論文 / 著書情報
Article / Book Information

題目(和文)	大規模動的グラフ解析の性能最適化手法
Title(English)	Performance Optimization of Large-Scale Dynamic Graph Analytics
著者(和文)	金刺宏樹
Author(English)	Hiroki Kanezashi
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第11222号, 授与年月日:2019年6月30日, 学位の種別:課程博士, 審査員:松岡 聡,増原 英彦,遠藤 敏夫,脇田 建,額田 彰
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第11222号, Conferred date:2019/6/30, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	論文要旨
Type(English)	Summary

（博士課程）
Doctoral Program

論 文 要 旨

THESIS SUMMARY

専攻： Department of	数理・計算科学　　専攻	申請学位（専攻分野）： Academic Degree Requested	博士 Doctor of　（　理学　）
学生氏名： Student's Name	金刺 宏樹	指導教員（主）： Academic Supervisor(main)	特任教授　松岡 聡
		指導教員（副）： Academic Supervisor(sub)	

要旨（英文 800 語程度）
Thesis Summary（approx.800 English Words ）

Large-scale graph data analysis has been necessary for real-world analytics. Discrete simulations using graph structures are also essential to investigate and validate more detailed analytic results. Moreover, the scale of data has become much larger, and HPC systems with ample memory space and multiple processors and computation nodes are indispensable to conduct such analytics and simulations.
However, many performance issues happen, such as load imbalances and communication overheads. These problems usually occur while running an application handling dynamic graph data. In an agent-based traffic application, for example, even the simulation runtime divides road network equally the number of agents for each partition becomes imbalanced, and the overall application will not scale.
To address the problem, we proposed performance optimization methods for dynamic graph data management based on the structures. We set several goals of the performance optimizations: to gain scalability of parallel processing by the reduction of costs such as load imbalance and synchronization overhead and to reduce computation costs with effective graph analytics for dynamic graph analytics with incremental processing and adaptive parameter optimizations.
First, we proposed the performance optimization method to adjust synchronization intervals among all threads and processes in a parallel and distributed agent-based traffic simulation. It skips synchronization phases and allows all the distributed simulation processes to run individually for most simulation steps. We also proposed an adaptive migration method for an agent-based simulation framework to re-assign agent objects to threads and remote processes. From the previous execution logs, it partitions the road networks, and then it automatically re-assigns computation processes and threads to cross points and vehicle agents while running simulations. We evaluated these proposed methods in parallel and distributed computation nodes of TSUBAME 2.5 supercomputer with real road network and commutation trip data in the Dublin city. The synchronization interval adjustment method achieves up to 35% speed-up in more than four nodes, and dynamic agent object assignment method across computation nodes achieves 25% speed-up on average.
Second, in order to minimize the computation time of graph analysis for time-evolving graphs, we proposed incremental graph analytics algorithms.
We proposed an incremental community detection algorithm named IncrementalDEMON for time-evolving graphs more efficiently as an extended algorithm of the existing community detection algorithms to support incremental processing with vertex and edge additions or deletions. The IncrementalDEMON consists of three core functions to construct overlapping communities, and we added an incremental feature each of them. In the evaluation, we used time-evolving social networks with millions of edges with timestamps, and our incremental community detection algorithm processed each iteration of community detections in 1/101 execution time compared to the original non-incremental algorithm named DEMON.
We also proposed an incremental graph pattern matching (IGPM) algorithm as an extension of an existing approximate subgraph isomorphism algorithm, and parameter adjustment framework named Partial Execution Manager (PEM) with reinforcement learning to determine the vertex sets for re-computations. Similar to the IncrementalDEMON, IGPM algorithm consists of three core functions which are incrementally extended to find the most likely matching vertices and edges in the best effort. The PEM consists of graph clustering and reinforcement learning components. The graph clustering component detects the vertex sets around the updated vertices and edges for re-computations. The reinforcement learning component adjusts the granularity of the result of graph clustering with the size and density of data graph as input and the throughput of IGPM as the reward for finding more matched patterns.
Our incremental graph pattern matching processed each iteration in 1/10.1 execution time compared to the baseline non-incremental subgraph isomorphism algorithm named G-Ray, and our adaptive optimization

framework processed graph pattern matching processed each iteration in 1/14.8 execution time compared to G-Ray.
Finally, we made qualitative evaluations and discussions for our proposed methods about adaptive parameter adjustment from environments and incremental graph algorithms such as the applicability and limitations for other types of real-world graphs, other graph algorithms, and agent-based simulations. We validated how these methods can apply to other real-world applications and graph datasets with several representative cases, and how each optimization method affects the precision of outputs such as the travel time of each vehicle in the traffic simulation and subgraph size distributions in the community detections and pattern matchings.
This thesis presents some contributions to discover the movement of real-world with a graph structure. Our series of researches can apply to various fields of social data analytics applications with other graph algorithms and social simulations with more extensive and time-evolving graph data.

備考 ： 論文要旨は、和文 2000 字と英文 300 語を 1 部ずつ提出するか、もしくは英文 800 語を 1 部提出してください。
Note : Thesis Summary should be submitted in either a copy of 2000 Japanese Characters and 300 Words (English) or 1copy of 800 Words (English).