
Medusa A Parallel Graph Processing System On Graphics

Kindle File Format Medusa A Parallel Graph Processing System On Graphics

Getting the books [Medusa A Parallel Graph Processing System On Graphics](#) now is not type of challenging means. You could not deserted going taking into account ebook stock or library or borrowing from your friends to approach them. This is an agreed simple means to specifically get guide by on-line. This online message Medusa A Parallel Graph Processing System On Graphics can be one of the options to accompany you in imitation of having other time.

It will not waste your time. understand me, the e-book will unconditionally tell you new issue to read. Just invest little era to right of entry this on-line broadcast **Medusa A Parallel Graph Processing System On Graphics** as with ease as review them wherever you are now.

Medusa A Parallel Graph Processing

Medusa: A Parallel Graph Processing System on Graphics ...

Medusa is a parallel graph processing system on graphics processors (GPUs) The core design of Medusa is to enable developers to leverage the massive parallelism and other hardware features of GPUs by writing sequential C/C++ code for a small set of APIs This simplifies the implementation of parallel graph processing on the GPU

Medusa: Simplified Graph Processing on GPUs

21 Graph Processing Parallel algorithms have been a classical way to improve the performance of graph processing On multi-core CPUs, parallel libraries such as MTGL [7] have been developed for parallel graph algorithms Similar to Medusa, MTGL offers a set of data structures and APIs for building graph algorithms The

Medusa: Simplified Graph Processing on GPUs

21 Graph Processing Parallel algorithms have been a classical way to improve the performance of graph processing On multicore CPUs, parallel libraries such as MTGL [7] have been developed for parallel graph algorithms Similar to Medusa, MTGL offers a set of data structures and APIs for building graph algorithms The MTGL API is modeled after

Medusa - University of Cambridge

Graph processing algorithms are often inherently parallel GPUs consist of many processors running in parallel But... writing this code is hard The Solution Medusa is a C++ framework for graph processing on (multiple) GPUs High programmability (expressive) Related Work MTGL Parallel graph library for multicore CPUs Pregel

PARALLEL GRAPH PROCESSING ON GRAPHICS PROCESSING ...

processing due to the irregularities of graph structures Even worse, there lacks an efficient method and runtime system on the GPU to support concurrent graph processing tasks from multiple applications and/or users To address those challenges, we develop the Medusa system to simplify parallel graph processing on the GPU and to support high

Optimizing Graph Processing on GPUs

Medusa provides a more fine-grained programming interface than Pregel, exposing fine-grained data parallelism on edges, vertices and messages, which is called EMV model This model enhances the vertex-centric model to provide support for efficient graph processing on GPUs Using the APIs offered by Medusa, programmers can define

(THE LANDSCAPE OF) PARALLEL GRAPH PROCESSING: A ...

(THE LANDSCAPE OF) PARALLEL GRAPH PROCESSING: A VIEW FROM HOLLAND What to do when your graphs get out of control ? Ana Lucia ...

Gunrock: A High-Performance Graph Processing Library on ...

rectly for different graph algorithms In Medusa [30], Zhong and He presented their pioneering work on GPU-based programming model for parallel graph processing using message passing model CuSha [16] implements the parallel-sliding-window (PSW) graph representation on the GPU to avoid non-coalesced memory access Both frameworks offer a small

Scalable SIMD-Efficient Graph Processing on GPUs

Scalable SIMD-Efficient Graph Processing on GPUs Farzad Khorasani Rajiv Gupta Laxmi N Bhuyan Computer Science and Engineering Department University of California Riverside, CA, USA {fkhor001, gupta, bhuyan}@csucredu Abstract—The vast computing power of GPUs makes them an attractive platform for accelerating large scale data parallel

A Distributed Multi-GPU System for Fast Graph Processing

We present Lux, a distributed multi-GPU system that achieves fast graph processing by exploiting the aggregate memory bandwidth across a multi-GPU cluster In Lux, the entire graph representation is distributed onto the DRAM and GPU memories of one or multiple nodes The distributed graph placement is designed to minimize data trans-

SEP-Graph: Finding Shortest Execution Paths for Graph ...

SEP-Graph: Finding Shortest Execution Paths for Graph Processing under a Hybrid Framework on GPU Hao Wang†, Liang Geng†§, Rubao Lee‡, Kaixi Hou¶, Yanfeng Zhang§, Xiaodong Zhang†* †Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA,

Accelerating Dynamic Graph Analytics on GPUs

stream processing and lack support for graph stream processing Stinger [19] is a parallel solution to support dynamic graph analytics on a single machine More recently, Kineo-graph [14], CellIQ [28] and GraphTau [27] are proposed to address the need for general time-evolving graph processing under the distributed settings However, to our

Review of Graph Algorithms on GPU using CUDA Architecture

Review of Graph Algorithms on GPU using CUDA Architecture Trupti R Desale Student, Computer Engg Department, MCOERC Nashik for execution on a parallel processing device like the GPUs a software interface, CUDA API can be defined as graph processing algorithm Medusa hides a GPU specific programming details with a small set of system

Review Paper on Optimised and Accelerated Parallel Graph ...

CUDA kernels Medusa storage component allow developers to initialize the graph structure through the use of system APIs like Add Edge and Add

Vertex then Medusa runtime component which is responsible for executing the user-defined APIs in parallel on GPU Medusa BFS method, the Medusa applies L threads to vertex has L edges

Performance Characterization of Multi-threaded Graph ...

Performance Characterization of Multi-threaded Graph Processing Applications on Many-Integrated-Core Architecture Lei Jiang Langshi Chen Judy Qiu School of Informatics, Computing, and Engineering, Indiana University Bloomington {jiang60, lc37, xqiu}@indianaedu Abstract—In the age of Big Data, parallel graph processing has

Efficient and Simplified Parallel Graph Processing over CPU ...

Efficient and Simplified Parallel Graph Processing over CPU and MIC Linchuan Chen Xin Huo Bin Ren Surabhi Jain Gagan Agrawal Department of Computer Science and Engineering The Ohio State University Columbus, OH 43210 {chenlinc,huox,ren,jainsu,agrawal}@cseohio-stateedu Abstract—Intel Xeon Phi (MIC architecture) is a relatively

final USC-2016

Parallel Graph Processing 2 - Limited Memory Storage System - Limited Device Memory & Bandwidth - Communication High Latency (GraphChi, XStream etc) GPUs + Massive Parallelism Power Efficiency Clusters + Scalability Memory & Cores Multicores + Efficient Parallelism GPUs (VWC, Medusa, Totem etc) Clusters (GraphLab, GraphX etc) Multicores

MultiGraph: Efficient Graph Processing on GPUs

ent GPU graph processing frameworks in terms of their merits and challenges CuSha [11]: The CuSha framework was the first to address the limitation of uncoalesced global memory data accesses for GPU graph processing, by performing updates in shared memory Instead of the standard CSR data structure, it uses an alternative G-Shard representation

Poster P5260 Towards Realizing Topology Mutation for ...

Towards Realizing Topology Mutation for Iterative Graph Processing on a GPU??? ?!?! ¾ Useful to realize pointer jumping employed in many parallel graph Medusa : Simplified graph processing on GPUs , IEEE Trans Parallel and Distributed System , 25 (6):1543 -1556 ,

Performance Characterization of Multi-threaded Graph ...

Performance Characterization of Multi-threaded Graph Processing Applications on Many-Integrated-Core Architecture Lei Jiang Langshi Chen Judy Qiu School of Informatics, Computing, and Engineering, Indiana University Bloomington {jiang60, lc37, xqiu}@indianaedu Abstract—In the age of Big Data, parallel graph processing