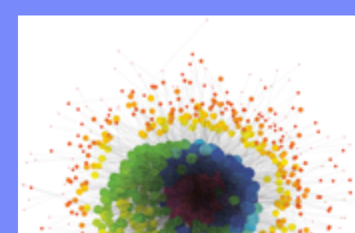
Billion-Scale Graph Analytics

Toyotaro Suzumura, Koji Ueno, Charuwat Houngkaew, Masaru Watanabe, Hidefumi Ogata, Miyuru Dayarathna, ScaleGraph Team

Tokyo Institute of Technology / JST CREST
IBM Research



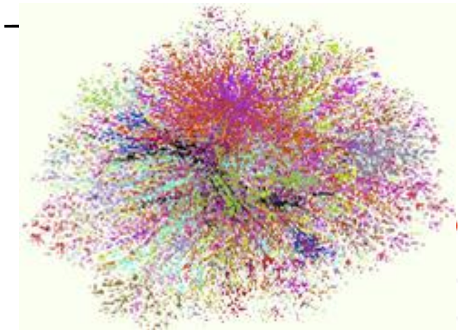
Japan Science and Technology Agency

Outline

- Introduction
- Graph500 Benchmark
- ScaleGraph: Billion-Scale Graph Analytics Library
- Time-Series Analysis for Whole Twitter Network
- Summary

Large-Scale Graph Mining is Everywhere

Cybersecurity
Medical Informatics
Data Enrichment
Social Networks
Symbolic Networks



Internet Map

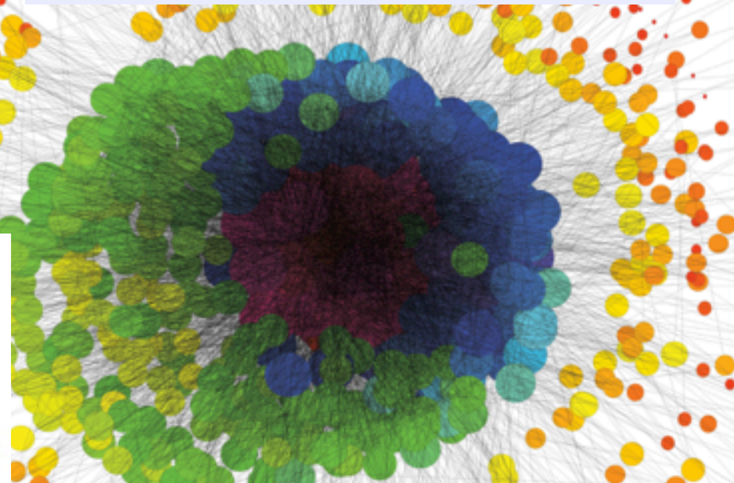
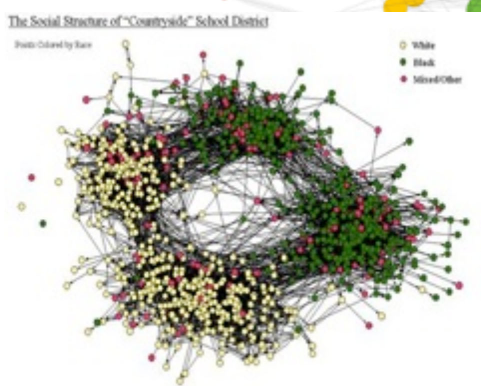
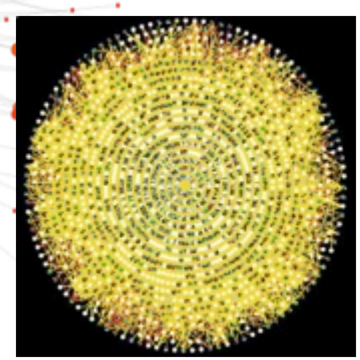


Image: Illustration by Mirko Ilic

Symbolic Networks:



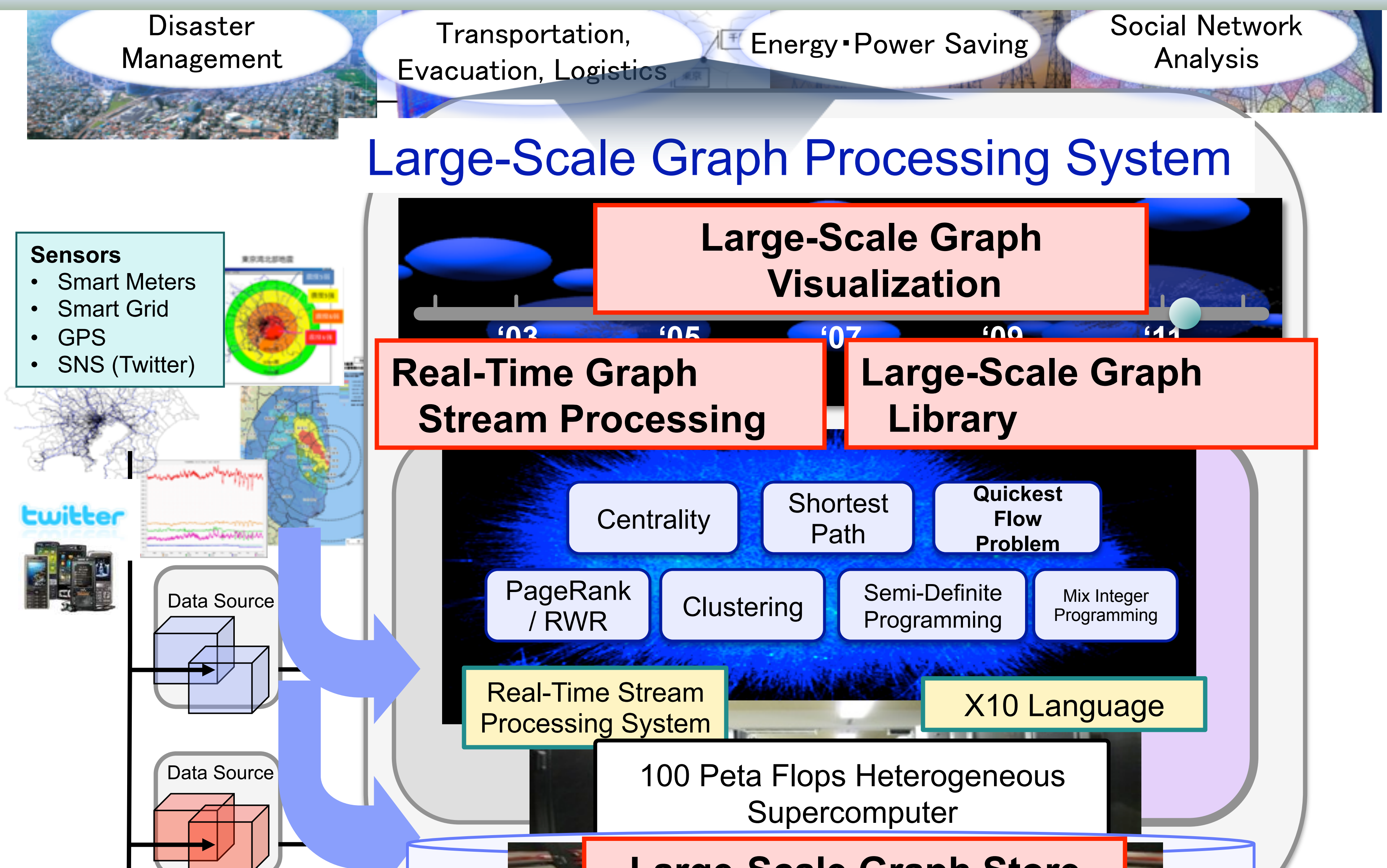
Social Networks



Protein
Interactions

Cyber Security (15 billion log entries / day for large enterprise)

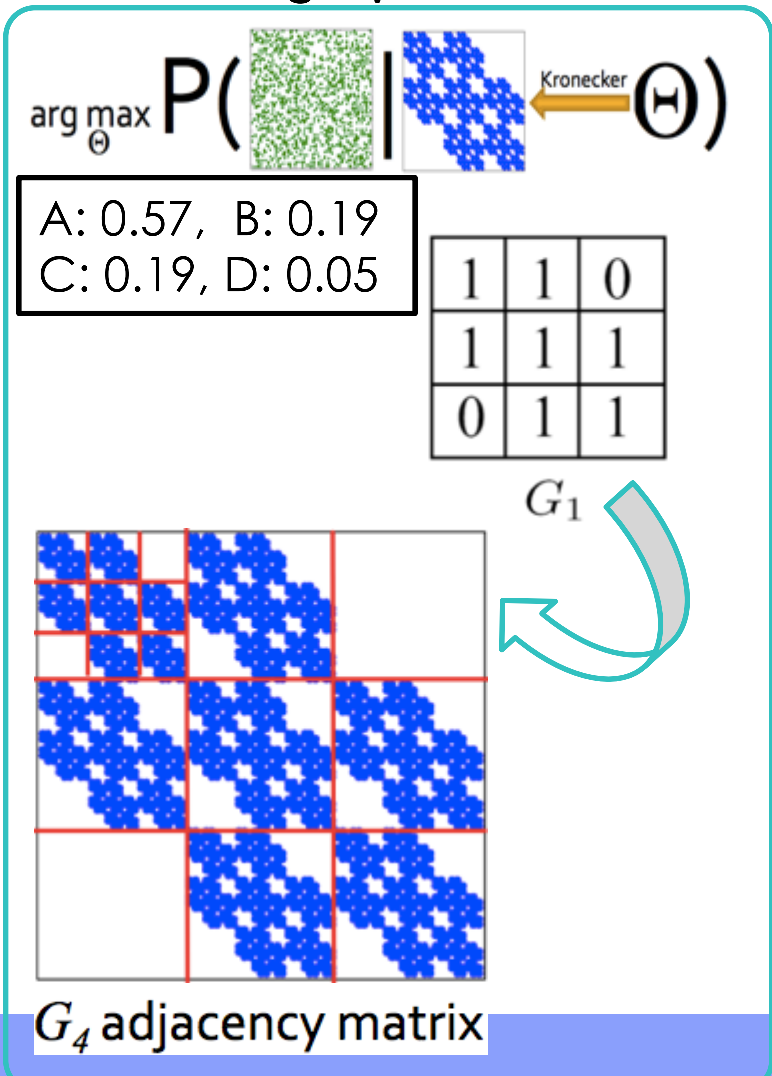
Large-Scale Graph Processing System (2011-2018)



Japan Science and Technology Agency

Graph500: Big Graph Competition

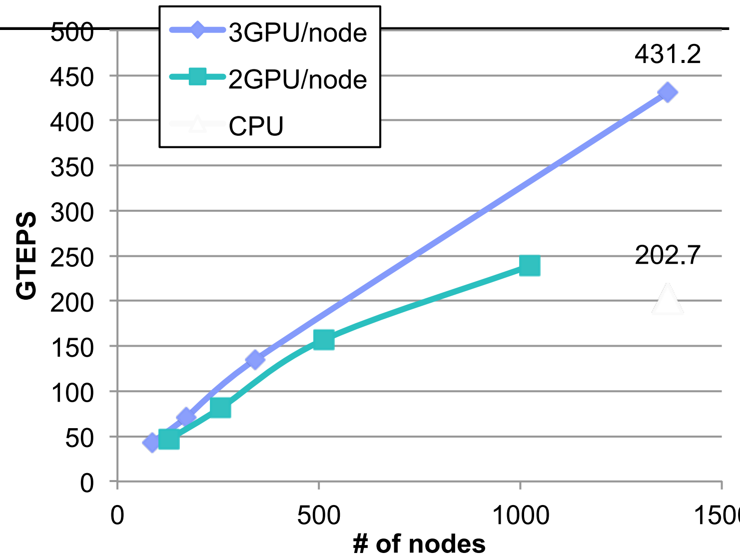
Kronecker graph



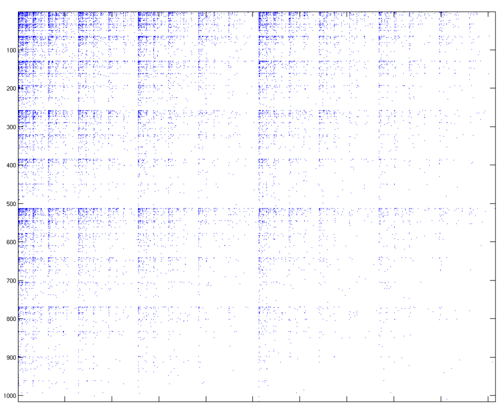
- Graph500 is a new benchmark that ranks supercomputers by executing a large-scale graph search problem.
- The benchmark is ranked by so-called **TEPS (Traversed Edges Per Second)** that measures the number of edges to be traversed per second by searching all the reachable vertices from one arbitrary vertex with each team's optimized BFS (Breadth-First Search) algorithm.

Highly Scalable Graph Search Method for the Graph500 Benchmark

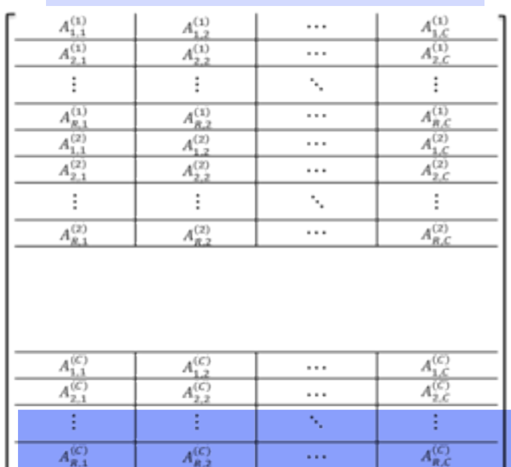
- We propose an optimized method based on 2D based partitioning and other various optimization methods such as communication compression and vertex sorting.
- We developed CPU implementation and GPU implementation.
- Our optimized GPU implementation can solve BFS (Breadth First Search) of large-scale graph with 2^{35} (34.4 billion) vertices and 2^{39} (550 billion) edges for 1.275 seconds with 1366 nodes (16392 cores) and 4096 GPUs on TSUBAME 2.0
- This record corresponds to 431 GTEPS



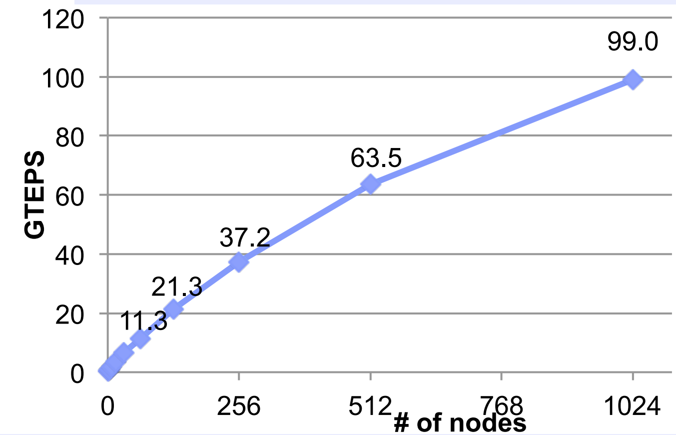
Vertex Sorting by utilizing the scale-free nature of the Kronecker Graph



2D Partitioning Optimization



Scalable 2D partitioning based CPU Implementation with Scale 26 per 1 node



Performance Comparison with CPU and GPU Implementations

TSUBAME 2.5 Supercomputer in Tokyo

TOP500 SUPERCOMPUTER SITES

TOP 10 Systems - 06/2011

1	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect
2	Tianhe-1A - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C
3	Jaguar - Cray XT5-HE Opteron 6-core 2.6 GHz
4	Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU
5	TSUBAME 2.0 - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows

GRAPH 500

THE GREEN 500

Green500 ... 2011

Complete Results - November 2011

Rank	Machine	Owner	Problem Size	TEPS
1	NNSA/SC Blue Gene/Q Prototype II (4096 nodes / 65,536 cores)	NNSA and IBM Research, T.J. Watson	32	254,349,000,000
2	Lomonosov (4096 nodes / 32,768 cores)	Moscow State University	37	103,251,000,000
3	TSUBAME (2732 processors / 1366 nodes / 16,392 CPU cores)	GSIC Center, Tokyo Institute of Technology	36	100,366,000,000
4	Jugene (65,536 nodes)	Forschungszentrum Jülich	37	92,876,900,000
5	Intrepid (32,768 nodes / 131,072 cores)	ANL	35	78,869,900,000

ch

ch

Technology

Our Scalable Algorithm continuously achieves 3rd or 4th place in the World since 2011/11

Outline

- Introduction
- Graph500 Benchmark
- ScaleGraph: Billion-Scale Graph Analytics Library
- Time-Series Analysis for Whole Twitter Network


Virtex-5

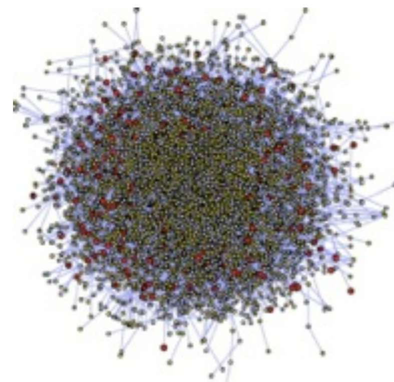

Intel Xeon Phi


Single-chip Cloud Computer


NVIDIA Tesla

Building Large-Scale Graph Analytics Library

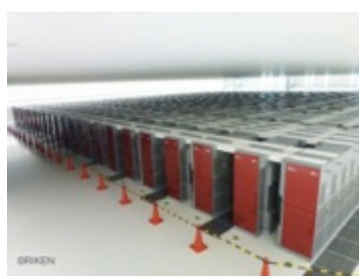
- Programming models that offer performance and programmer productivity are very important for conducting big data analytics in Exascale Systems.
- HPCS languages are an example for such initiatives.
- It is very important for having complex network analysis software APIs in such languages


Human Protein Interaction Network (P.M. Kim et al, 2007).

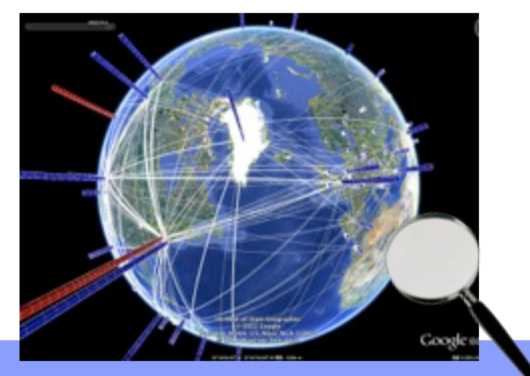
BigData

Crawled the entire Twitter follower/followee network of **826.10 million vertices and 28.84 billion edges. How could we analyze this gigantic graph ?**


Tsubame 2.0


K computer

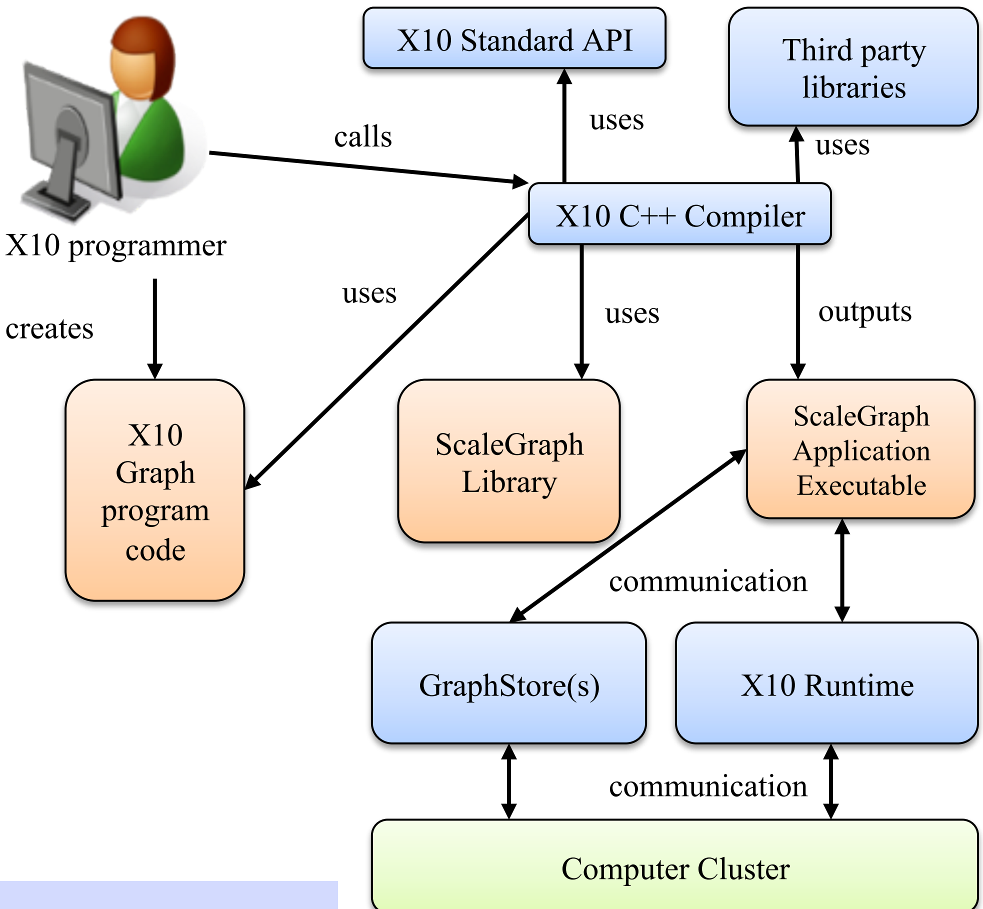

Titan



Supercomputers

ScaleGraph : Large-Scale Graph Analytics Library

- **Aim** - Create an open source **X10-based Large Scale Graph Analytics Library** (beyond the scale of billions of vertices and edges).
- **Objectives**
  - To define concrete abstractions for Massive Graph Processing
  - To investigate use of X10 (I.e., PGAS languages) for massive graph processing
  - To support significant amount of graph algorithms (E.g., structural properties, clustering, community detection, etc.)
  - To create well defined interfaces to Graph Stores
  - To evaluate performance of each measurement algorithms and applicability of ScaleGraph using real/synthetic graphs in HPC environments.

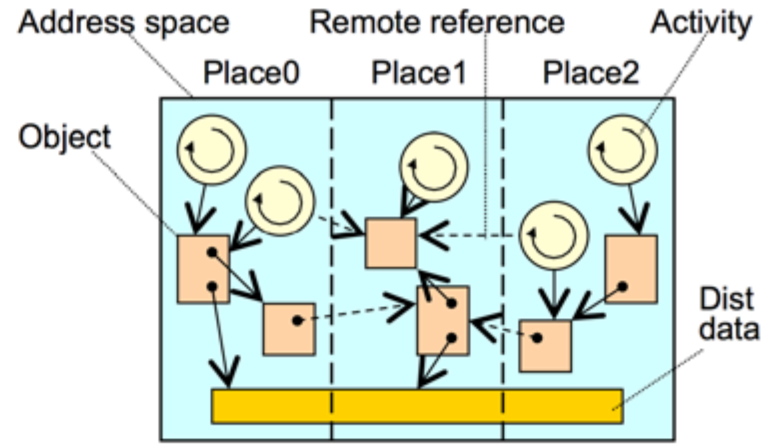


URL: http://www.scalegraph.org/

Programming Language X10

X10 is a new parallel distributed programming language being developed by IBM Research.

- **X10 aims at improving the productivity of highly parallel and distributed applications.**
  - Enables scalable programming for parallel distributed environment, where many multicore SMP chips and GPGPUs are interconnected.
- **X10 adopts APGAS** (Asynchronous Partitioned Global Address Space) **programming model.**
  - Can manage multiple machines as a global memory space partitioned into “Places”.
  - Can create lightweight asynchronous “Activities”.
  - Supports creation and reference of activities and objects in remote places.
- **X10 supports various execution environments.**
  - Can run both on Java execution environments and native environments.
  - Provides development tools integrated into Eclipse.
- **X10 is being developed as an open source project.**
  - See http://x10-lang.org/ for more information.



APGAS programming model

```
public class MyDistCalc {
  public static def main(Array[String]) {
    val R = 1..1000; val D = Dist.makeBlock(R);
    val arr = DistArray.make[Int](D, ([i]:Point)=>i);

    val places = arr.dist.places();
    val tmp = new Array[Int](places.size);
    finish for ([i] in 0..places.size-1) async {
      tmp(i) = at (places(i)) {
               val a = arr | here;
               var s:Int = 0; for (pt in a) s += a(pt)*a(pt);
               s  // return value of at
           };
    }
    var result:Int = 0; for (pt in tmp) result += tmp(pt);
    Console.OUT.println(result);        // -> 333833500

    // We can actually use DistArray.map and reduce
    val r = arr.map((i:Int)=>i*i).reduce(Int.+, 0);
    Console.OUT.println(r);             // -> 333833500
} }
```

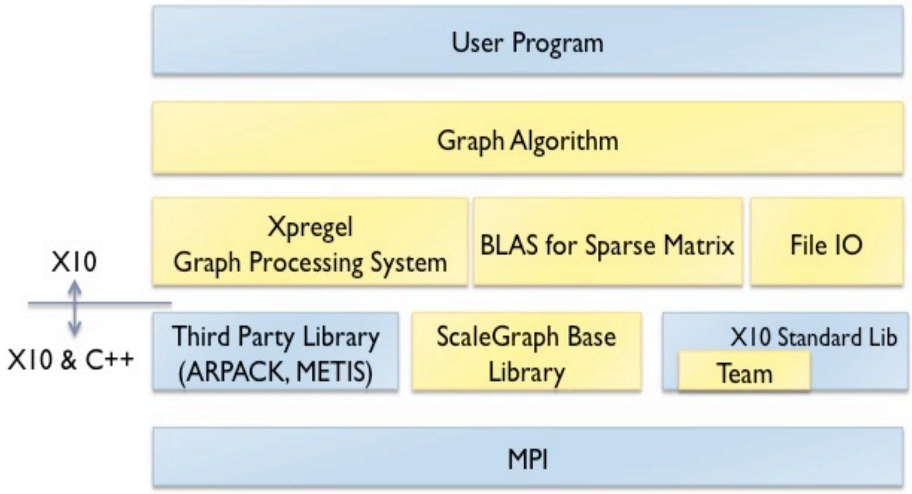
Distributed programming by X10

Features of ScaleGraph

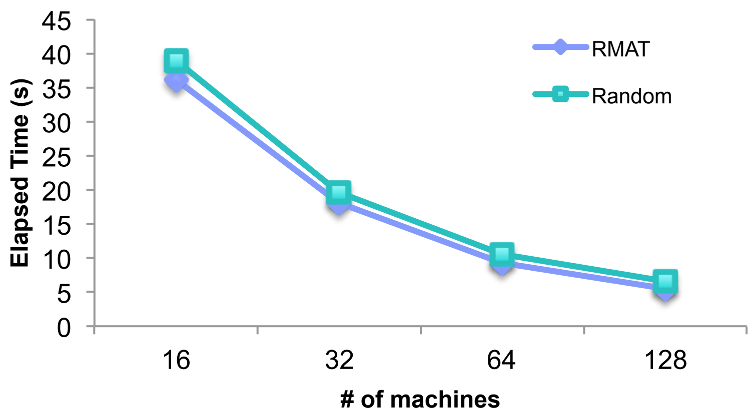
- XPregel frame work which is based on Pregel computation model[1] proposed by Google
- Optimized collective routines (e.g., alltoall, allgather, scatter and barrier)
- Highly optimized array data structure (i.e., MemoryChunk) for very large chunk of memory allocation
- Rich graph algorithms (e.g., PageRank, spectral clustering, degree distribution, betweenness centrality, HyperANF, strongly-connected component, maximum flow, SSSP, BFS)
- We achieved running PageRank, spectral clustering, degree distribution on huge Twitter graph with 469M of users and 28.5B of relationships

[1] Malewicz, Grzegorz, et al. "Pregel: a system for large-scale graph processing." Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010.

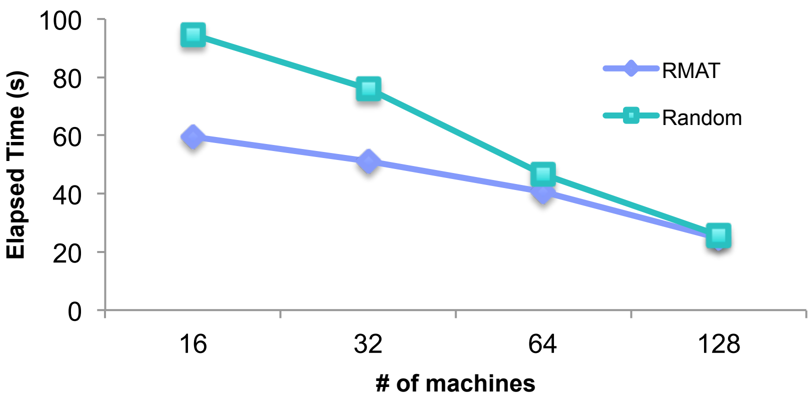
ScaleGraph Software Stack and Evaluation



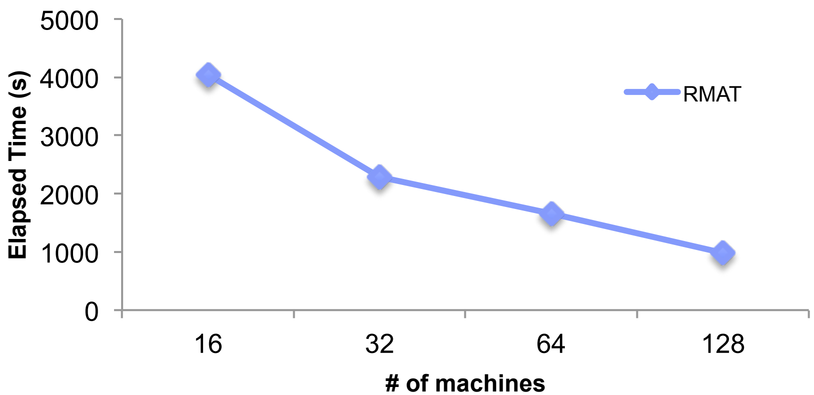
Strong-scaling result of degree distribution (scale 28)



Strong-scaling result of HyperANF (scale 28)



Strong-scaling result of spectral clustering (scale 28)



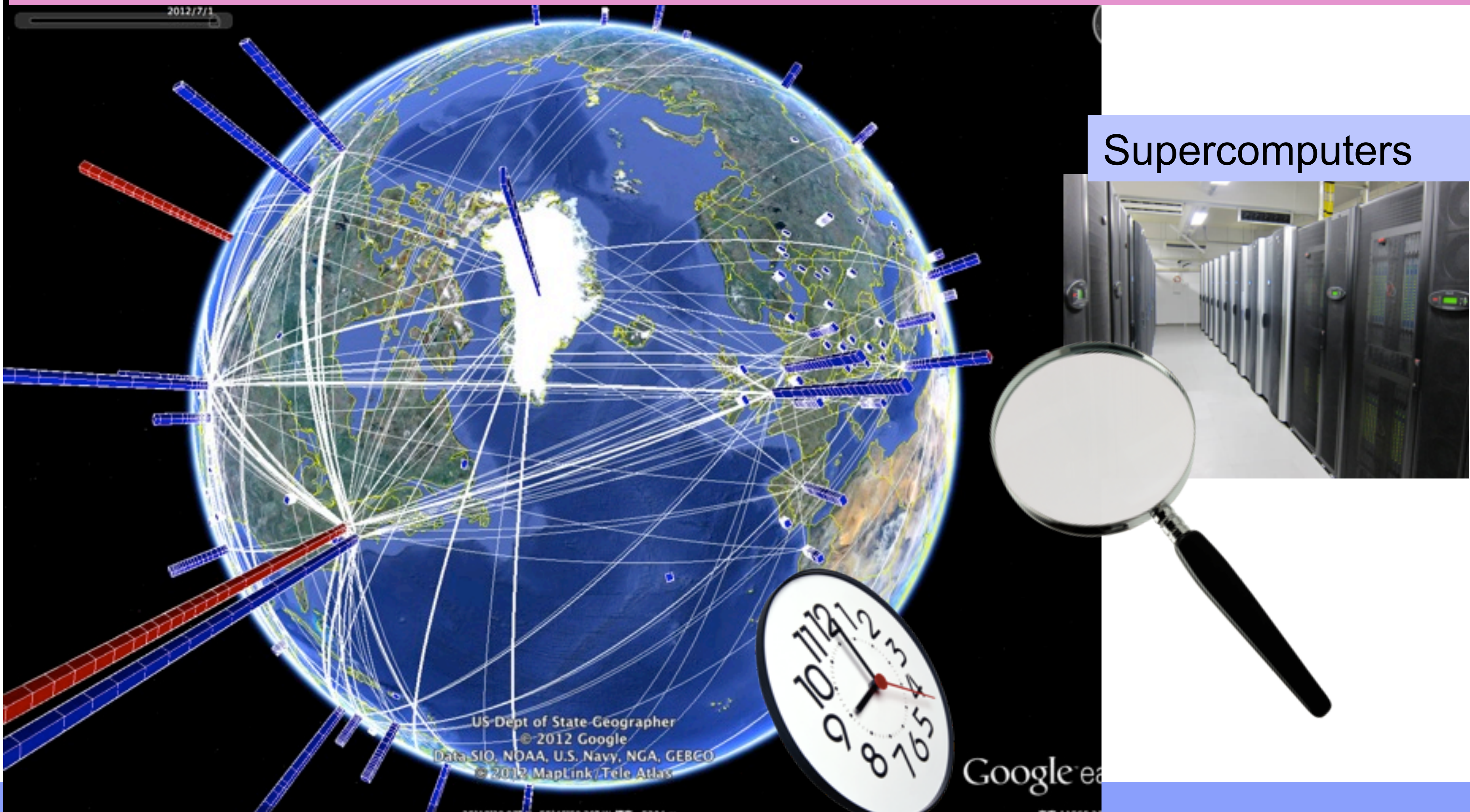
The scale-28 graphs we used have 2^{28} (≈268M) of vertices and 16×2^{28} (≈4.29B) of edges

Outline

- Introduction
- Graph500 Benchmark
- ScaleGraph: Billion-Scale Graph Analytics Library
- Time-Series Analysis for Whole Twitter Network

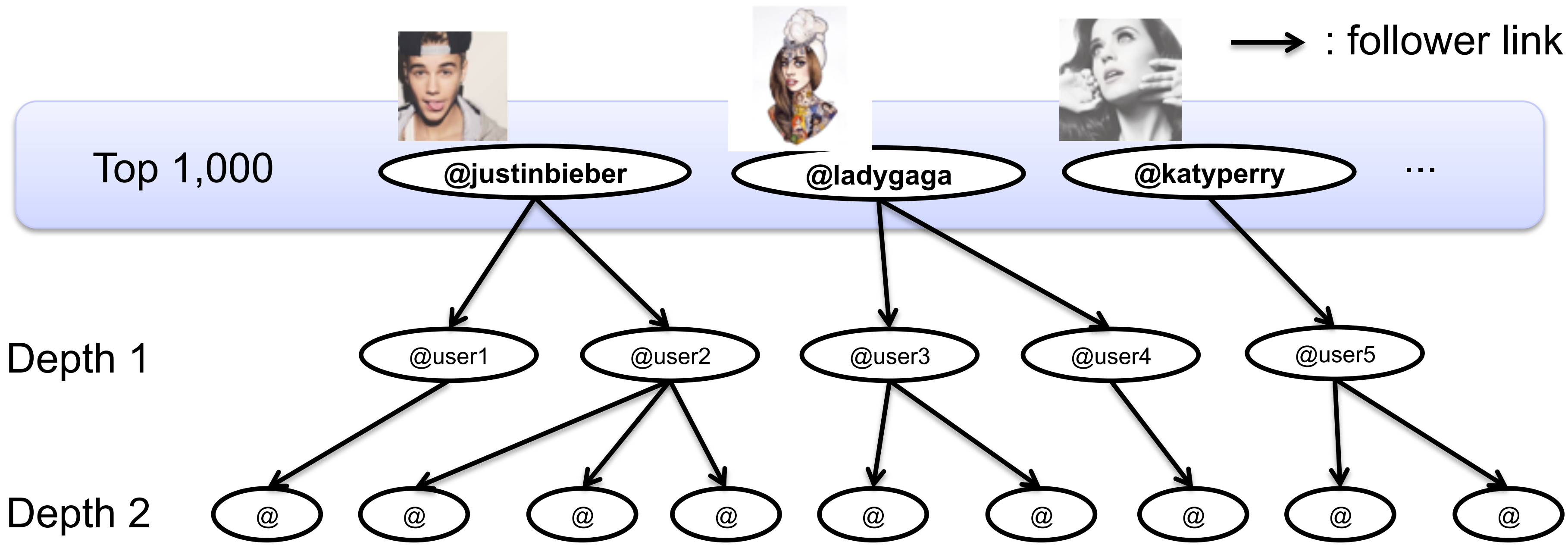
Understanding time-series nature of large-scale social networks (e.g. separation of degree, diameter, clustering, ..)

Crawled the entire Twitter follower/followee network of **826.10 million vertices and 29.23 billion edges. How could we analyze this gigantic graph ?**



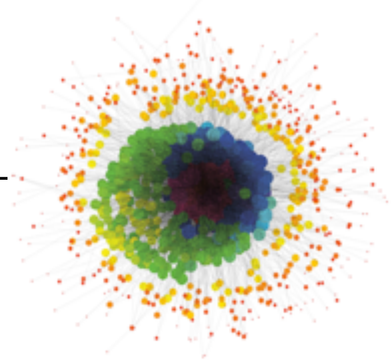
Crawling Billion-Scale Twitter Follower-Followee Network

- with Twitter API (v1.0) from Jul. 2012 to Oct. 2012 (around 3 months).
- begin with top 1,000 users[*1] with the largest number of followers
- according to breadth-first search along the direction of follower



*1 : Twitaholic. http://twitaholic.com/top100/followers/

Crawled Data Set – Big Data !!



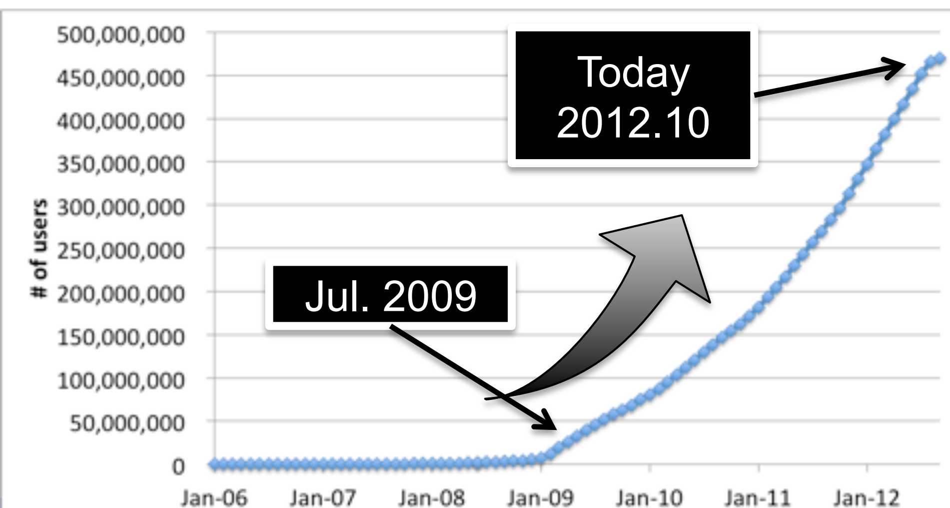
- We stopped our crawling at depth 29
  - Because the user after depth 26 was less than 100.
  - Finally, we collected **469.9 million user data**.
- Collect two kind of user data by crawling for 3 months
  - 1. User profile
    - Include user id, screen_name, description, account creation time, time zone, etc.
    - The serialized data size is **91GB**
  - 2. Follower-friend
    - Adjacency list of followers and friends
    - The compressed(gzip) data size is **231GB**
- To perform the Twitter network analysis
  - **Apache Hadoop** for large-scale data processing
  - **HyperANF** for approximate calculation of degree of separation and diameter
    - Lars Backstrom[*1] also use HyperANF for Facebook network analysis

*1 : "Four degrees of separation" *ACM Web Science 2012*

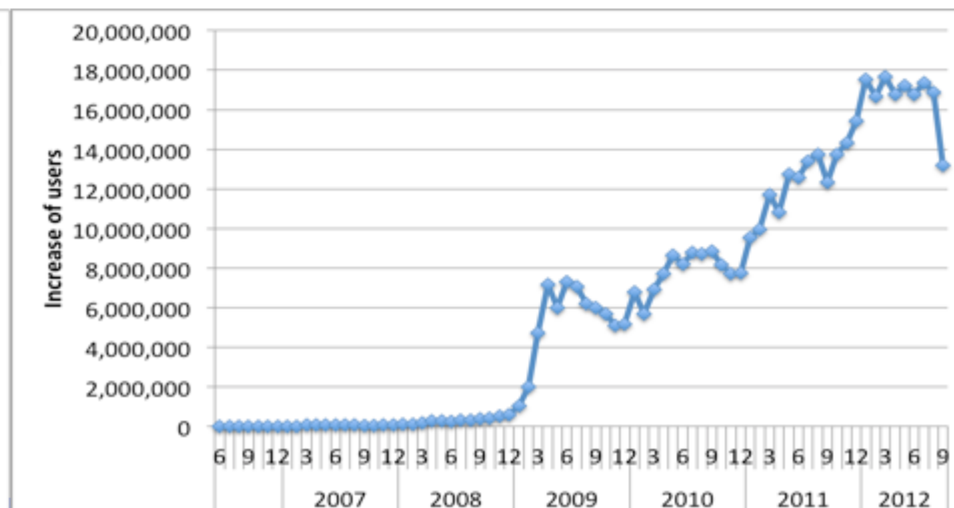
Explore Twitter Evolution (1/2)
- Transition of the number of users

- Total user count (left fig.)
  - Twitter started at June 2006 and rapidly expanded from beginning of 2009.
  - Haewoon Kwak *1 studied Twitter network on July 2009
- Monthly increase of users (right fig.)
  - Twitter users increase, but it seems a little unstable...

Total user count



Monthly increase of users

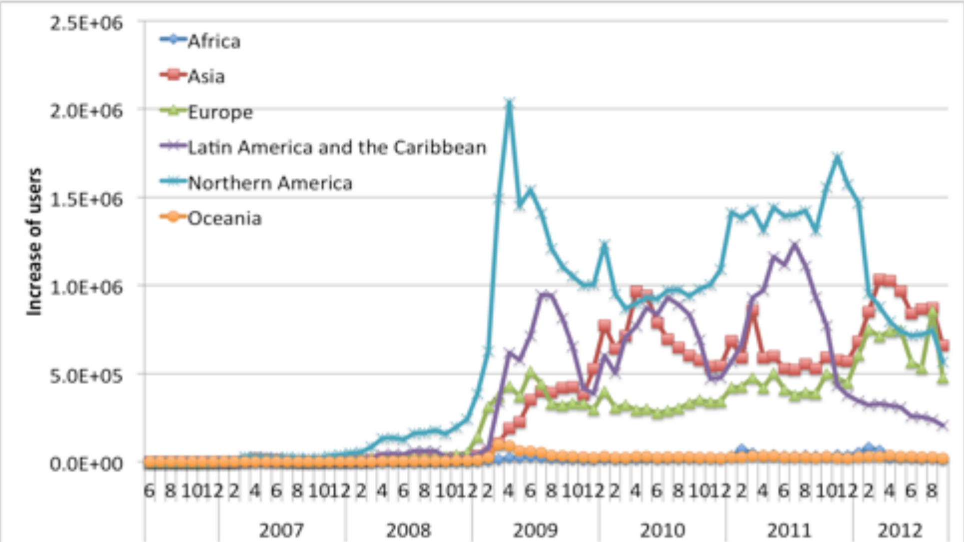


*1 : “What is Twitter, a social network or a news media?”

Explore Twitter Evolution (2/2)
- Transition of the number of users by regions-

- **Classify 131 million users by "Time zone" property under 6 regions**
  - Africa, Asia, Europe, Latin America and Caribbean (Latin), Northern America (NA), Oceania
  - Only 131 million user correctly set one's own "Time zone"
- **Massive change of ratio of users by region**
  - Asia users : 8.30% => 20.8% (12.5% up)
  - NA users : 54.4% => 40.4% (14.0% down)

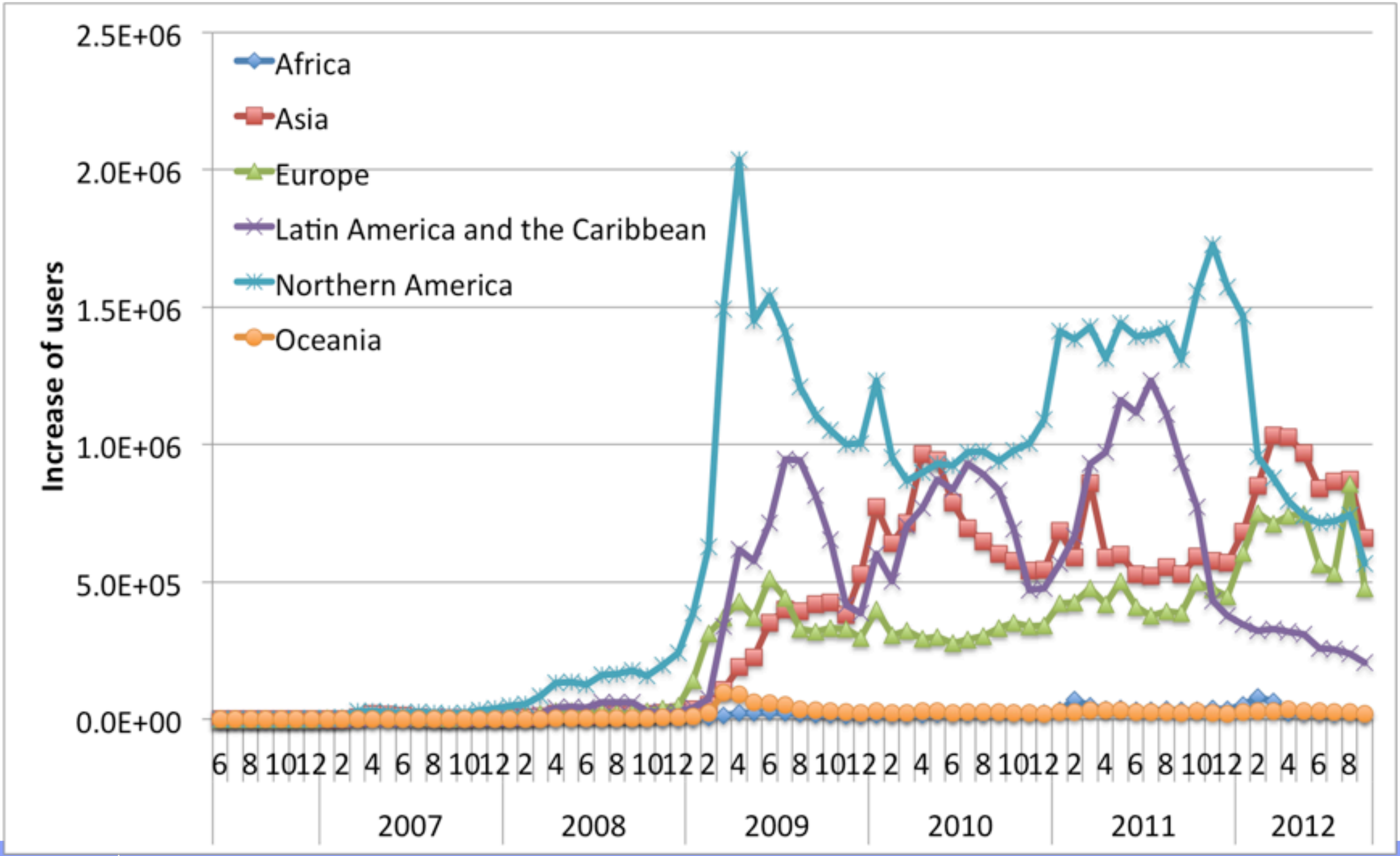
Characteristic of Twitter network also change?



Monthly increase of users by region

	July 2009		October 2012	
	# users	ratio (%)	# users	ratio (%)
Africa	0.13M	0.66	1.27M	0.96
Asia	1.65M	**8.30**	27.4M	20.8
Europe	3.01M	15.1	19.8M	15.1
Latin	3.80M	19.0	28.5M	21.6
NA	10.9M	**54.6**	53.1M	40.4
Oceania	0.45M	2.29	1.52M	1.15
Total	19.9M	100	131M	100

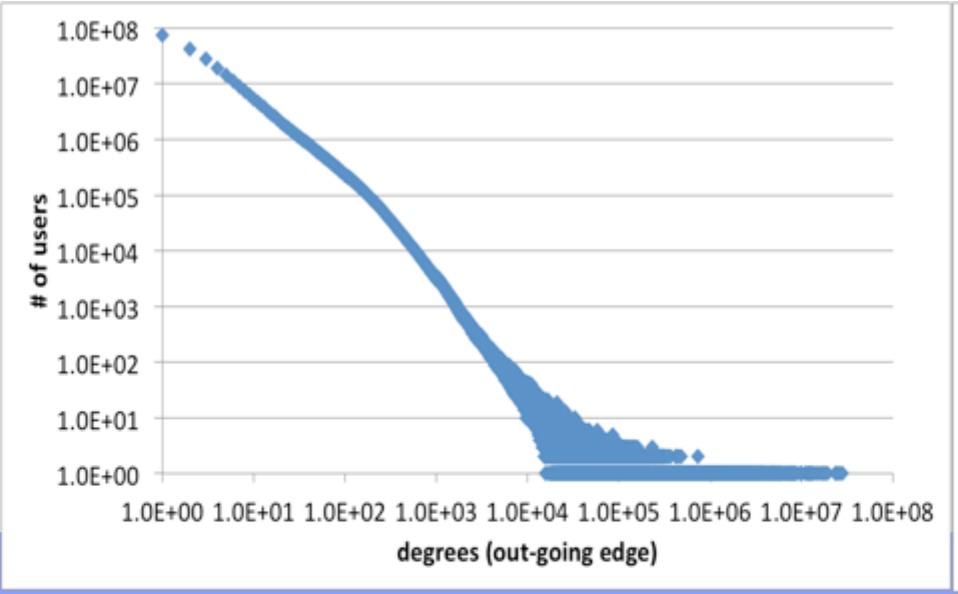
Monthly Increase of Users by Regions



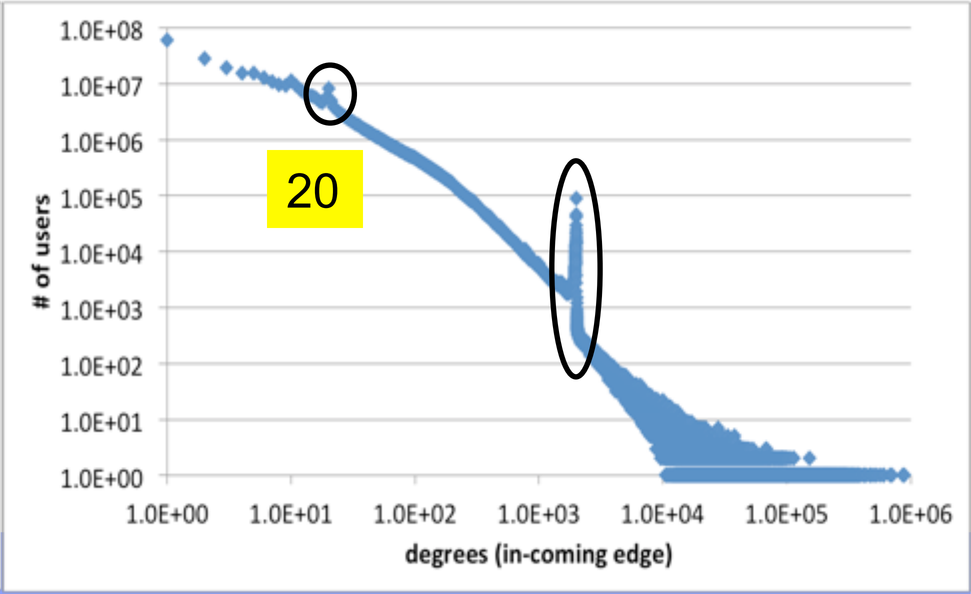
Degree Distribution: Unexpected value in in-degree distribution

- “Scale-free” is one of the features of a social graph
- Unexpected value in in-degree distribution
  - at x=20 due to Twitter recommendation system
  - at x=2000 due to upper bound of friends before 2009

Out-degree distribution (follower)

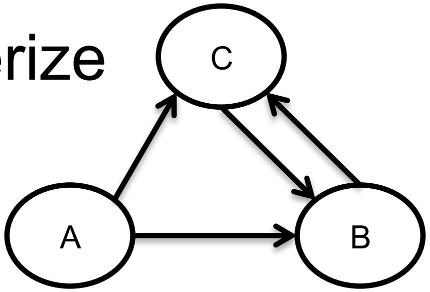


In-degree distribution (friend)



Reciprocity : **decline from 22.1% to 19.5%**

- Reciprocity is a quantity to specifically characterize directed networks. Traditional Definition:



$$r = \frac{L^{\leftrightarrow}}{L}$$

$L^{\leftrightarrow}$: # of edges pointing in both directions

L : # of total edges

$L^{\leftrightarrow} = 1$

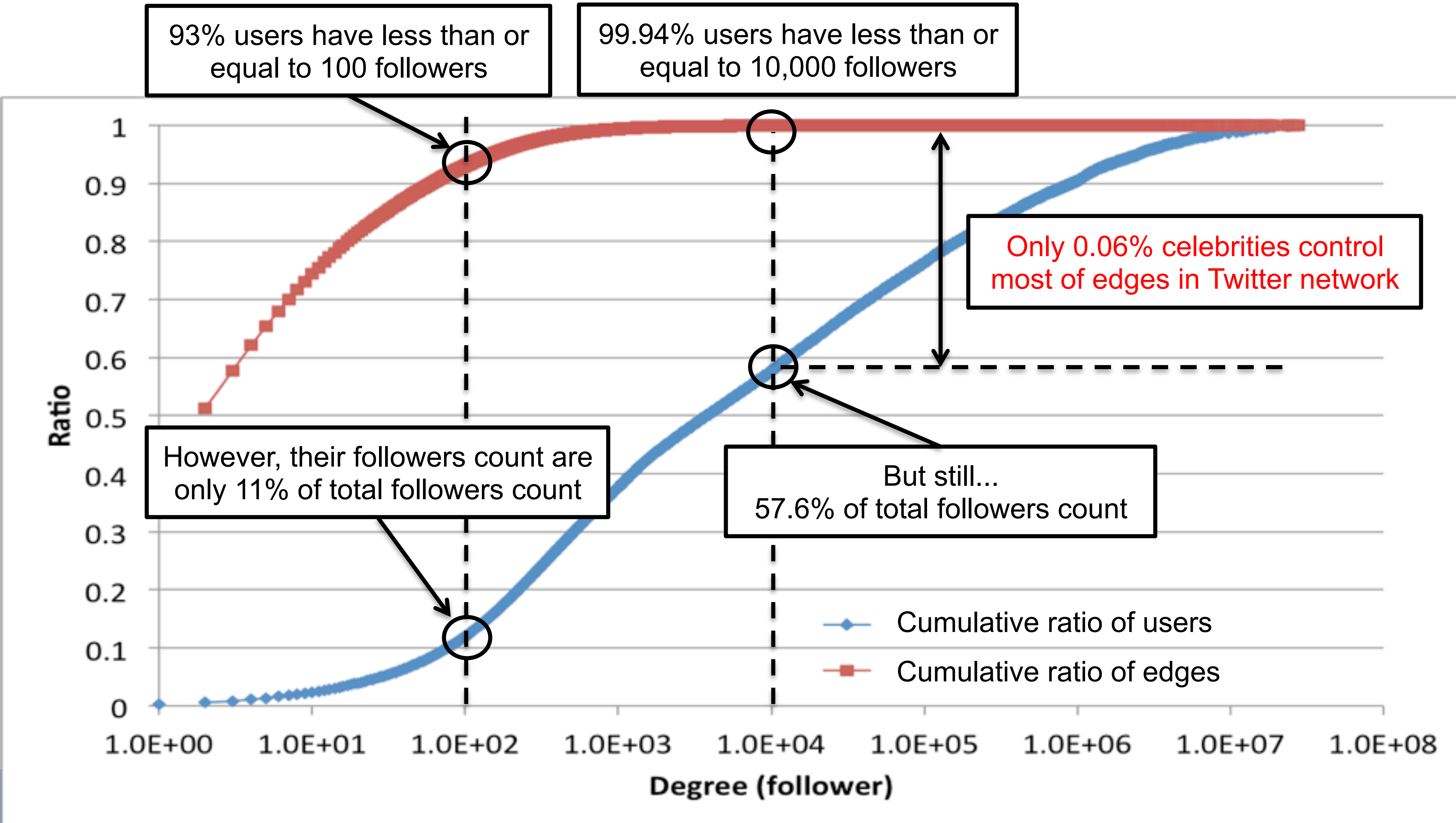
$L = 3$

- As a result, **Twitter network reciprocity decline from 22.1% to 19.5%**

	July 2009	October 2012
# of users	41.6 M	465.7 M
# of edges	1.47 B	28.7 B
Reciprocity	22.1% *1	19.5%

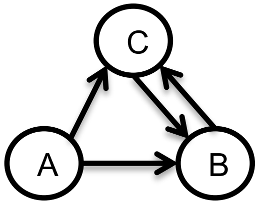
*1 : "What is Twitter, a social network or a news media?"

How many edges do celebrities have in Twitter network ? → Only 0.06% celebrities control most of edges



Degree of Separation and Network Diameter (1/3)

- Both degree of separation and diameter are measures to characterize networks in terms of scale of graph.
- Definition
  - **Degree of Separation**
    - **Average** value of the shortest-path length of all pairs of users.
  - **Diameter**
    - **Maximum** value of the shortest-path length of all pairs of users
  - Note : unreachable pairs are excluded from calculation



(A, B) = 1
(A, C) = 1
(B, A) = ∞
(B, C) = 1
(C, A) = ∞
(C, B) = 1

Degree of Separtion : 1
Diameter : 1

Degree of Separation and Network Diameter (2/3)

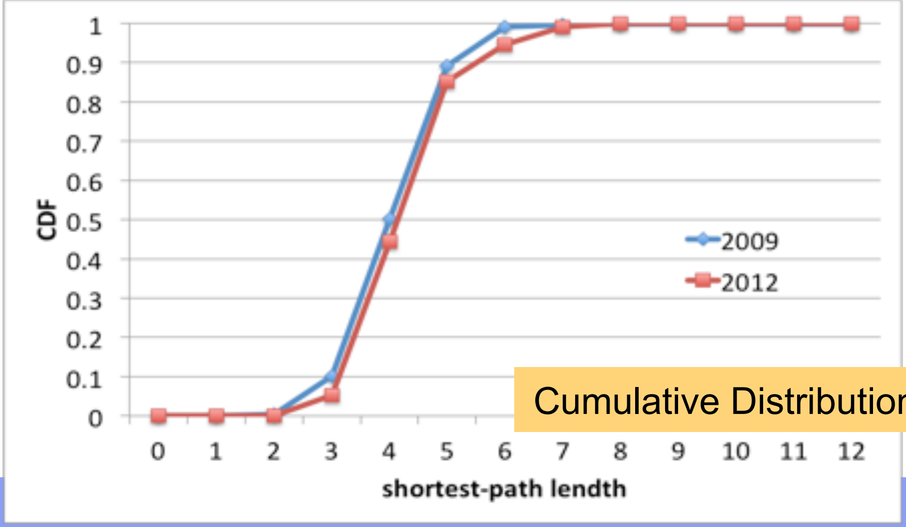
- Experimental environment
  - Using **HyperANF [Paolo, WWW'12]** on TSUBAME 2.0 (Supercomputer at TITECH)
    - TSUBAME 2.0 Fat node
      - **64 cores, 512 GB memory**, SUSE Linux Enterprise Server 11 SP1
    - HyperANF Parameters
      - We set the logarithm of the number of registers per counter to 6 in order to reduce an error.

  - Four times executions
    - Degree of Separation
      - take a average of 4 calculation
    - Diameter
      - take a minimum value of 4 calculation
      - because HyperANF guarantee lower bound of diameter
    - **Each execution on 2012 took more than 42,000 sec.**

Degree of Separation and Network Diameter (3/3)

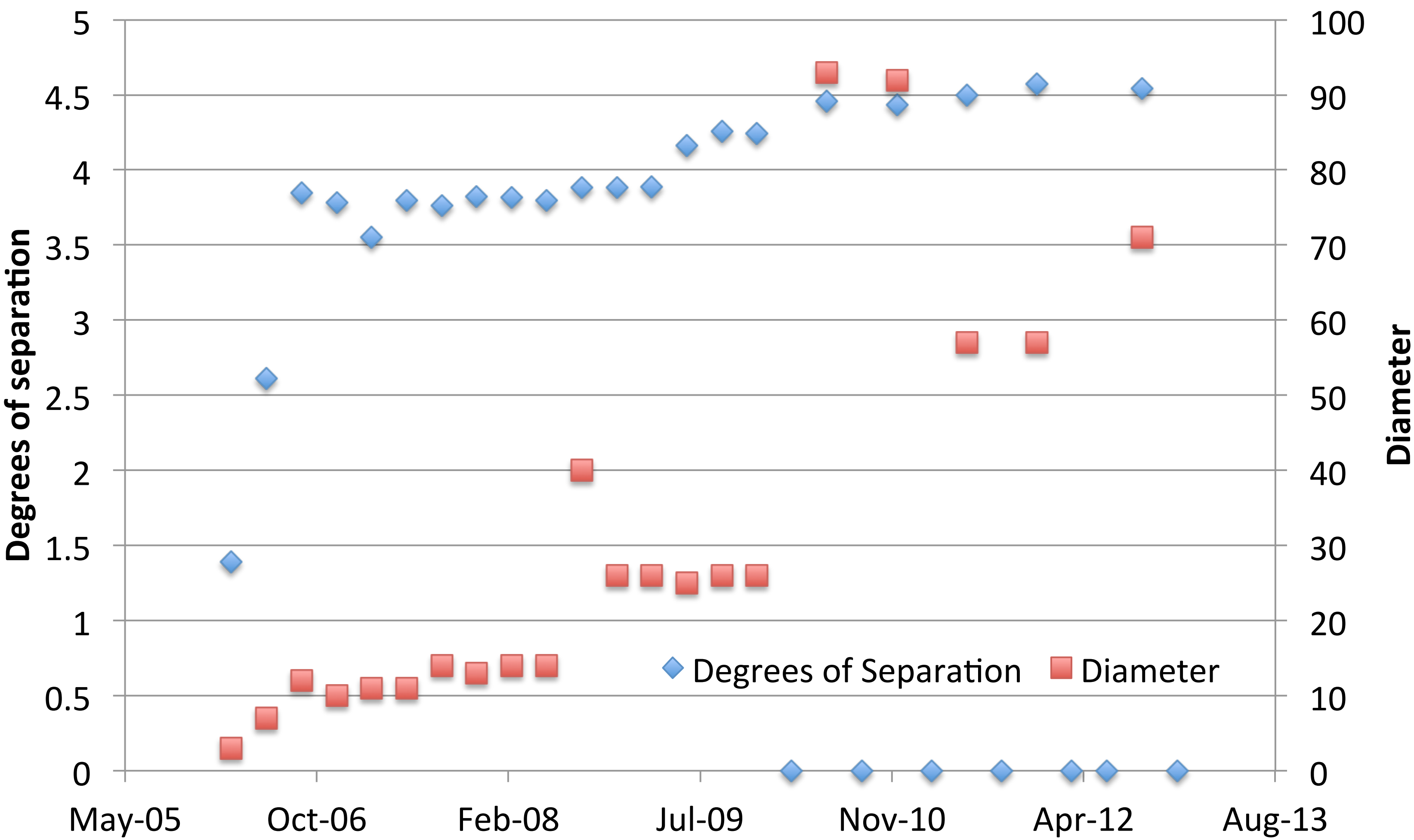
- Degree of Separation
  - Only a little difference between '09 and '12 in spite of the lapse of three years.
- Diameter
  - Diameter of 2012 is much larger than the one of 2009.
- Cumulative Distribution
  - In 2009
    - 89.2% of node pairs whose path length is 5 or shorter
    - 99.1% pairs whose it is 6 or shorter.
  - In 2012
    - 85.2% pairs whose it is 5 or shorter
    - 94.6% pairs whose it is 6 or shorter

	Degree of Separation		Diameter	
	2009	2012	2009	2012
1st	4.39	4.48	25	70
2nd	4.46	4.65	26	71
3rd	4.53	4.54	25	70
4th	4.62	4.71	25	71
Result	**4.50**	**4.59**	**26**	**71**



Cumulative Distribution

Degree of Separation and Diameter for Time-Evolving Twitter Network



Classifying Degree of Separation by Spoken Language

	Spanish	Portuguese	Japanese	Turkish	French
# of Users	64,927,267	22,456,938	20,279,402	10,402,846	10,743,511
Follow ratio to its own language	64%	58%	89%	57%	51%
Follow ratio to English	31%	36%	9%	39%	44%
# of Nodes for DOS	60,708,434	21,152,308	19,682,116	9,638,906	8,964,888
# of Edges for DOE	2,266,838,184	1,098,723,999	1,394,986,423	271,513,323	177,419,512
Average Degree	37.33	51.94	70.87	28.16	19.79
Degree of Separation (Average path length between two users)	4.625	4.253	4.014	4.340	4.699
Diameter（Lower bound value）	42	23	27	39	22

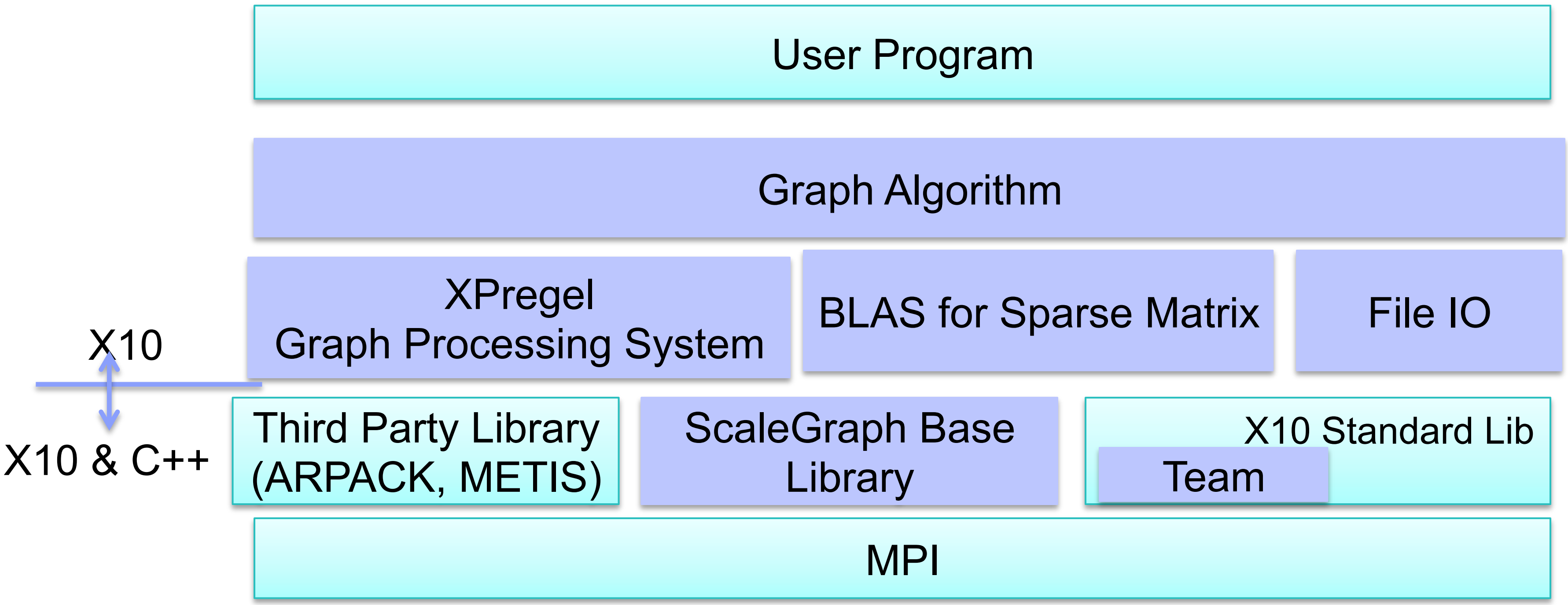
Summary and Call for Collaboration

- **Official web site** – http://www.scalegraph.org/
  - Project information
  - Source code distribution / VM Image
  - Documentation
  - Source Code Repository : http://github.com/scalegraph/
- **Call for Collaboration**
  - Sharing our whole Twitter network and all the user profile as of 2012/10
  - More application-driven research and development in the ScaleGraph project

Supplemental materials

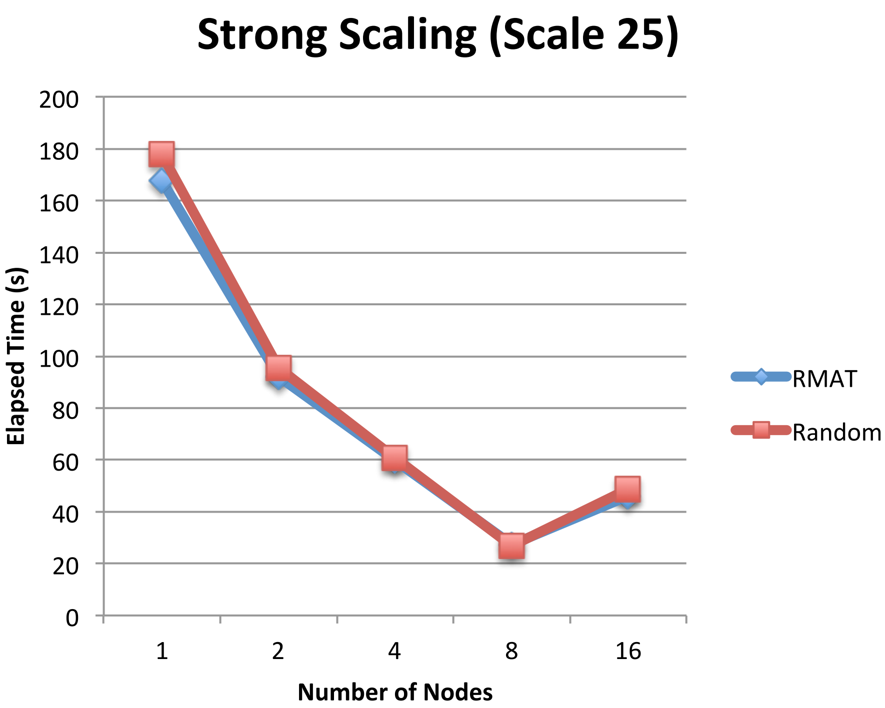

US Dept of State Geographer
© 2012 Google
© 2012 MapLink/Tele Atlas
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
©2010

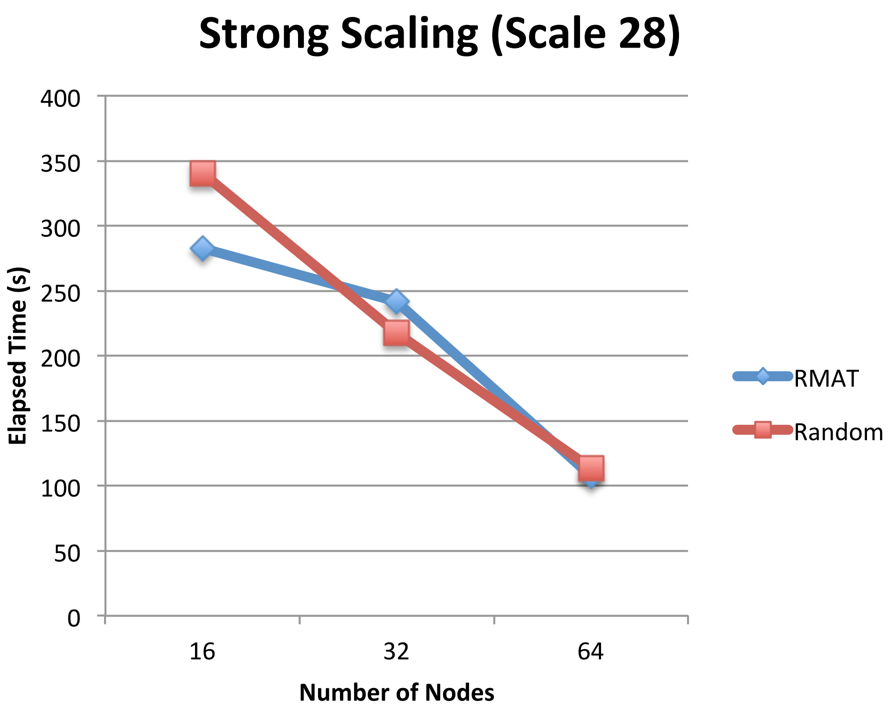
ScaleGraph Software Stack



Degree of Separation

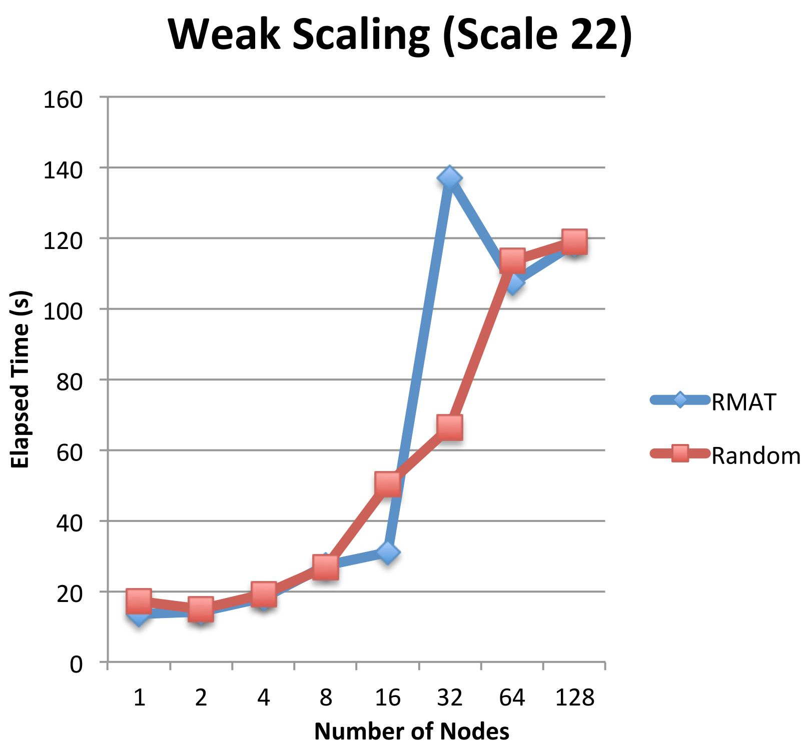
HyperANF – Strong Scaling Performance Analysis



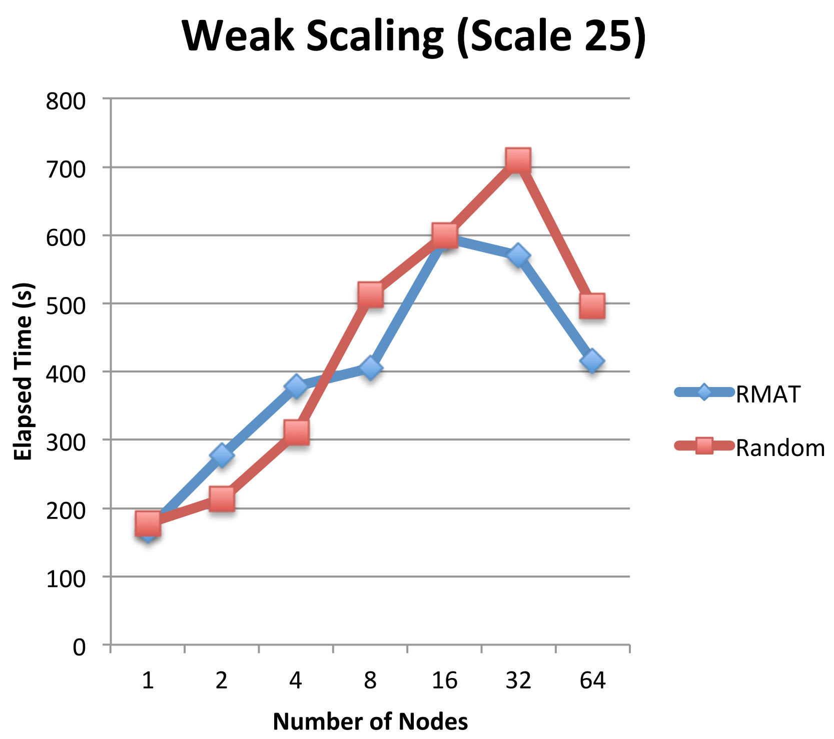


(Scale = 25, B=7, R-MAT Graph, 33.33 Million Vertices and 536 Million Edges
Scale = 28, R-MAT Graph, 268.43 Million Vertices and 4.295 Billion Edges)

HyperANF – Weak Scaling Performance Analysis

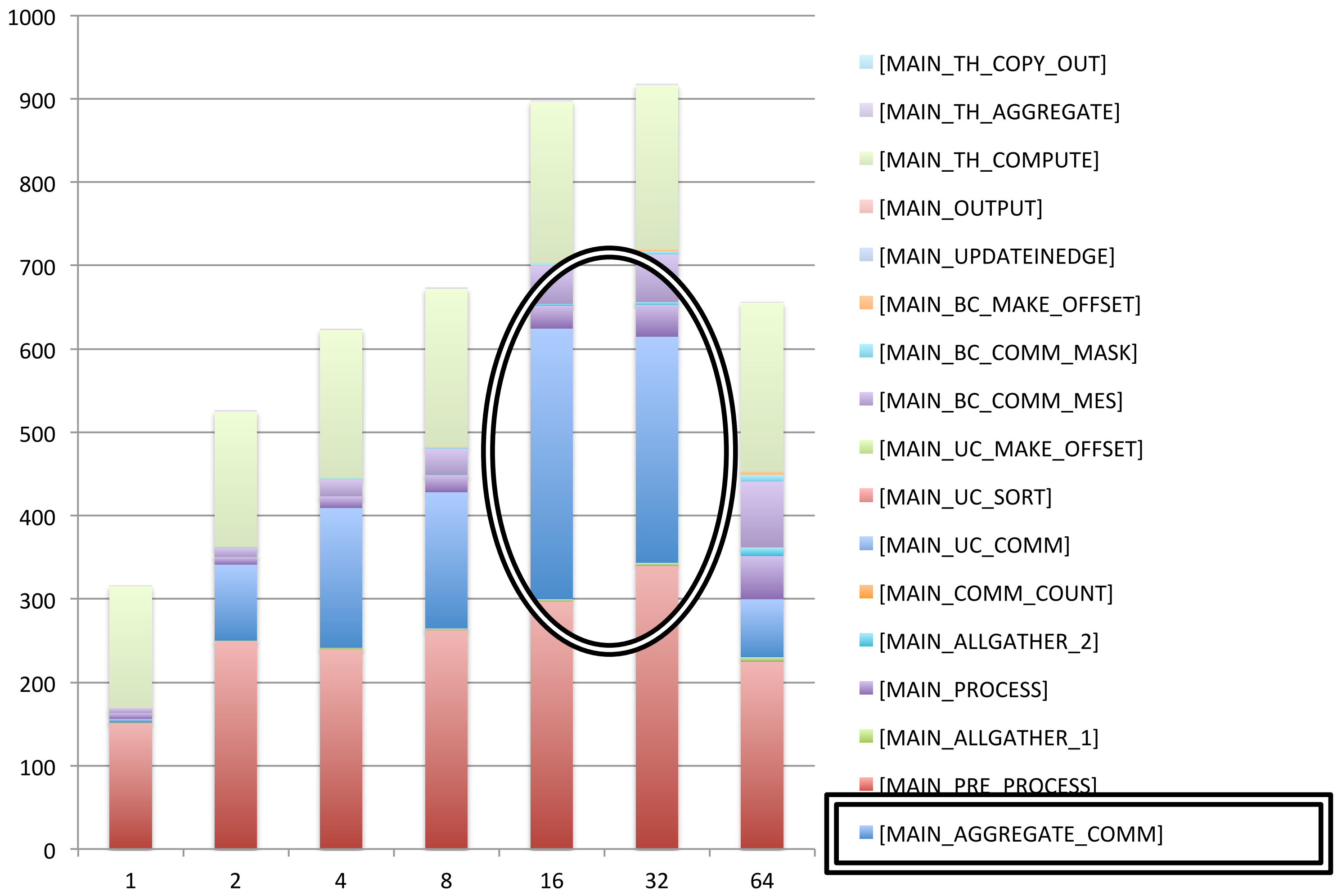


Scale 29 for 128 nodes



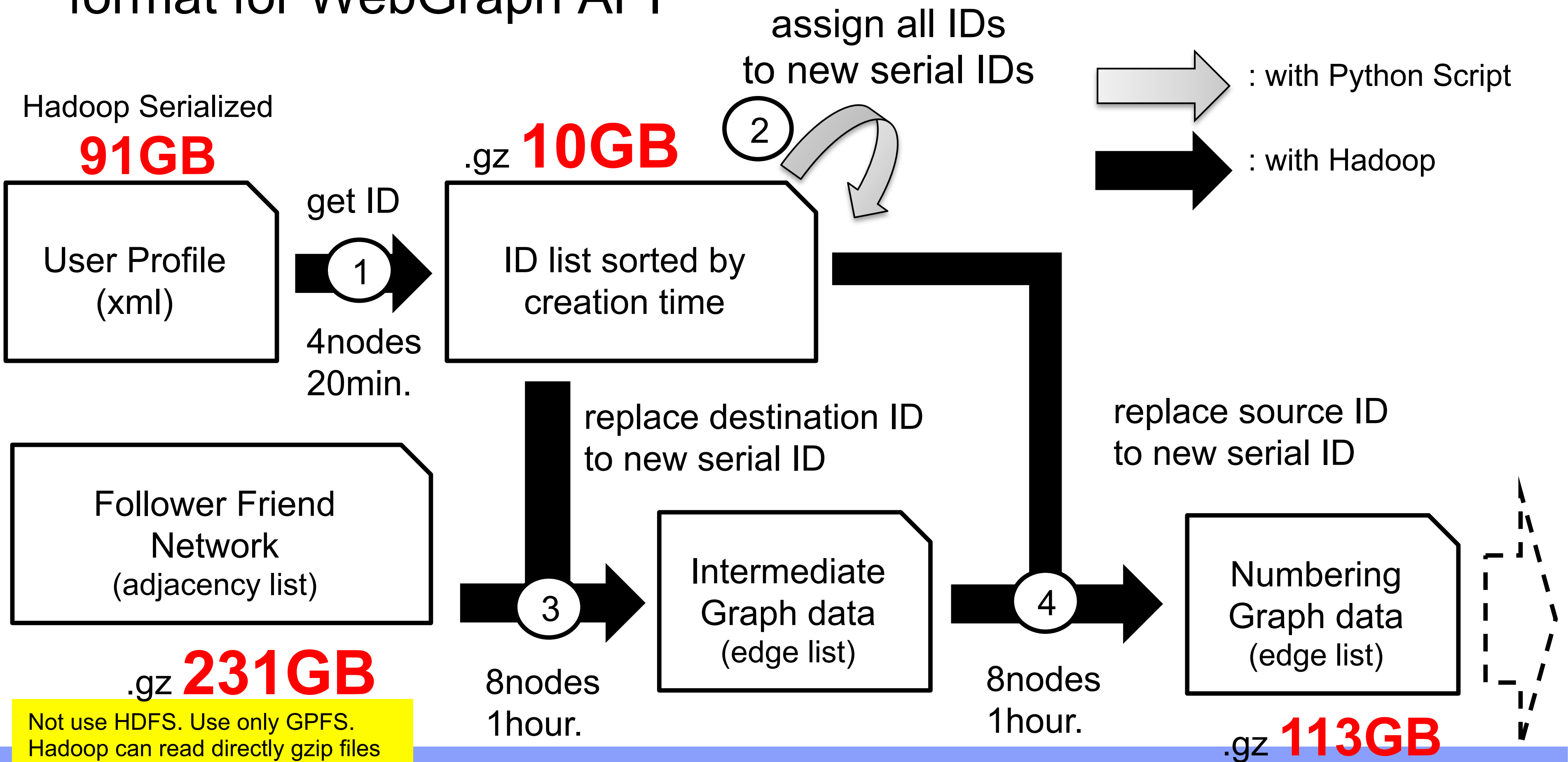
Scale 31 for 64 nodes

Weak Scaling – Profiling (Scale 25 per node, RMAT)



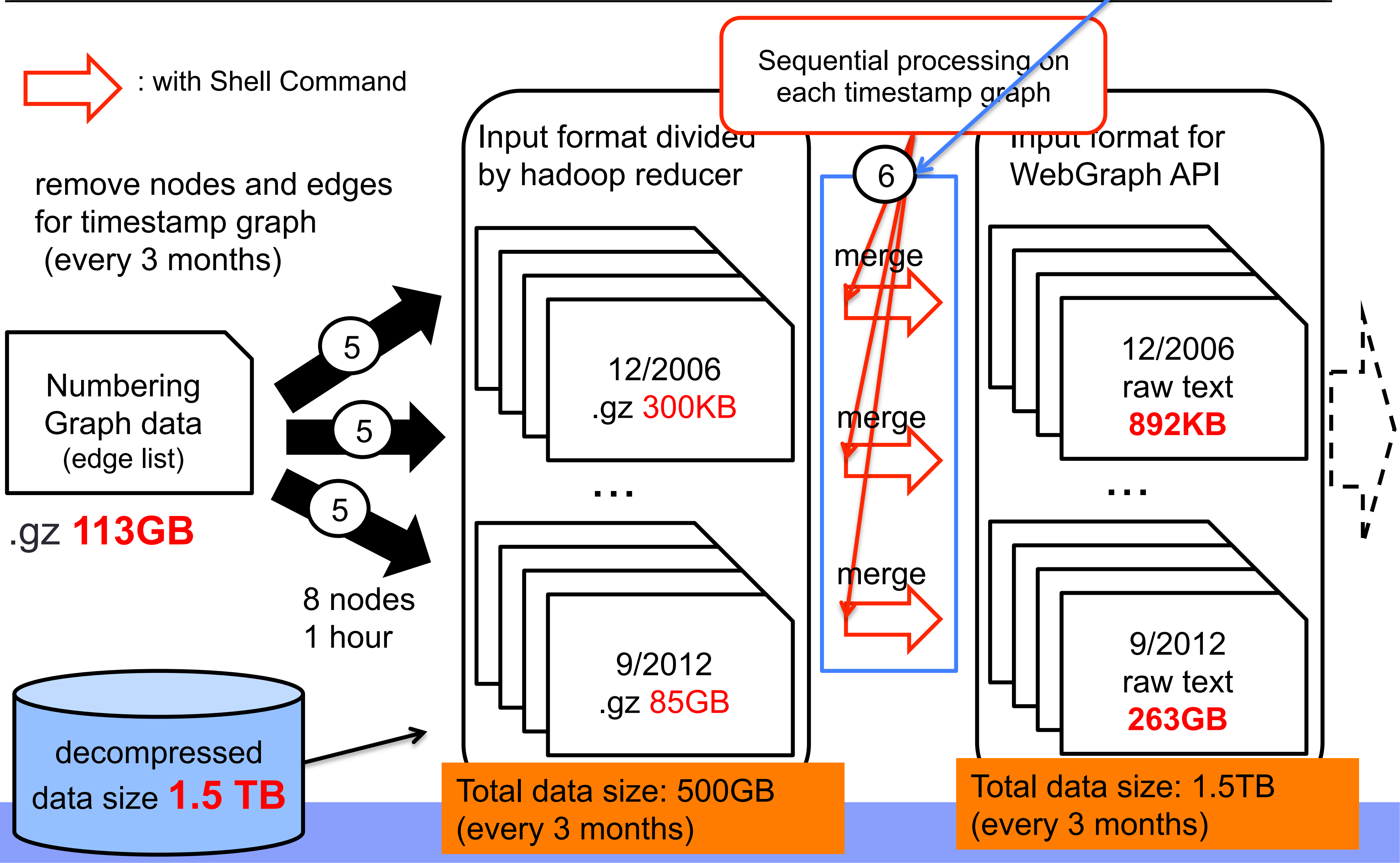
Workflow for Temporal Analysis (1/3)

- Convert Twitter user profile and network files to input format for WebGraph API

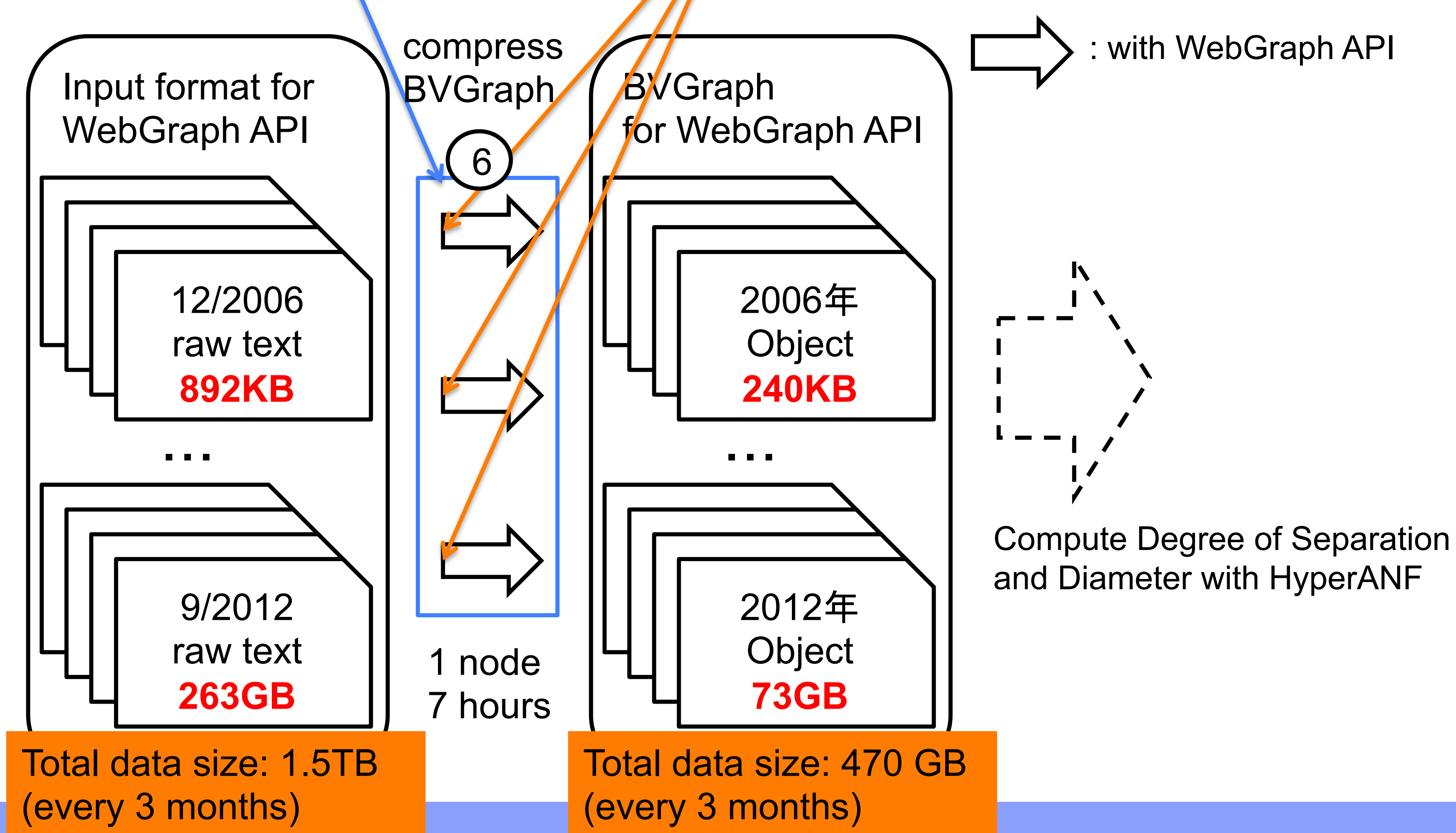


Not use HDFS. Use only GPFS.
Hadoop can read directly gzip files

Workflow for Temporal Analysis (2/3)

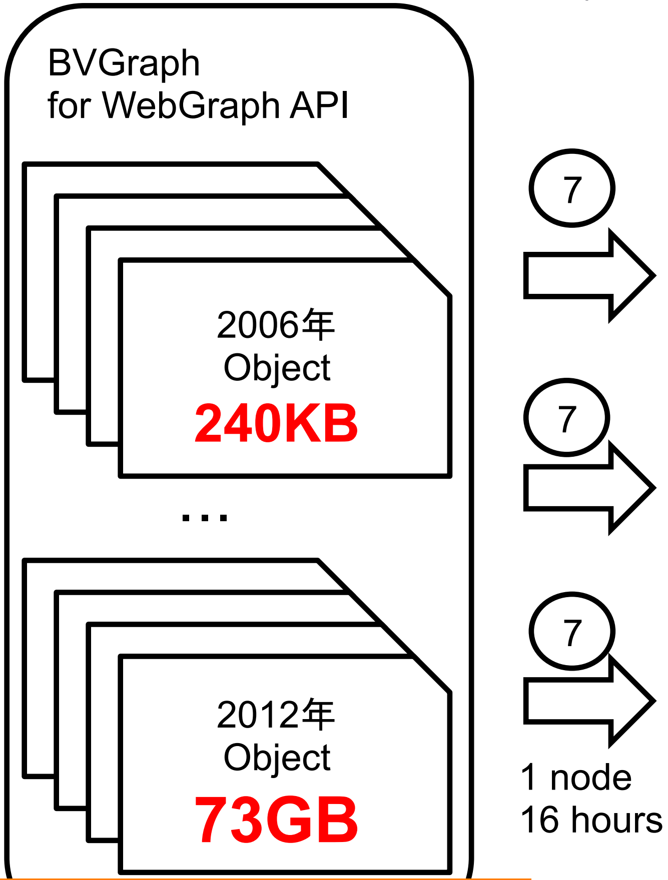


Workflow for Temporal Analysis (3/3)



Workflow : Degree of Separation

- Use HyperANF in WebGraph on TSUBAME 2.0 Fat Node
  - take 16 hours with 1node (64cores, 512 GB RAM)



Total data size: 470 GB (every 3 months)