論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Automatic Tuning and Software Modularization Techniques for Optimized Large Scale Data Analytics
著者(和文)	MIYURUDAYARATHNA
Author(English)	Miyuru Dayarathna
出典(和文)	学位:博士(学術), 学位授与機関:東京工業大学, 報告番号:甲第9347号, 授与年月日:2013年9月25日, 学位の種別:課程博士, 審査員:鈴村　豊太郎,徳田　雄洋,横田　治夫,村田　剛志,小林　隆志
Citation(English)	Degree:Doctor (Academic), Conferring organization: Tokyo Institute of Technology, Report number:甲第9347号, Conferred date:2013/9/25, 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	計算工学 専攻	申請学位（専攻分野）： Academic Degree Requested	博士 Doctor of Philosophy （ 学術 ）
学籍番号： Student ID Number		指導教員（主）： Academic Advisor(main)	鈴村 豊太郎
学生氏名： Student's Name	Miyuru Dayarathna	指導教員（副）： Academic Advisor(sub)	

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

Data intensive large scale software systems need optimized software architectures focusing on concurrency, and efficiency. Currently MPI, MapReduce, and stream programming models are famous for programming data intensive tasks. However, they are less optimal in terms of programmability, maintainability, and execution performance which are fundamental problems that need to be addressed in future systems. It is a technically challenging problem because it is very hard to achieve a single generic solution in variety of systems, and programming environments. The hardware environments change frequently hence the applications need to adapt for such dynamic environments. This thesis proposes new software modularization, automatic code transformation, and benchmarking techniques as solutions for improving productivity, and execution performance of large scale data analysis systems. Specifically, it focuses on optimizing the current state of the art stream/graph data processing paradigms via the aforesaid techniques. The thesis consists of 12 chapters which spans across three main sub sections.

First half of the thesis describes the contributions made for performance optimization of stream data analytics. The thesis investigates on new methods for workload characterization, and benchmarking of large scale data analytics. The thesis evaluates performance of stream processing system architectures using System S, S4, and Esper stream processing software in chapter 4. It was observed that stream processing applications having higher operator density at the beginning of the flow graph tend to produce better performance. By considering the results obtained from the performance analysis and empirical results in previous work, the thesis proposes techniques for automatic optimization of stream applications in chapter 6. The thesis introduces operator graph transformation, a notion for improving the concurrency of stream analytics via automatic software tuning. The thesis introduced novel algorithms for transformation of data stream programs called Tri-Operator, Bi-Operator transformation which produces large sample program space. The sample program space is pruned using profiling/performance prediction techniques to obtain high performance versions of the stream applications. It was observed that the proposed code transformation and automatic tuning techniques produce improved performance (E.g., 31.1, 2.7, 2.2 times performance improvement for CDR, VWAP, and Twitter input applications). Thesis compared the proposed code transformation techniques to fusion only technique and showed that the proposed technique outperforms the performance of purely fusion based optimization mechanisms. The stream application code transformation techniques were extended to the areas of multiple stream job performance optimization and performance

maintenance of stream programs.

The second half of the thesis describes graph analytics optimization. In order to understand performance features of graph database systems and also to create realistic benchmarking scenarios, the thesis implemented a scalable graph database benchmark. The thesis demonstrated the ability of modeling real world attribute graphs by comparing the benchmark data generator's synthetic graph model (the MAG model) with famous R-MAT model through analyzing the cluster prominence. The thesis conducted a performance analysis of graph database servers using the new benchmark. The thesis describes the importance of having well defined abstractions for large graph analysis tasks. In order to achieve this objective, the thesis described the architecture of ScaleGraph 1.0, a library for analyzing graphs in the scale of billions of vertices, and edges in chapter 10. ScaleGraph has been designed following object-oriented concepts targeting distributed memory and heterogeneous environments and was implemented in X10 language. In Tsubame 2.0 distributed memory computing environment it was observed that ScaleGraph 1.0 is able to calculate in-degree distribution of 69 million edges LiveJournal social network within 998 seconds. In chapter 10 the thesis implemented a domain specific language for large graph analysis called Exerda, and showed how such languages can support specification of optimized graph data analysis tasks efficiently by using multiple graph analysis software such as ScaleGraph.

Thesis identifies future expansions in the areas of high productive software development, massive graph analysis, software maintainability. Development of a distributed graph database server that operates in a hybrid cloud environment is one such example directions for future work. Another research direction is implementation of performance prediction in multipath DAGs.

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 2 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 2 copies of 800 Words (English).