ISSN 2186-7437

NII Shonan Meeting Report

No. 2016-1

Analytics on Complex Networks: Scalable Solutions for Empirical Questions

Organizers:
George Fletcher
Taro Takaguchi
Yuichi Yoshida

February 8–11, 2016

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Analytics on Complex Networks: Scalable Solutions for Empirical Questions

Organizers:
George Fletcher (Eindhoven University of Technology)
Taro Takaguchi (National Institute of Informatics)
Yuichi Yoshida (National Institute of Informatics)

February 7–11, 2016

1 General summary

1.1 Motivations of the meeting

Real-world networks, such as web graphs, social networks, and biological networks, have remarkable topological features in common: the scale-free property, that is, the degree distribution obeys a power-law function, and the small-world property, that is, the average distance between two vertices is small. These networks are collectively referred to as complex networks and have been intensively studied in the last decade.

In their empirical study of complex networks, researchers across the fields of data engineering, theoretical computer science, and network science are developing solutions for analytics. Due to rapid growth of the sizes of real-world networks, however, we are facing challenges in scaling these tools, as we will describe below. We desire to solve these issues by bridging the knowledge in the three fields with the single key notion: complex networks.

Researchers in the engineering of data-intensive systems, such as database and data mining systems, have traditionally focused on the study of scalable solutions for analytics on big data collections. With the increasing availability and broad interest in massive complex networks, researchers in data engineering have hence focused considerable effort in recent years on scalable mining and querying over such data collections. However, very little work has been done on coordinating the results of these investigations with the foundational work in this area in the theoretical computer science and network science communities. Stronger ties between these communities would help deepen the understanding and development of theoretically grounded analytics solutions for the empirical investigations of interest in the broader scientific community.

From the perspective of theoretical computer science, the important point is that we need algorithms for real-world networks but not for all the possible networks. The majority of theoretical studies on graph problems pay attention to worst-case time complexity or average-case time complexity, which may have little to do with real-world networks. On the other hand, practical algorithms on graph problems proposed in the fields of databases, data mining, and machine

learning have shown their efficacy on real-world networks, but these algorithms lack theoretical groundings to understand why they work well in practice. By applying theoretical analysis frameworks to these algorithms, we want to answer foundational questions such as "what is the right time complexity of this particular problem when the given graph is a complex network?" and "why does this particular heuristic make the algorithm faster or more accurate when the given graph is a complex network?"

Network science, a new subfield of statistical physics seeking universality hidden behind real-world networks, can move on to a new stage with aids provided by data engineering and theoretical computer science. For example, a large number of models describing growth processes of complex networks and information spreading on them have been studied in network science. However, their computational properties are largely unexplored, which prevents us from discussing their properties with rigorous theoretical groundings. As another example, various kinds of vertex centralities, which measure the importance of vertices, have been proposed and applied to network data. However, the majority of these centralities are computationally hard and efficient (approximation) algorithms should be studied to apply them to real-world networks at a large scale. In theoretical computer science, strong analytical tools for graphs such as graph minor theory and spectral graph theory have been developed and these theories may shed new light on novel characteristics of complex networks.

As we have seen, complex networks are studied in diverse fields from different perspectives. The main goal of this workshop was to bring researchers in these fields together to take the first steps towards bridging works on scalable analytics over complex networks.

1.2 Results of discussions

After thorough discussions between the members of the working groups on several research topics as well as the discussion among the whole members, we reached consensus on four important problems requiring our collaborations: Graph algorithms, Graph characterization, Temporal graphs, and Graph systems (see Fig. 1 for the overview of the topics discussed). Each member summarized from their perspective the problems to be solved, his/her possible contributions, and potential collaborations between us. The following sections will be devoted to summarize the reports of the members.

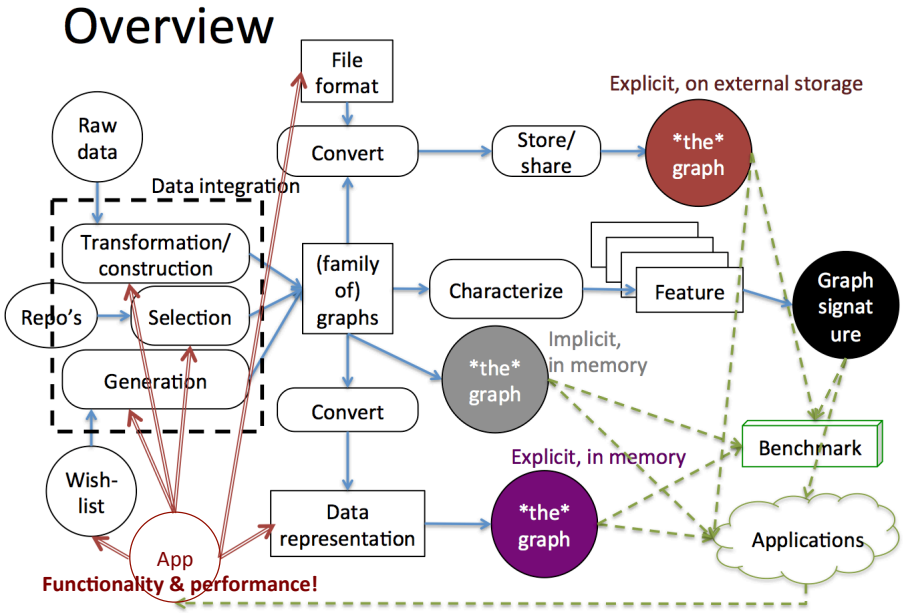

Figure 1: Overview schematic of the discussions, courtesy of Dr. Ana Lucia Varbanescu.

2 Algorithms

2.1 Report by Dragan Bosnacki

Problem 1: Transitive reduction for temporal networks In ordinary static graphs a transitive reduction of a given graph G is the graph R that has the same transitive closure as G and contains the smallest number of edges. In other words, R is the minimal (in the number of edges) graph that preserves the connectivity of G, i.e., if there is a path between nodes u and v in G, then there is a path between u and v also in R and vice versa. It seems that the problem is not trivial. For instance, when a temporal network is represented as a sequence of static networks, or equivalently, as a multilayer network, the assumption of static networks that being directly connected is different from not being connected brakes down. So, techniques from static network theory cannot be applied directly to a graph of the sequence (layer). Interesting subproblems and related questions are:

- Definition of transitive closure for temporal networks
- Definition of transitive reduction for temporal networks
- Algorithms for transitive closure and reduction for temporal networks
- Transitive reduction in the context of different colors of nodes. Assume that the set of graph nodes Is partitioned in different subsets. The requirement is to preserve the connectivity not only on a the level of the entire graph, but also within the subsets (colors) of the graph nodes. This is a combination with the problem idea proposed by Mathias Korman.
- Is the polynomial complexity of the transitive closure and reduction algorithms (essentially boils down to applying of the Floyd-Warshall algorithm) preserved in the context of temporal networks?
- These questions can be considered also in the context of weighted temporal networks
- Applications of temporal transitive reduction

Problem 2: Parallel algorithms based on topological sort with external memory for finding cycles in large networks In the context of model checking, an automated technique for program verification, it is beneficial to have efficient concurrent algorithms for finding special cycles in a graph. There is some existing work on parallel algorithms for finding cycles that boil down to topological sort. Since the huge state space graphs are the main bottleneck for applications of model checking having an algorithm that can efficiently use external memory would increase the size of the models that can be analyzed using model checking. Collaboration with Herman Haverkort.

Problem 3: Identifying relevant features (parameters) of (temporal) biological networks There is some extensive work in the literature on showing that the biological networks are scale free. To sow this, usually some parameters are computed. Some of these parameters/notions, can have direct applications (e.g., metabolites which are hubs in a metabolic networks are biomarkers.)

It would be also interesting to consider applications of more sophisticated approaches presented during the seminar, e.g., dynamic Laplacians. Collaboration with George Fletcher and Mykola Pechenizkiy.

2.2 Report by Herman Haverkort

I would want to work voluntarily with Dragan Bosnacki on topological sorting in external memory: on what graphs do existing algorithms [1] work in practice? Why? In particular: do existing algorithms [1] work for Dragan's application, or do we need to engineer something new?

[1] Deepak Ajwani, Adan Cosgaya-Lozano, Norbert Zeh: A topological sorting algorithm for large graphs. ACM Journal of Experimental Algorithmics 17(1) (2011)

I would want to work voluntarily with David Gleich and/or Matias Korman [unconfirmed]: how to efficiently generate random connected subgraphs of a undirected, connected graph G, according to a uniform probability distribution over all connected subgraphs of G? (purpose: creating the "control group" for testing the statistical significance of findings on a given set of subgraphs of G).

2.3 Report by David Gleich

Problems find a random k-node connected subgraph efficiently to enable significance testing for a set in a graph. ($k = 2$ is a random edge, $k = 3$ is a random triangle or wedge.) it should be a vertex induced subgraph. The goal would be to generate $O(10000)$ samples on a graph with $O(10$ million edges) in $O($minutes), where k is something between 10-250 (say).

Problems What are frequent temporal motifs?

Solutions If you know relevant temporal motifs, then you can partition temporal networks using the Motif-Laplacian work we have done.

Collaborations Yuichi on vertex reinforced random walks with the nonlinear Laplacian?

Collaborations Maybe Jean-Charles?

Collaborations Maybe Herman?

2.4 Report by Yuichi Yoshida

Background and Problems A multi-criteria network is a network for which each edge has two or more types of costs. Many real-world networks are actually multi-criteria networks. For example, a road-network have several kinds of costs such as distance, travel time, and travel fee.

A killer application of a multi-criteria network is the shortest path problem. Suppose that the costs associated with an edge is represented by a vector. Then, given a preference of a user (as a vector), the actual cost of an edge for the user is determined by the inner product of the preference vector and the cost vector of the edge. This means that, given a preference vector, we can think of a weighted

graph constructed by taking inner products with vectors on edges. The question is finding a set of representative paths (between two specified vertices) in the sense that, no matter what the preference vector of a user, one of them is a nearly shortest path for the user.

Algorithmically, a general question is which problem for single-criteria networks can be extended to multi-criteria networks, and whether we can create efficient algorithms for them. From network science view, it is interesting to inspect the structure of a multi-criteria network. For example, it will be interesting to see the spectrum of a betweenness centrality when the preference vector changes.

Collaborations Problems on multi-criteria networks can be often formalized as geometric problems, so it would be possible to collaborate with Matias Korman. He actually says that the problem of finding a set of representative paths have been considered in computational geometry if we forget about the network part and focus on its geometric part.

3 Graph characterization and related problems

3.1 Report by Keisuke Nakano

Graph data generation from given features (e.g., the number of vertices/edges, diameter, and treewidth) and specifications (e.g., logical formula, graph schema, and embedded path/pattern) is useful for benchmarking, verifying, and understanding the graph processing queries/programs. The current generation is either too general where it randomly generates graph without any feature, or too specific where it is required to develop algorithms for the particular feature and specification. It might be helpful to apply our current work of data generation based on reverse processing, in which possible input data is generated by from the query and its output. We can obtain a set of graph data by giving features and schema as a graph query. It is still a difficult issue to guarantee the randomness of graph data generation, though.

3.2 Report by Makoto Onizuka

Findings I found problem:

- The users generate a family of graphs from a single raw data by varying a certain parameter and try to capture how the analysis result changes according to the parameter.
- We can make clusters of various graph mining algorithms that have similar access patterns.
- Temporal graphs are still struggling to find killer apps.
- Data Integration is also an important problem in graph data: alignment, correspondence detection, bisimilarity.
- The centrality and PageRank are similar concept.

Possible contribution

- I can design an efficient mining algorithms for a family of graphs (Alex's example), in particular, graph clustering and PageRank computation. For modularity-base clustering, we compute clusters efficiently by identifying redundant processing among the family of graphs and remove them automatically.
- I can design incremental mining algorithms when graphs are augmented with new entities or bulk of subgraphs. It is possible for PageRank, clustering, and some others, but we should do it efficiently. In the case of modularity clustering, when we get a new bulk of vertex and edges are added, first we compute clusters in the bulk and then we update the previous clusters incrementally. Since we cannot obtain the optimum clusters, we can compute it in greedy manner but the result should be as good as we compute it in batch processing.

3.3 Report by Ana Lucia Varbanescu

Findings
Day1:

- temporal graphs are emerging as an interesting problem and data structure, but the challenges are not yet clear. It is also unclear to me whether for this type of temporal data graphs are the best solution.
- incremental analysis of temporal graphs is an interesting possibility from the perspective of systems and performance.
- benchmarking is a significant missing link in the field of graph processing. We are missing (1) the framework, (2) the metrics and the associated goals (performance evaluation, accuracy, validation, ...), (3) algorithms/data processing, (4) datasets, and (5) reporting the results. In this context, it is interesting to have synthetic generators NOT ONLY for data, but also for workloads.
- data generation is definitely interesting, but state of the art is rather scarce. High-control over the structure of the data AND the basic features of the resulting graph is desirable. It is unclear what are the possibilities and challenges there. It is even more interesting how to generate graphs together with different ground truths for problems where this is necessary.

Day2:
We identified the list of features to be used for graph characterization. We identified 6 different classes of challenges:

- Graph characterization
- Graph generation
- Graph transformation
- Data integration/fusion

- Data representation
- Data storage

Day3:
We improved the discussion from Day2 with examples. We identified a few specific examples for, data generation, data transformation, and data integration. See PPT for these examples.

Future plan My personal research interests are to pursue the following:

1. Graph characterization and graph signatures.

   - I would like to define a close-to-comprehensive list of graph features (for static graphs). I am interesting to design and implement an efficient framework to compute these features.
   - I am interested to work on generalizing this list to temporal graphs and/or dynamic graphs.
   - I am very interested in the idea of graph signatures.

   Possible collaborators (to be confirmed): David, Yuichi, Taro.

2. Graph generation.
   I am very interested in models for graph generation. I would like to evaluate existing generators from the control, generality, and performance perspective. I will be working on a survey of such graph generation techniques.
   Possible collaborators (to be confirmed): Taro, Keisuke, Makoto, David.

3. Data integration.
   - I am interested in working out algorithms for data integration for specific applications.
   Possible collaborators (to be confirmed): Alexandru, Dragan, Matias.

4. Data transformation.
   - I am interesting in applications where data needs to be transformed from "raw" to graphs. Specifically, I am interested in the type of algorithms needed, and in the applications that exist for that.
   Possible collaborators (to be confirmed): Matias, Alexandru.

   - I am interested in multi-modal graphs: constructions and, more importantly, in applications of such graphs.
   Possible collaborators (to be confirmed): ??

   - I am interested in the implicit vs explicit graphs, and the possibilities to incrementally process while transforming data into graphs.
   Possible collaborators (to be confirmed): Andrew

5. Systems, Queries, Programming Models.
   - I am interested in benchmarking, and I will work on this further, especially in the overview of systems and algorithms.
   Possible collaborators (to be confirmed): Alexandru, Aydin, Andrew, Mykola

I am interested in the difference between queries and algorithms, and the potential transformation from one to the other.
Possible collaborators (to be confirmed): George.

My near-future research agenda (implicitly, possible contributions):

- Graph characterization
- Application analysis, algorithm design, performance analysis and modeling for graph applications
- Benchmarking of systems and applications, focusing on dataset generation
- Query-to-algorithm analysis: how to move from queries in database-like systems to algorithms in Programming Models-like systems

3.4 Report by Andrew Lumsdaine

Findings The discussions in the breakout sessions raised a huge number of interesting and important open questions. The primary area that kept coming up in almost all settings was that of "temporal" (similarly, dynamic, incremental, etc) graphs. That the discussion was much more productive in discovering problems rather than solutions is indicative of the richness of this problem area as well as the enormous opportunities.

Some particular needs: 1) Standard terminology and notation for temporal / dynamic graphs 2) Temporal / dynamic algorithms for computing with temporal / dynamic graphs 3) Implementations for computing with temporal / dynamic graphs 4) Killer app(s)

Future Plans Small to medium sized subsets of us should create a number of white papers regarding the open challenges. We should approach appropriate funding agencies for national and international collaborations.

Possible Contributions We have started some theoretical work that could potentially serve as a starting point for reasoning about temporal graphs. In addition, our huge infrastructure for graph computations (BGL, PBGL 1 & 2, PXGL, HPX-5) can provide a platform for rapid exploration of various implementations.

Haiku: Temporal networks
Scalable algorithms
They are connected

3.5 Report by Herman Haverkort

Future plan I will want to work voluntarily with Laetitia Gauvin and Jean-Charles Delvenne on generating random public transportation networks, to be used for the evaluation of algorithms to draw schematic maps of such networks.

3.6 Report by Taro Takaguchi

Findings Three different disciplines of us, theoretical computer science, data engineering, and network science, deal with the same line of problems in common, which was not expected before this workshop. For example, network science needs good benchmark graph models to investigate significant structural feature in real-world networks compared to baseline/reference models. On the other hand, high-performance computing needs it too, to evaluate performance of systems and characterize network features that affect performance.

Future plan Notion of Temporal graphs attracts most of people who are interested in complex networks but we are far from consensus about when the temporality of networks actually matters. Keeping discussions about this point would be fruitful for all fields.

Possible contributions I can provide known results from physical perspective. Examples include: (1) fundamental structural features of static and temporal networks and correlations between the features (2) random graph models which satisfies certain "wish-list" of required network features and generation techniques of them (Monte Carlo simulations, etc).

4 Temporal graphs

4.1 Report by Hang-Hyun Jo

As a "network scientist" in the complex system research, I found a number of interesting questions and issues raised in computer science and engineering, such as: why are we concerned with temporal properties in networks? how can we devise and implement algorithms for calculating relevant quantities (or metrics)? what are the best and most efficient representations for temporal networks?

As a "network modeler" I could contribute to the network generation for collaboration. We can devise a more general framework that provides a family of static/temporal networks, some features of which are fixed (or given), while other features are controllable. The correlations between known features for networks can be more systematically studied. This will tell us which features are genuinely controllable and which are not. This is also very important to understand how and why the networks observed in natural and social phenomena take such specific forms, such as broad distributions of network quantities, community structure, and assortativity.

4.2 Report by Jean-Charles Delvenne

What is the interest of temporal networks? They come in two sorts.

One is to better model real life systems from small scale detailed temporal networks.

For instance from small scale temporal social networks we see that there is memory in the way people establish contacts: it is non Markovian (link-link correlations) or bursty (time between contacts is not memoryless). This has

consequences in the way of modelling spreading eg epidemics or opinion dynamics, usually in the sense of slowing down spreading. This can be incorporated in a large-scale classical model (dynamics quantities such as size of infected population in a node that now represents a city) as a tuning of the parameters of the model, adding a phenomenological correction for memory effects. That does not require specifically clever systems solutions in terms of scalability since we talk about small data. The challenging part is the modelling part. Id say thats what the literature is doing now mainly.

The other possible interest is direct data-driven modelling and simulation for large-scale temporal networks. In this case we can use directly the time-stamped data instead of statistical model for it (Markovian or not, etc.). This is an interesting road for the future.

The latter option requires scaling up the elementary operations for temporal networks such as find (time-consistent) shortest paths, connected components, etc. This requires a careful thinking about data structures, data bases, distributed computing required to treat massive temporal relational data.

I plan to study that latter possibility, for instance with the Dutch team of computer scientists, including George Fletcher, and look for the right language of queries needed to treat common (and challenging) problems of interest for temporal networks, that is also implementable efficiently in a database.

Common and challenging problems of interests on temporal include shortest paths (e.g., Google Maps itineraries with traffic) and recommendation systems for time-stamped content (e.g., news articles that get quickly outdated).

4.3 Report by Laetitia Gauvin

Main topics mentionned during the discussions

1. **Temporal network vs static network.**

   - What is the gain of using a temporal network vs a static network?
     - The temporal dimension obviously enables to capture more information but it also adds complexity. Does the information have to be represented as temporal graphs?

     -Current applications of temporal graphs exist: among them, we can cite the developpment of recommendation systems and applications in the field of epidemiology (immunization strategy). Some work in finding the applications requiring temporal graphs (vs static) still needs to be done but this can be achieved "only" through going back and forth between devising new measures and tools for temporal networks and thinking about possible applications.
   - Which measures/idea from static network can be directly extented to temporal networks? Some quantities can be computed on each of the snapshot in order to look at their evolution. Another question of interest in this context is how to map temporal network to static network in order to re-use existing theory and applications on static network.
   - How to re-think about temporal networks from "scratch"?
     - Definition of new measures specific to temporal network

- Generation of models of temporal networks
- Learning features of temporal networks
- Motif discovery

2. **Challenges in temporal networks in an interdisciplinary context.** An important point of the discussion was to think about challenges in temporal networks and their respective impact on Network science, Analytics and Systems. An example is the case of reachability, a related question in network science is to find influential nodes while the concern in Systems would be to provide support to compute the reachability in multiple ways. The current state of the art on temporal network, makes the navigation between these 3 fields still difficult.

3. **Possible collaborations.**

   Université Catholique de Louvain: for instance, on the use of spectral theory to study temporal network as a whole not as a succession of static network.

   Eindhoven University of Technology: for instance, on transportation network generation: both structural and temporal.

4.4 Report by George Fletcher

Just as every graph is an uncertain graph (we just usually ignore or forget the uncertainties assocatiated with each node/edge), every graph is also a temporal graph (we live in time). What happens when we explicitly consider temporal information in all of our graph reasoning? We shouldn't wait for the "killer app" to start working on the basic science arising from this question.

4.5 Report by Melanie Wu

Definition and interpretation of temporal graphs, especially how users (network scientists) view them and use them. I view temporal graphs as a special case of general graphs where nodes and edges are announced with temporal information, which may be time stamp(s). In addition, temporal graph can be rendered or transformed into either temporal graphs or non-temporal graphs via transformation processes. I think such a view point will help the database community to reason about how temporal graphs can be processed in existing graph database systems and whats amiss and need improvement. From the database point of view, the structural and content of the raw and derived graph will direct how we want to represent and store the data and how the data are derived will direct how queries can be structured to serve the data transformation.

Concerning what are the key problems that trickle down data analytical and system research, the problems I am particularly interested in are problems that can be abstracted as database queries against temporal graphs. In particular, I am interested in discussing with network scientists the semantics of queries on temporal graph that best serve their need, proposing query language on temporal graphs and algorithms for evaluating them. One example is the temporal reachability problem. Open problems are: the algebraic definition of temporal reachability; how data should be pre-processed and indexed (cached); and how

to incorporator existing graph, parallel computing, etc., algorithms to evaluate such queries.

A whitepaper would be a wonderful outcome of this workshop, in which we summarize the view points and existing practices from different research communities, define temporal network, discuss its importance (killer applications), and identify important research directions and research questions.

5 Graph systems

Report of Aydın Buluç, George Fletcher, Alexandru Iosup, and Andrew Lumsdaine.

1. We discussed what is a relevant set of interesting topics related to Systems, Query, and Programming models for graph processing, and the result includes two lists:
   (a) Topics derived from the plenary session, including input from all other participants:
      i. Benchmarking;
      ii. Mutable graph processing and streaming graphs;
      iii. Workflows and execution models for graph processing;
      iv. Query and programming models;
      v. Parallel and distributed system models.
   (b) Topics derived only from the participants in this session:
      i. Storage, persistence, durability, and (higher-level) provenance;
      ii. Cloud and in general service-based graph processing;
      iii. Multi-user, multi-tenant operation;
      iv. Optimization (plan, query, program) and scheduling;
      v. Optimization for (heterogeneous) hardware;
      vi. Resource management, fault tolerance;
      vii. Relationship between temporal graphs and systems, query, and programming models.

      As exemplary problems and applications, we have identified overall:
      - Online gaming and meta-gaming, for which analytics and recommendations are important operations, for example to improve the quality of gameplay experience by grouping matching players or by making sure same-guild players get enough resources to play together, to prevent toxicity from appearing and from spreading, etc.
      - Sports such as the US MLB, for which analytics are important in deriving game strategies and tactics.
2. We discussed in detail three of the topics identified at point 1:
   - (point 1(a)ii) Mutable graph processing and streaming graphs;
   - (point 1(a)iv) Query and programming models; and
   - (point 1(a)v) Parallel and distributed system models.

For each, we have discussed the key concepts, the key problems and stakeholders, state-of-the-art and direct contribution to it from the participants, important research questions, and various collaboration opportunities.

Among the stakeholders, we identified overall:

- DevOps teams creating graph processing products. The Ops make sure interference between concurrent users is manageable, and overall that the system can continue its operation and meet the service level agreements. The Devs primarily ensure the core operations of the application, by developing techniques for partitioning and processing, and by packaging important common parts into libraries.
- In cloud settings, it is possible to have layers of DevOps, developing and running the different elements of a value chain (e.g., IaaS, PaaS, PaaS with added value, lab/company gateway, real users). Integrators of graph processing products also fit here.
- Application operators and operators of groups of applications, such as the game or tournament operator, who consume graph processing services.
- Single or small independent users of applications, who can also consume graph processing services using relatively simple analytics tools.
- Businesses specialized in the accreditation, assessment, and mediation of graph processing services. Benchmarking companies and their products fit here.

Collaboration opportunities could include:

- US-NL collaboration on big data, especially on privacy, brain mapping, and other important core scientific and other societal topics.
- Colaboration with Japan, which could include Satoshi Matsuoka (TITECH), the RIKEN labs, and Makoto Onizuka (U. Osaka).

Details about the topics discussed in more detail follow next in point 3.

3. Topics discussed in detail.

   We report here only the main research questions we have identified during the discussions, for each topic, in turn.

   (a) Mutable graph processing and streaming graphs.
   Research questions:

      i. (query and programming models) How to model system capabilities for programmers to use? This addresses primarily the situation where system executes low-level operations, while users typically create code in high-level DSMs.
      ii. (systems) How to partition data for workloads (algorithms, data) of streaming graphs to increase performance/efficiency? How to partition data for workloads (algorithms, data) of mutable graph processing to increase performance/efficiency?

iii. (systems) How to process workloads of streaming graphs in-time and, if possible, efficiently (i.e., with minimal memory footprint)? ... of mutable graph processing with low computational and memory overhead? – This topic requires a revisit of the database concept of views computed by queries/caches (especially maintenance of views). Similarly, revisit checkpoint/restart and/or transaction logging in distributed systems and databases.

iv. How to build efficient/high performance caches and/or replicas? This could also lead to machine learning techniques, but also to simple heuristics, hierarchies of caches in the NUMA spectrum.

v. How to manage persistent storage and/or checkpointing backends when supporting multiple users? Revisit background loads in datacenters and databases.

vi. How to design eternal-memory algorithms/out-of-core algorithms/single-pass algos than match the systems characteristics? SST processing - vertices in memory, (some) edges on disk?

vii. How to evaluate success in the design of a new approch? How to compare approches? Conducting a survey of existing benchmarking approaches is meaningful here. Using benchmarking is not the only approach possible here.

(b) Query and programming models.

Main research questions:

i. What are the DSL's (domain specific languages) for the "new" types of graphs we are working on in the community (temporal-, spatial-, uncertain-, dynamic-, streaming- graphs)? How to support path query processing? Unearthing the basic logics underlying analytics in these domains would be the interface between data/domain scientists and systems researchers (just as SQL is the interface in the tabular data world).

ii. What are the (intrinsic vs system-dependent) performance, reliability/availability, non-func limits of a specific query/programming model? To what extent does a query model impose limits to graph-partitioning and -processing?

iii. What are good patterns in engineering graph-partitioning and -processing software using a query/programming model?

iv. How to evaluate success in the design of a new query/programming model? How to compare query/programming models? Conducting a survey of existing benchmarking approaches is meaningful here. Using benchmarking is not the only approach possible here.

v. The "primitives" problem. There are many linear algebra libraries but virtually all use BLAS because not using BLAS would be a suicide. Could the graphs domain expand on the existing GraphBLAS? Are there other useful primitives that are non-linear algebraic?

vi. Are there particular cases of graph (types) and/or algorithm (types) lead to big execution penalties, when using a specific query/programming model?

vii. What are good query/programming models, and DSLs, for specific application domains, that also do not lose too much performance, resilience, non-func properties?

(c) Parallel and distributed system models.

Main research questions:

i. How to build a standard theory of graph processing, and to link it to working systems? Can we also include in such models graph processing characteristics, such as graph structure and chars, algo structure and chars, etc.? Are there also "abstractions" (see outcome of point/question 3(b)v) that are fundamental and can be analyzed?

ii. How to design resource management and scheduling approaches for parallel and distributed graph-proc systems?

iii. How to support distributed+parallel (heterogeneous) processing of graphs?

iv. How to use modern hardware to accelerate/overcome existing overheads (FPGAs, FusionIO, etc.)?

v. How to support many users in the same graph proc environment? how to support their migration and/or operation across multiple envs?

vi. How to build dynamic, adaptive runtimes?

vii. Lightweight vs heavyweight system designs – which to use?

viii. how to scale efficiently (laptop to exaflop)? how to be elastic, if cloud service?

ix. How to be resilient at scale, esp for brittle (commodity) hardware?

x. How to support HPC and HTC on the same hardware?

xi. How to manage background load for value-adding services without interfering with the operation of the users of the system?

xii. How to evaluate success in the design of a new system? (benchmarking, survey)

xiii. How to compare different systems? (benchmarking, survey)

xiv. What is relevant workload: truly large-scale problems, large-scale workloads (many users), ...?

xv. How to get access to an archive of operational data to tune systems?

List of Participants

- Dragan Bosnacki, Eindhoven University of Technology
- Aydın Buluç, Lawrence Berkeley Laboratory
- Jean-Charles Delvenne, Université catholique de Louvain
- George Fletcher (organizer), Eindhoven University of Technology
- Laetitia Gauvin, ISI foundation
- David F. Gleich, Purdue University
- Herman Haverkort, Eindhoven University of Technology
- Alexandru Iosup, Delft University of Technology
- Hang-Hyun Jo, POSTECH
- Matias Korman, Tohoku University
- Andrew Lumsdaine, Indiana University
- Keisuke Nakano, The University of Electro-Communications
- Makoto Onizuka, Osaka University
- Juyong Park, KAIST
- Mykola Pechenizkiy, TU Eindhoven
- Taro Takaguchi (organizer), NII
- Ana Lucia Varbanescu, University of Amsteram
- Yuqing Melanie Wu, Pomona College
- Yuichi Yoshida (organizer), NII

Meeting Schedule

Check-in Day: February 7th (Sun)

- 15:00-18:30 Check-In in Shonan Center
- 19:00-20:30 Welcome Dinner
- 21:00- Free Time

Day1: February 8th (Mon)

- 07:30-09:00 Breakfast
- 09:00-09:10 Shonan Introduction by Staff
- 09:10-12:00
  - Presentations by the organizers
  - Short introduction by each member
- 12:00-14:00 Group photo and Lunch
- 14:00-18:00
  - Short introduction by each member (cont.)
  - Group formation (4 groups of 5 members each)
  - Group break-out discussions
  - Goal: to identify multidisciplinary (broad) research themes
- 18:30-19:30 Dinner
- 19:30- Free Time

Day2: February 9th (Tue)

- 09:00-12:00
  - Plenary discussions by the groups
  - Goal: to present results of group discussions (20min for each) & select 4 most interesting themes to explore more deeply
  - Group reformation around interests of 4 themes
- 12:00-14:00 Lunch
- 14:00-18:00
  - Group break-out discussions
  - Goal: to identify concrete interdisciplinary research problems within group themes
- 18:30-19:30 Dinner
- 19:30 Free Time

Day3: February 10th (Wed)

- 07:30-09:00 Breakfast
- 09:00-12:00
  - Plenary discussion
  - Goal: to present and share the results of the group break-out discussions on concrete problems
- 12:00-14:00 Lunch
- 14:00-21:00 Excursion (including dinner)
- 21:00- Free Time

Day4: February 11th (Thu)

- 07:30-09:00 Breakfast
- 09:00-12:00
  - Plenary discussion to consolidate problems
  - Plenary discussion: multidisciplinary action routes (e.g., new research collaborations, research exchange visits)
- 12:00-14:00 Lunch
- 14:00- Departure