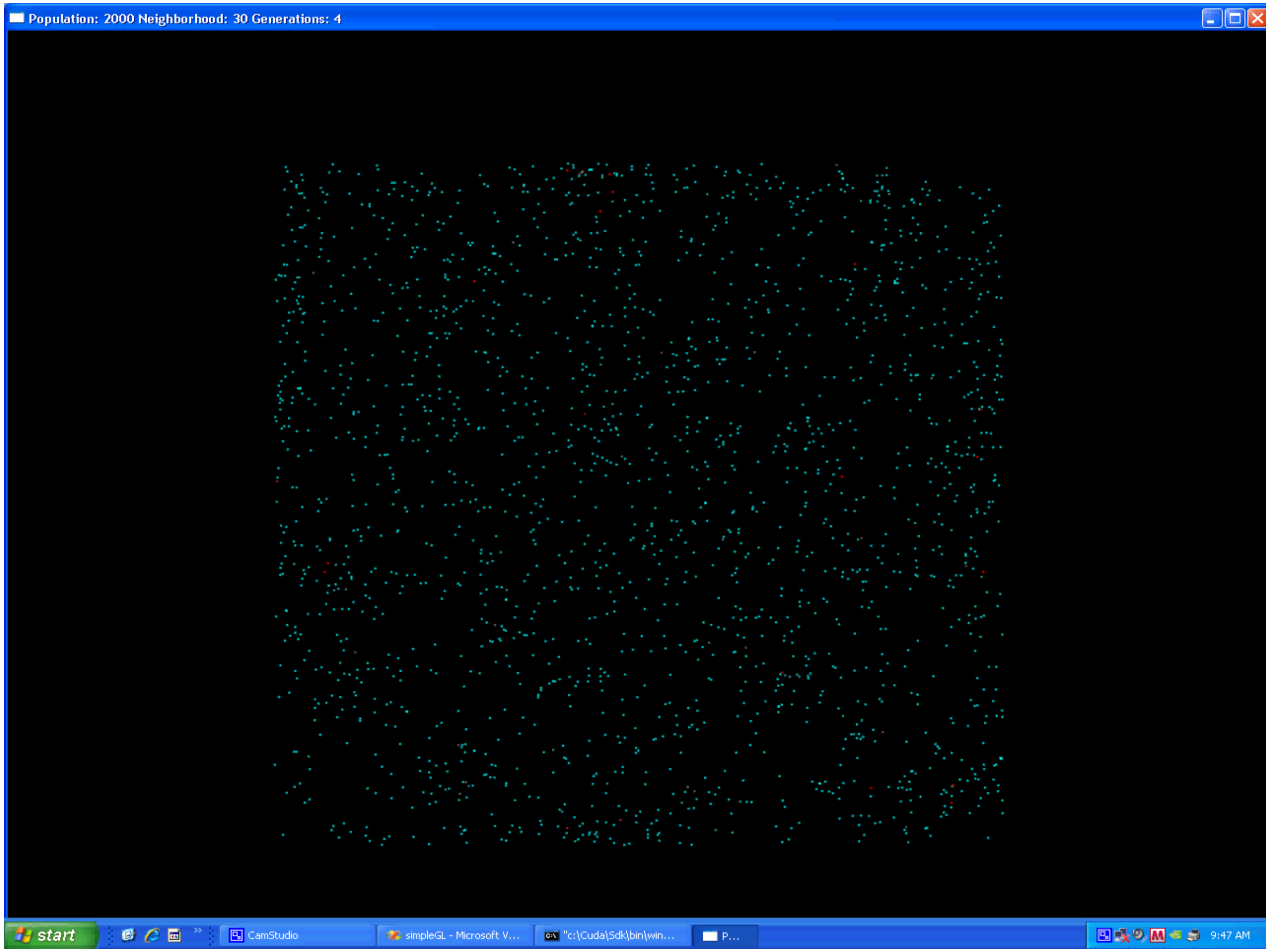
The Document collection Dataset



The clustering results of K-means, Ant clustering and MSF clustering Algorithm on synthetic* and document datasets after 300 iterations**

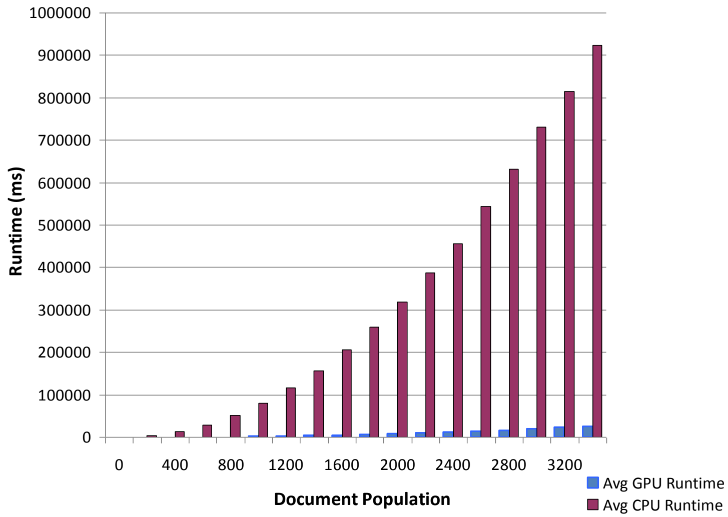
	Algorithm s	Average cluster number	Average F-measure value
Synthetic Dataset	MSF	4	0.9997
	K-means	(4)***	0.9879
	Ant	4	0.9823
Real Documen t Collection	MSF	9.105	0.7913
	K-means	(11)***	0.5632
	Ant	1	0.1623

Bird Flocking Document Clustering on GPU

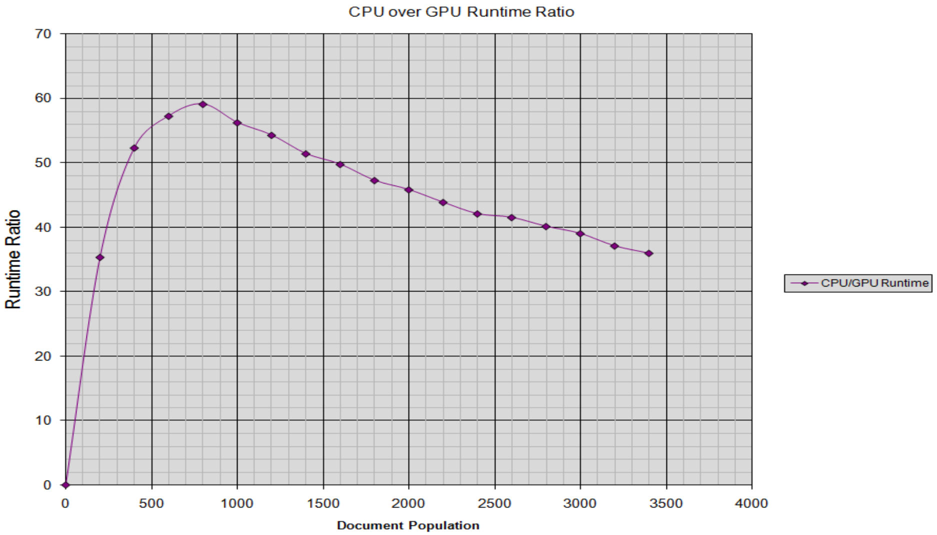


OAK RIDGE
National Laboratory

Document Clustering on GPU



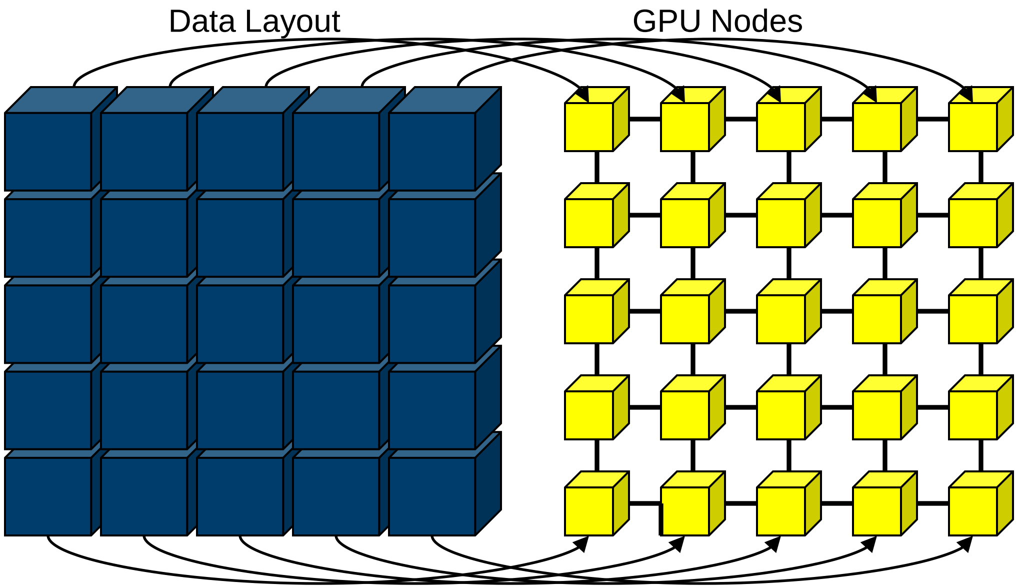
Running Time Comparing for Data Clustering on GPU and CPU



GPU speedup on Document Clustering

A GPU Programming Model for Massive Data Parallelism

New Program Model

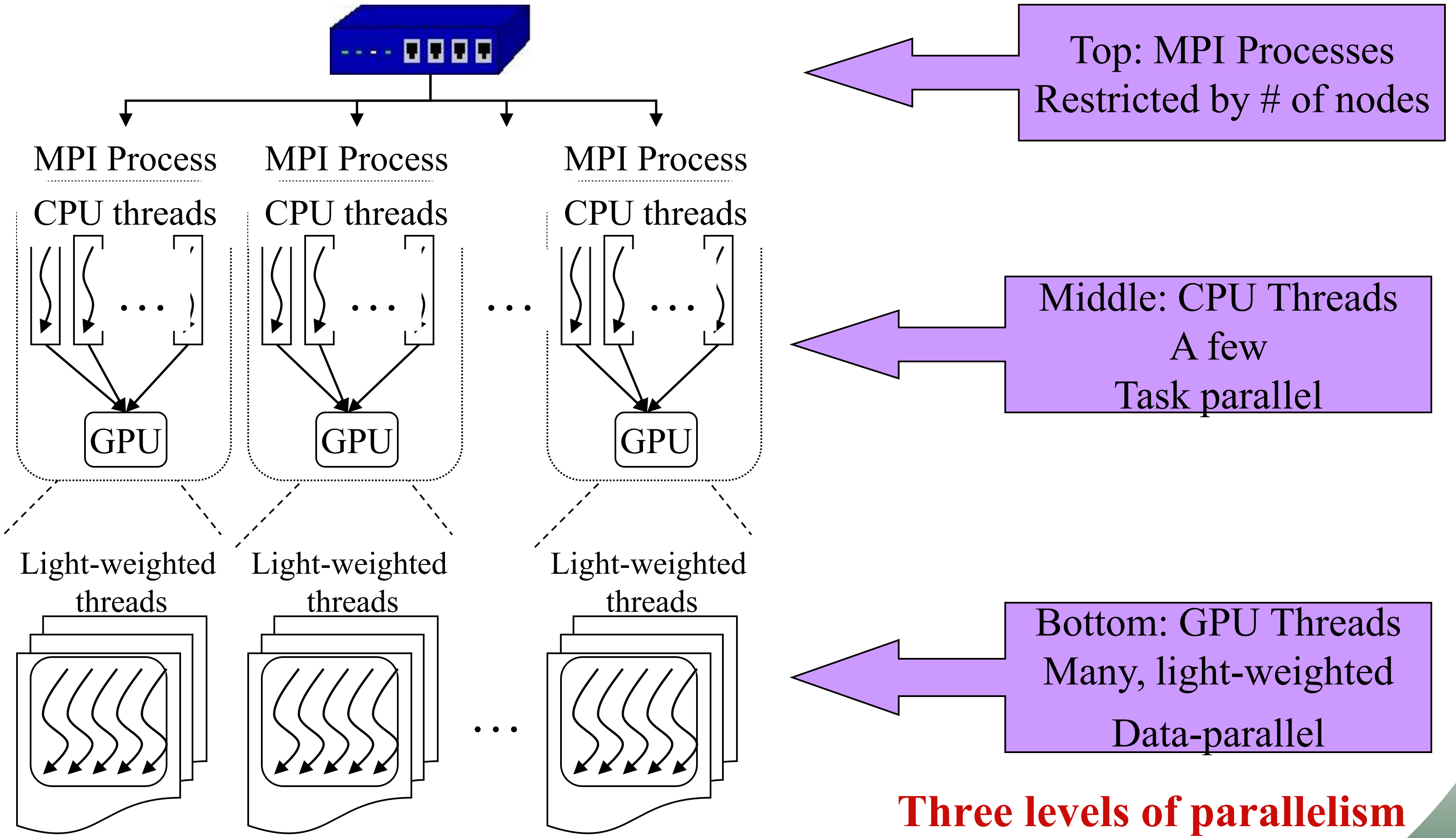


Divide/ conquer paradigm, Map massive data to distributed GPUs, Each GPU works on a portion of the problem

Massively Parallel

GPU Cluster Unit:
•4 GPUs; 3.73 TFLOPS;
•960 Processors; 408 GB/s max memory bandwidth
•16G Memory, 800W

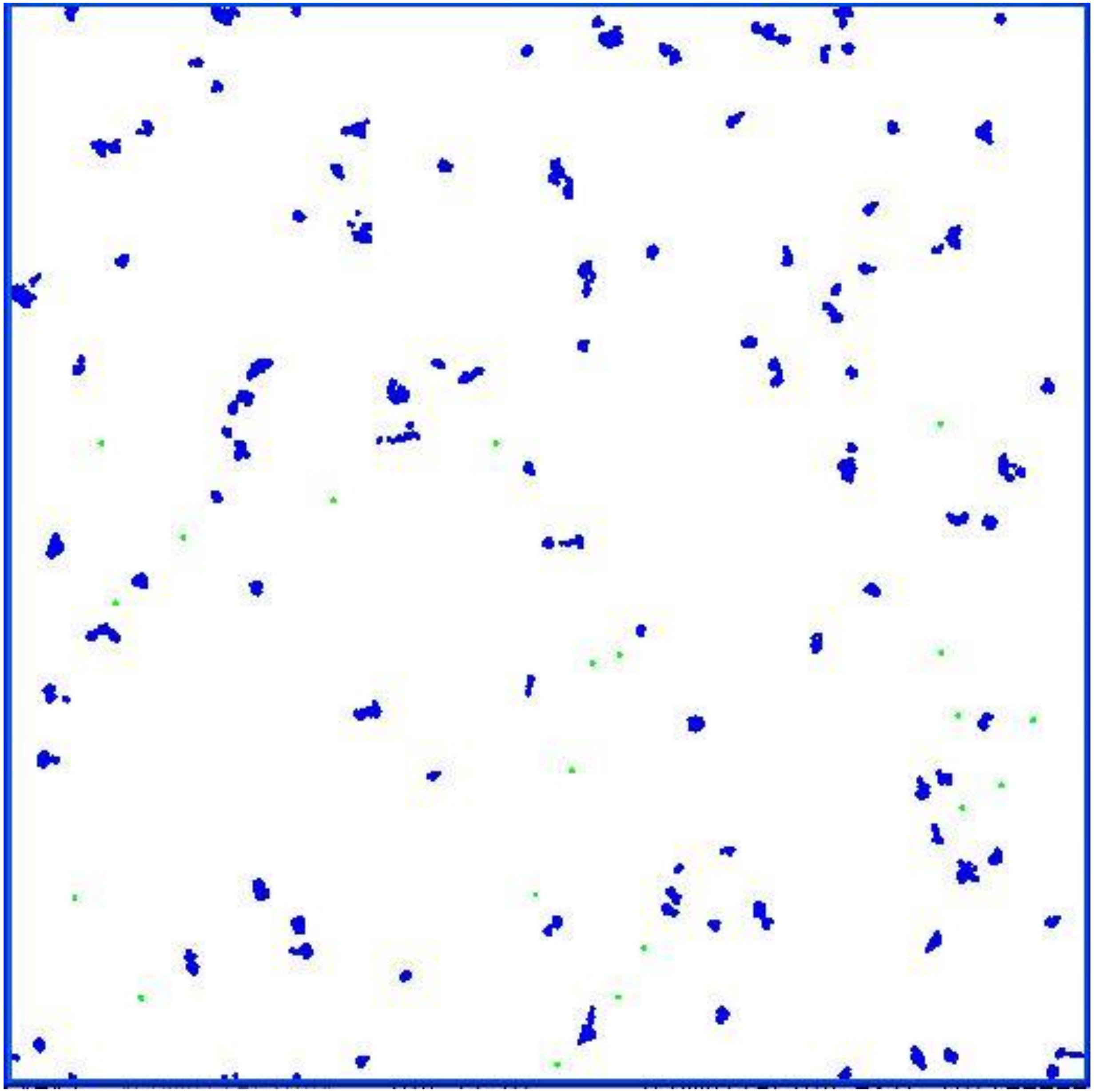
Our Target Platform

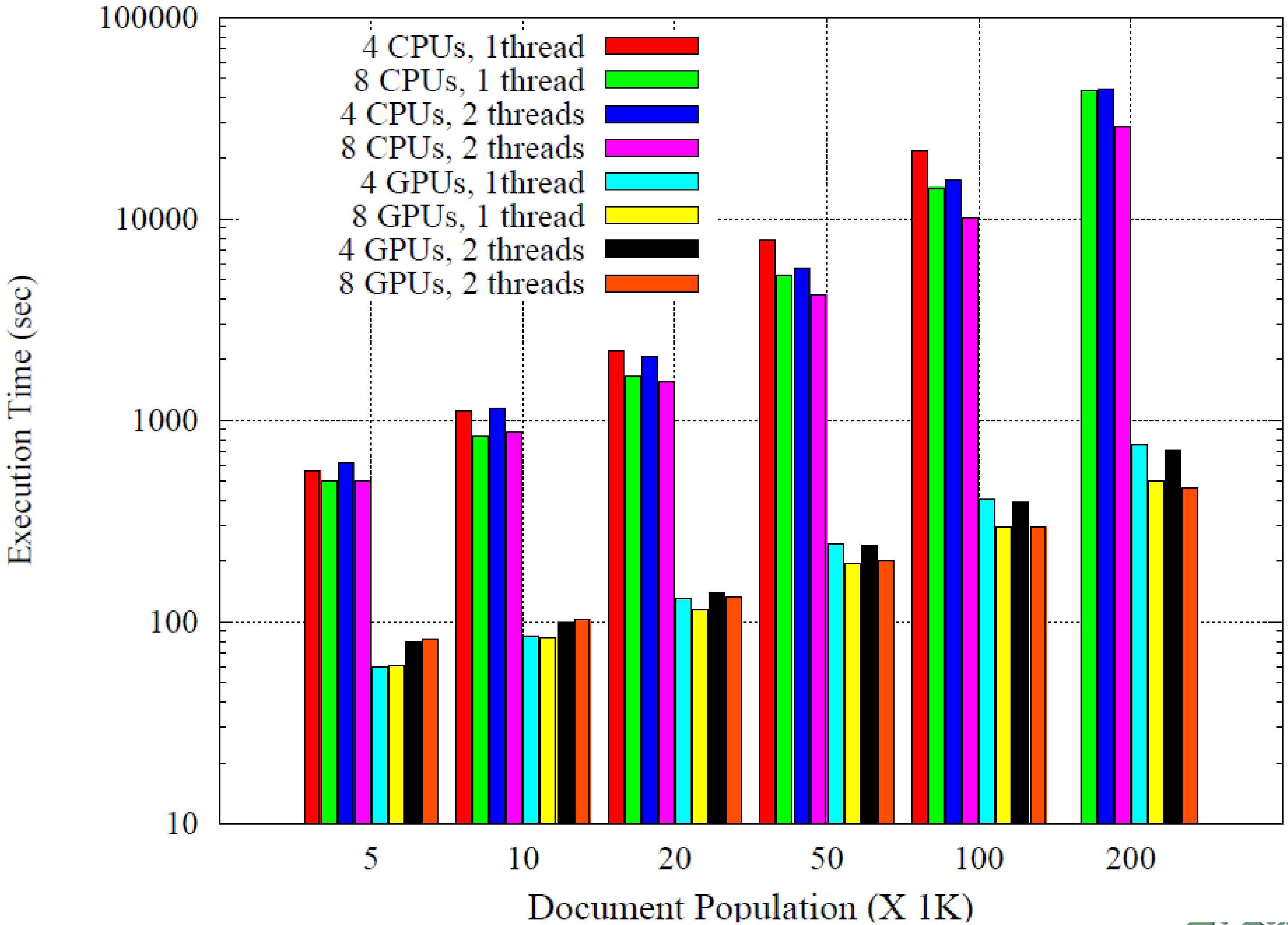


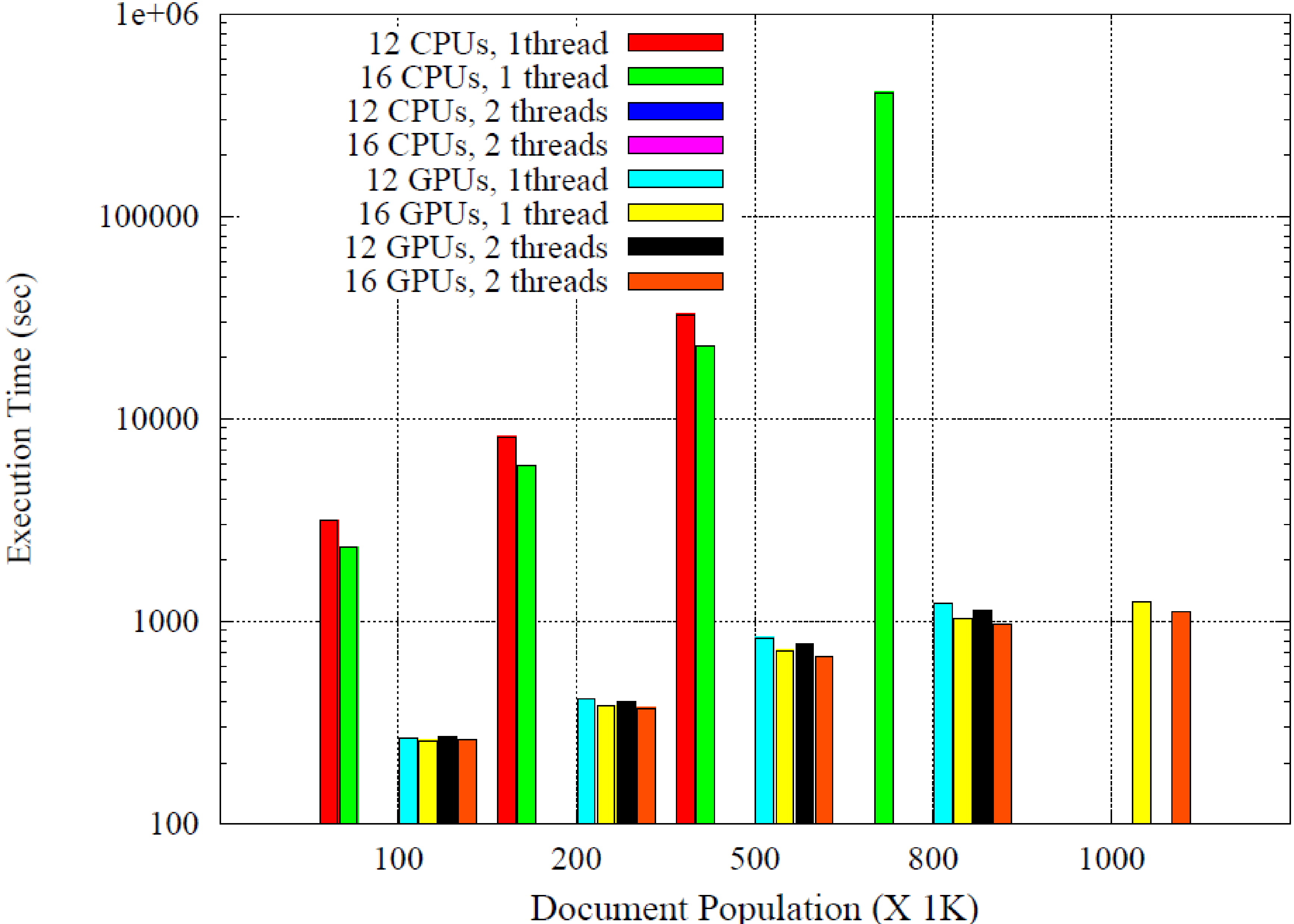
Three levels of parallelism

Experimental Results

	GPU Cluster	CPU Cluster
Nodes	10	10
CPU	AMD Athlon Dual Core	AMD Athlon Dual Core
CPU Freq.	2.0 GHz	2.0 GHz
Host Memory	1G	1G
GPU	GTX 280	N/A
Network	Giga-bit Ethernet	Giga-bit Ethernet




100000
10000
1000
100
10
Execution Time (sec)
4 CPUs, 1thread
8 CPUs, 1 thread
4 CPUs, 2 threads
8 CPUs, 2 threads
4 GPUs, 1thread
8 GPUs, 1 thread
4 GPUs, 2 threads
8 GPUs, 2 threads
5
10
20
50
100
200
Document Population (X 1K)



1e+06
100000
10000
1000
100
12 CPUs, 1thread
16 CPUs, 1 thread
12 CPUs, 2 threads
16 CPUs, 2 threads
12 GPUs, 1thread
16 GPUs, 1 thread
12 GPUs, 2 threads
16 GPUs, 2 threads
Execution Time (sec)
100
200
500
800
1000
Document Population (X 1K)

Information Extraction on GPU

Example Task:

Part-of-Speech Tagging

Did	**I**	**mention**	**that**	**we**	**surrendered**	**?**
VBD	**PRP**	**VB**	**DT**	**PRP**	**VBD**	**?**

- **Example IE model type: Conditional Random Field - looks at the conditional probability of a state sequence, s, given some observed input sequence, o.**

$$P(s \mid o) = \frac{1}{Z_o} \exp\left(\sum_{i=1}^{N} \sum_{k} \lambda_k f_k(s_{i-1}, s_i, o, i)\right)$$

- **Dynamic Programming can be used to calculate the most probable sequence.**

$$\delta_{t+1}(s_i) = \max_{s'} \left[\delta_t(s') \exp\left(\sum_k \lambda_k f_k(s', s_i, o, t) \right) \right]$$

Parallelizing the workload is not the core problem. Rather, it is keeping the model in local SIMD memory (~16KB) for evaluation of the feature functions.

Millions of Features are common. Many word-based, leading to large models, removal of even rare features hurts model accuracy. Thus, we will target methods for separately evaluating the feature functions, etc.

Prove the feasibility of Porting Information Extraction Algorithms to the GPU for speedup the process

Thanks!