

Parallel Processing

MT18_HK1_2020_2021

Ho Chi Minh City University of Technology

References

1. Introduction to Parallel Computing: https://computing.llnl.gov/tutorials/parallel_comp/
2. Introduction to Parallel Computing: From Algorithms to Programming on State-of-the-Art Platforms
http://www-e6.ijs.si/~roman/files/Book_jul2018/book/book.pdf
3. Parallel Computing - theory and practice:
http://www.cs.cmu.edu/afs/cs/academic/class/15210-f15/www/tapp.html#_preface
4. <http://www.csd.uwo.ca/~moreno/HPC-Resources.html>
5. Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, Barry Wilkinson and Michael Allen, Second Edition, Prentice Hall, 2005.
<https://www.cse.unr.edu/~fredh/class/415/text/pp-2ed/parallel.pdf>
6. Open MP: (1) <https://www.openmp.org>, (2) <https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5.0.pdf>
7. MPI: <http://www.mcs.anl.gov/research/projects/mpi/tutorial/>
8. Map/Reduce & Hadoop: <https://hadoop.apache.org>
9. Spark: <https://spark.apache.org>
10. https://www.researchgate.net/publication/2578544_Computational_Models_For_Parallel_Computing_and_BSPlab
11. <http://parallelcomp.uw.hu/ch02lev1sec4.html>
12. http://www.compsci.hunter.cuny.edu/~sweiss/course_materials/csci493.65/lecture_notes/chapter06.pdf
13. Distributed Algorithms, Nancy Lynch, Morgan Kaufmann, 1997.
14. Scalable Parallel Computing: Technology, Architecture, Programming, Kai Hwang & Zhiwei Xu, McGRAW-HILL, 1997.
15. Designing and Building Parallel Programs, I. Foster, Addison Wesley, ISBN 0-201-57594-9, 1995, <https://www.mcs.anl.gov/~itf/dbpp/text/node1.html>

Lectures

Lectures (90 min)	Topics	References	Notes
Lecture 1	Introduction	[1-5]	
Lab 1	Lab: Introduction & multithreading		T. Nguyễn Mạnh Thìn
Lecture 2	Abstract machine models – PRAM & BSP Multithreading OpenMP	[1, 2, 10] [3] [6, 2, 5]	
Lab 2	Lab: OpenMP (1) Xeon Phi		
Lecture 3	MPI	[7, 2, 5]	
Lab 3	Lab: OpenMP (2) & Xeon Phi		
Lecture 4	Parallel machine architectures: – Flynn classifications – Pipeline, Processor array, Multiprocessor, Data flow computer – Processor organizations	[1, 2, 11]	
Lab 4	Lab: MPI – Point-to-point communication		
Lecture 5	Speedup: – Amdahl – Gustafson	[1, 2, 3, 5, 12]	
Lab 5	Lab: MPI – Collective communication		
Lecture 6	Map/Reduce	[8]	
Lab 6	Lab: Hadoop 1		
Lecture 7	Parallel & distributed computing techniques (1) – EPC – Partition, Divide & Conquer – Pipeline	[5]	

Lab 7	Lab: Hadoop & Spark		
Lecture 8	Parallel & distributed computing techniques (2) <ul style="list-style-type: none"> – Synchronous computations – Asynchronous computations – Load balancing 	[5]	
Lab 8	Lab: GPU		
Lecture 9	Parallel algorithms	[2, 3, 4, 5]	
Lab 9	Lab: Algorithms 1		
Lecture 10	Review		
Lab 10	Lab: Algorithms 2		

HPC Lab

- SuperNode-XP 24 nodes x (2 CPUs x 12 cores, 2 Xeon Phi x 61 cores, 512/256/128 GB RAM, 1 TB HD/SSD), Infiniband 56 Gbps
- GPUs system: P100, P4

Evaluation

- Final exam: 40% HK
- Project: 30% HK
- Lab: 30%

Contact:

- Thoại Nam: namthoai@hcmut.edu.vn
- **Đại diện lớp (Lớp trưởng):**

Class:

- Lecture: H2-403, Monday 2-3-4 (from 21 Sep 2020)
- Lab:

Dataset

- (1) https://hadoopilluminated.com/hadoop_illuminated/Public_Bigdata_Sets.html
- (2) Many free open data sources in Internet

Đề Project môn TTSS

Quy định:

1. Mỗi nhóm tối đa 2-3 sinh viên (trùng với nhóm seminar).
2. Các nhóm đăng ký danh sách và đề tài cho Trưởng lớp, **hạn cuối 5/10/2020**:
 - Ghi rõ họ tên, MSSV, email của các thành viên trong nhóm
3. Nộp báo cáo 2-4 trang mô tả nội dung đề tài thực hiện, nguồn dữ liệu, dự kiến kết quả, **hạn cuối 19/10/2020**
4. Nộp báo cáo sơ bộ về tiến độ và kết quả đạt được, **hạn cuối 23/11/2020**
5. Nộp báo cáo cuối kỳ, **hạn cuối 28/12/2020**, tất cả các nhóm phải nộp:
 - Báo cáo tối đa 8 trang A4
 - Mã nguồn.

Gợi ý (khuyến khích chủ đề từ SV):

Đề 1: Viết chương trình nhân ma trận trên nhiều máy tính dùng MPI One-Sided Communication.

Lý thuyết:

- Tìm hiểu các giải thuật nhân ma trận sử dụng MPI
- Tìm hiểu MPI One-Sided Communication

Hiện thực:

- Viết chương trình nhân ma trận: (1) không sử dụng MPI One-Sided Communication & (2) sử dụng MPI One-Sided Communication
- So sánh thời gian thực thi trong 2 trường hợp trên với các kích thước ma trận từ 100×100 , 1.000×1.000 , 10.000×10.000 khi sử dụng mạng giao tiếp: (1) Gigabit Ethernet 1Gbps và (2) Infiniband 40 Gbps trên hệ thống SuperNode-XP.
- Vẽ biểu đồ đánh giá hiệu năng (speedup) trong các trường hợp.

Đề 2: Viết chương trình nhân ma trận dùng giải thuật Strassen

Lý thuyết:

- Tìm hiểu phương pháp chia để trị (Divide-and-Conquer)
- Tìm hiểu giải thuật nhân ma trận Strassen
- Tìm hiểu về lập trình multithreading

Ref:

- <https://www.mcs.anl.gov/prism/lib/techsrc/w333.ps>
- <https://www.cse.unr.edu/~fredh/class/415/text/pp-2ed/parallel.pdf>

Hiện thực:

- Viết chương trình:
 - Load ma trận A & B
 - Dùng Divide-and-Conquer để chia nhỏ bài toán
 - Dùng giải thuật Strassen để thực hiện (có thể dùng các thư viện đã có)
- So sánh thời gian thực thi với các kích thước ma trận từ 100×100 , 1.000×1.000 , 10.000×10.000 với các giải pháp Divide-and-Conquer khác nhau và số lượng core tính toán trong máy vật lý tối đa là 4, 8 & 16 cores
- Vẽ biểu đồ đánh giá hiệu năng (speedup) trong các trường hợp trên.

Đề 3: Viết chương trình và đánh giá về giải thuật PageRank

Lý thuyết:

- Tìm hiểu giải thuật PageRank và Personalized PageRank
- Tìm hiểu chương trình PageRank và Personalized PageRank

- Tìm hiểu giải pháp song song/phân tán hay tính nhanh cho PageRank và Personalized PageRank

Ref:

- *PageRank*: <http://infolab.stanford.edu/~ullman/mmds/ch5.pdf>
- *Personalized PageRank*:
https://cs.stanford.edu/people/plogfren/bidirectional_ppr_thesis.pdf
- *Distributed Algorithms for Fully Personalized PageRank on large-scale graph*:
<https://arxiv.org/pdf/1903.11749.pdf>

Hiện thực:

- Viết chương trình song song/phân tán về PageRank và Personalized PageRank
- Đánh giá hiệu năng giải thuật.

Bài 4: Viết chương trình tìm các đối tượng gần nhau (Finding Similar Items) sử dụng Min-Hash và LSH (Locality Sensitive Hashing)

Lý thuyết:

- Tìm hiểu Min-Hash và LSH
- Phát triển giải thuật Min-Hash và LSH
- Phát triển giải thuật song song cho Min-Hash và LSH

Hiện thực:

- Viết chương trình & đánh giá hiệu suất trên hệ thống thực.

Ref:

- Min-Hash và LSH:
(1) <http://infolab.stanford.edu/~ullman/mmds/ch3.pdf>
(2) <http://web.stanford.edu/class/cs246/slides/03-lsh.pdf>
(3) http://web.stanford.edu/class/cs246/slides/04-lsh_theory.pdf

Đề 5: Giải thuật tìm kiếm (gần đúng) trong tập dữ liệu lớn (search techniques for large-scale data set/big data).

Lý thuyết:

- Tìm hiểu về bài toán tìm kiếm trong tập dữ liệu lớn vì thời gian & không gian tìm kiếm lớn
- Lựa chọn/phát triển một giải thuật song song tìm kiếm trong tập dữ liệu lớn.

Hiện thực:

- Hiện thực giải thuật
- Phân tích kết quả.

Ref:

- https://www.researchgate.net/publication/322994594_A_survey_of_different_search_techniques_for_big_data

Đề 6: Đánh giá hiệu năng của Spark trong các bài toán phân tích dữ liệu lớn.

Lý thuyết:

- Tìm hiểu về Spark & thư viện về phân tích dữ liệu lớn & học máy
- Triển khai thực tế trên hệ thống máy tính mạnh SuperNode-XP.

Hiện thực:

- Chạy các ứng dụng để đánh giá hiệu năng Spark trên SuperNode-XP
- Phân tích kết quả.

Ref:

- Spark: <https://spark.apache.org>
- Spark & Hadoop: <https://towardsdatascience.com/big-data-analytics-apache-spark-vs-apache-hadoop-7cb77a7a9424>

Đề 7: Đánh giá hiệu năng của các thư viện về Deep Learning như TensorFlow, Torch, Caffe... trên các loại GPU cards khác nhau.

Lý thuyết:

- Tìm hiểu về Deep learning và các thư viện TensorFlow, Torch, Caffe
- Triển khai thực tế trên hệ thống máy tính mạnh SuperNode-XP dùng các loại GPU như P100, GTX 1080Ti, P4, trên CPUs.

Hiện thực:

- Chạy các ứng dụng để đánh giá hiệu năng các thư viện TensorFlow, Torch, Caffe trên SuperNode-XP với các loại GPU khác nhau
- Phân tích kết quả.

Đề 8: Trục quan hoá các phương thức lập lịch static, dynamic cho các threads trong OpenMP.

Lý thuyết:

- Tìm hiểu về lập trình OpenMP
- Tìm hiểu về lập lịch static, dynamic
- Tìm hiểu công cụ trục quan hoá.

Hiện thực:

- Viết chương trình
- Trục quan hoá và phân tích kết quả.

Đề 9: Viết chương trình Association Rules trên GPUs

Lý thuyết:

- Tìm hiểu về lập trình GPU (CUDA)
- Tìm hiểu về giải thuật cho Association Rules.

Hiện thực:

- Viết chương trình
- Đánh giá hiệu năng (speedup) với số lượng core khác nhau
- Nghiên cứu áp dụng giải thuật Reservoir Sampling để cải tiến khối lượng tính toán. Đánh giá thời gian xử lý và độ chính xác.

Bài 10: Viết chương trình so trùng ảnh dùng “The Skein Hash Function Family” trên Hadoop

Lý thuyết:

- Tìm hiểu Hadoop
- Tìm hiểu “The Skein Hash Function Family”
- Giải pháp loại các ảnh trùng nhau trong một tập ảnh lớn

Hiện thực:

- Viết chương trình
- Đánh giá hiệu suất trên hệ thống thực.

Đề 11: Đánh giá hiệu năng của BigDL (Distributed Deep Learning on Apache Spark) trong các bài toán phân tích dữ liệu lớn.

Lý thuyết:

- Tìm hiểu về BigDL
- Triển khai thực tế trên hệ thống máy tính mạnh SuperNode-XP.

Hiện thực:

- Chạy các ứng dụng để đánh giá hiệu năng BigDL trên SuperNode-XP
- Phân tích kết quả.

Ref:

- BigDL: <https://github.com/intel-analytics/BigDL>
- Spark: <https://spark.apache.org>
- Spark & Hadoop: <https://towardsdatascience.com/big-data-analytics-apache-spark-vs-apache-hadoop-7cb77a7a9424>

Đề 12: Đo Memory Bandwidth và ứng dụng vào bài toán nhân ma trận để rút ngắn thời gian chạy giải thuật hướng đến quan tâm đến memory bandwidth:

- Công cụ STREAM: <http://www.cs.virginia.edu/stream/>
- Benchmarking memory bandwidth:
<https://www.admin-magazine.com/HPC/Articles/Finding-Memory-Bottlenecks-with-Stream>
- Ref: (1)
https://people.eecs.berkeley.edu/~demmel/cs267_Spr16/Lectures/lecture02_memhimer_jwd16_4pp.pdf
<https://www.admin-magazine.com/HPC/Articles/Finding-Memory-Bottlenecks-with-Stream>

Đề mở rộng: Sinh viên có thể đề xuất bài toán của mình như:

- K-means hay SVM (Support Vector Machines)
- IoT caching
- Parallel photo mosaic algorithm
- Recommender systems (for student data)
- kBF - a probabilistic key value store:
<https://pdfs.semanticscholar.org/1c7d/1e83e6ba0c3d348cc1f074f65a9159be6c5f.pdf>
- Bài toán trên Graph...

Tham khảo

Video

High Performance Computing made easy, <http://www.hpc.uva.nl/>

Algorithms on Xeon Phi (XP)

1. Strassen algorithm: https://www.singularis-lab.com/docs/materials/07_Shapovalov_Strassen_CKA.pdf
2. Fast Smith-Waterman: <http://xsw.sdu-hpcl.org>
3. PCIT algorithm: <https://utexas.influent.utsystem.edu/en/publications/optimizing-the-pcit-algorithm-on-stampedes-xeon-and-xeon-phi-proc>
4. Breadth-first search: <http://www.dislab.org/docs/bfs-phi-paper-eng.pdf>
5. Graph coloring:
http://www.sandia.gov/~egboman/papers/Deveci_coloring_ipdps16.pdf
6. Pattern matching: <http://sbac.lip6.fr/2014/session%206/1-BitParallel.pdf>
7. Sort: <http://cass-mt.pnnl.gov/docs/ia3-2013/2-3.pdf>
8. OpenFoam:
<https://www.nersc.gov/assets/Uploads/IXPUGISC15OpenFOAMTCSV6.pdf>
9. Kalman Filter:
<https://facultystaff.richmond.edu/~ggilfoyl/research/keegan2014DNP.pdf>
10. SU2: http://stanford.edu/~economon/docs/SU2_IPCC_SciTech2015_final.pdf
11. Conjunction Gradient method:
https://en.wikipedia.org/wiki/Conjugate_gradient_method

Deep learning on XP

1. Deep Neural Networks for Financial Market Prediction:
https://www.researchgate.net/publication/281685181_Implementing_Deep_Neural_Networks_for_Financial_Market_Prediction_on_the_Intel_Xeon_Phi