

Biên soạn: TS. ĐỖ THANH NGHỊ (Chủ biên)
TS. NGUYỄN VĂN HÒA
TS. ĐỖ HIỆP THUẬN

GIÁO TRÌNH

LẬP TRÌNH SONG SONG



NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ
2014

**BIÊN MỤC TRÊN XUẤT BẢN THỰC HIỆN BỞI
TRUNG TÂM HỌC LIỆU TRƯỜNG ĐẠI HỌC CẦN THƠ**

Đỗ, Thanh Nghị

Giáo trình lập trình song song / Đỗ Thanh Nghị, Nguyễn Văn Hòa, Đỗ Hiệp Thuận .– Cần
Thơ : Nxb. Đại học Cần Thơ, 2014

148 tr. : minh họa ; 24 cm

Sách có danh mục tài liệu tham khảo

- | | |
|------------------------|-------------------------|
| 1. Computer algorithms | 2. Giải thuật song song |
| 2. Lập trình song song | |

I. Nhan đề

II. Nguyễn, Văn Hòa

III. Đỗ, Hiệp Thuận

005.1– DDC 22

MFN 187573

Ngh300

LỜI GIỚI THIỆU

Nhằm góp phần làm phong phú nguồn tư liệu phục vụ nghiên cứu, học tập cho sinh viên và bạn đọc Công nghệ thông tin, khoa Công nghệ thông tin và Truyền thông, Trường Đại học Cần Thơ, Nhà Xuất bản Đại học Cần Thơ ấn hành và giới thiệu cùng bạn đọc giáo trình “Lập trình song song” do Tiến sĩ Đỗ Thanh Nghị, Tiến sĩ Nguyễn Văn Hòa, Tiến sĩ Đỗ Hiệp Thuận biên soạn.

Giáo trình gồm 3 phần chính và 06 chương, giới thiệu về khái quát về tính toán song song, thiết kế giải thuật song song, lập trình song song với bộ nhớ chia sẻ và bộ nhớ phân tán. Thêm vào đó, cuối mỗi chương còn có nhiều tài liệu tham khảo hữu ích cho bạn đọc. Giáo trình là tài liệu học tập có giá trị cho sinh viên ngành Khoa học máy tính cũng như các bạn yêu thích muốn tìm tòi, học hỏi về lập trình song song.

Nhà Xuất bản Đại học Cần Thơ chân thành cảm ơn các tác giả và sự đóng góp ý kiến của quý thầy cô trong Hội đồng thẩm định trường Đại học Cần Thơ để giáo trình “Lập trình song song” được ra mắt bạn đọc.

Nhà Xuất bản Đại học Cần Thơ trân trọng giới thiệu đến sinh viên, giảng viên và bạn đọc giáo trình này.

Chân thành cảm ơn!

NHÀ XUẤT BẢN ĐẠI HỌC CẦN THƠ

LỜI NÓI ĐẦU

Đã trải qua nhiều thập kỷ, chúng ta chứng kiến sự phát triển của ngành khoa học máy tính. Xuất phát điểm từ công nghệ chế tạo phần cứng, lập trình cấp thấp, đến công nghệ thông tin, tổ chức, quản lý và xử lý hiệu quả hệ thống thông tin. Hiện nay, ngành khoa học máy tính đang bước vào thời kỳ công nghệ tri thức, ở mức trừu tượng cao hơn, làm máy tính trở nên thông minh hơn, để có thể giúp con người giải quyết nhiều vấn đề phức tạp trong thực tiễn. Những đòi hỏi này đã đặt ngành khoa học máy tính vào giải quyết các vấn đề lớn, phức tạp, đòi hỏi dung lượng bộ nhớ phải lớn, cần nhiều thời gian tính toán để hoàn thành. Một chương trình tuần tự trên một máy có bộ xử lý đơn, mất rất nhiều thời gian thậm chí là không thể hoàn thành việc xử lý các vấn đề phức tạp này. Trong khi, chương trình song song sử dụng đồng thời nhiều bộ xử lý để giải quyết một vấn đề, đã trở thành mô hình thống trị, cho phép tiết kiệm thời gian, chi phí, để xử lý các vấn đề lớn và phức tạp. Chính vì vậy, lập trình song song trở thành môn học chuyên ngành rất quan trọng của chương trình đào tạo cử nhân, kỹ sư và thạc sĩ công nghệ thông tin. Trong thời gian giảng dạy vừa qua, chúng tôi nhận thấy có quá ít tài liệu tham khảo trình bày một cách có hệ thống về lập trình song song. Chính vì lý do đó, chúng tôi đã thực hiện biên soạn quyển giáo trình “Lập trình song song” nhằm mục đích cung cấp thêm tài liệu tham khảo bằng tiếng Việt đến độc giả là sinh viên chuyên ngành công nghệ thông tin.

Giáo trình được soạn dựa trên kiến thức, kinh nghiệm học tập nghiên cứu, giảng dạy của chúng tôi trong suốt từ năm 2000 đến nay và các tài liệu tham khảo chính được trình bày trong mục tài liệu tham khảo của từng chương. Chúng tôi chỉ cung cấp các kiến thức cơ bản và các giải thích nhằm đơn giản việc trình bày các giải thuật học. Cuối mỗi chương có bài tập giúp độc giả ôn tập lại kiến thức của chương và trao dồi kỹ năng thực hành. Giáo trình được tổ chức thành sáu chương:

Chương 1 trình bày các kiến thức cơ bản của tính toán song song bao gồm các định nghĩa, các tài nguyên, hình thức tính toán song song, nhu cầu và các ứng dụng của tính toán song song trong thực tiễn.

Chương 2 giới thiệu kiến trúc máy tính, phân loại máy tính song song dựa trên kiến trúc bộ nhớ, các mô hình lập trình song song và các tiêu chí đánh giá hiệu năng của tính toán song song.

Chương 3 trình bày nguyên lý thiết kế giải thuật song song. Nội dung giới thiệu về mô hình tác vụ/kênh (task/channel) và tập trung trình bày mô hình 4 bước thiết kế giải thuật song song của Foster.

Chương 4 cung cấp các kỹ thuật lập trình song song trên kiến trúc đa nhân với bộ nhớ chia sẻ như: song song hóa tự động, lập trình song song với OpenMP và lập trình song song với Pthreads.

Chương 5 tập trung trình bày kỹ thuật lập trình song song với bộ nhớ phân tán sử dụng MPI (Message Passing Interface).

Chương 6 giới thiệu các công cụ để ghi nhận thông tin và phân tích hiệu suất của chương trình song song.

Nội dung giáo trình được giảng dạy cho sinh viên kỹ sư, thạc sĩ công nghệ thông tin trong thời lượng 30 tiết lý thuyết. Giảng viên có thể phân bổ thời gian dạy lý thuyết kết hợp với các ứng dụng minh họa của lập trình song song. Chương 1, 2 có thể được giảng dạy trong 5 tiết lý thuyết. Chương 3, 4, 5, 6 nên được giảng dạy trong 15 tiết lý thuyết và 20 tiết thực hành.

Trong thời gian biên soạn tài liệu, chúng tôi nhận được sự hỗ trợ quý báu về vật chất, tinh thần từ Khoa Công Nghệ Thông Tin và Truyền Thông, Trường Đại Học Cần Thơ, Khoa Kỹ thuật - Công nghệ - Môi trường, Đại học An Giang. Chúng tôi đặc biệt gửi lời cảm ơn đến Quý Thầy Cô thuộc Bộ Môn Khoa Học Máy Tính, Truyền Thông – Mạng Máy Tính, Tin Học Ứng Dụng đã nhiệt tình góp ý cho bản thảo.

Do thời gian có hạn và lần đầu tiên biên soạn, tài liệu không thể tránh khỏi những thiếu sót. Chúng tôi mong nhận được sự góp ý chân thành từ quý độc giả để quyển sách ngày được hoàn thiện hơn.

Cần thơ, tháng 11 năm 2013

NHÓM TÁC GIẢ

MỤC LỤC

PHẦN I. GIỚI THIỆU	1
Chương 1. GIỚI THIỆU	3
1.1 TÍNH TOÁN SONG SONG	3
1.1.1 Tính toán song song là gì?	3
1.1.2 Tại sao cần tính toán song song?	6
1.2 NHU CẦU TÍNH TOÁN SONG SONG	7
1.2.1 Tính toán song song cho khám phá tri thức và khai mô dữ liệu	7
1.2.2 Tính toán song song trong sinh tin học	8
1.2.3 Tính toán song song trong công nghiệp khai thác mỏ	9
1.2.4 Tính toán song song trong khoa học môi trường	11
1.3 LĨNH VỰC ỨNG DỤNG CỦA TÍNH TOÁN SONG SONG	11
CÂU HỎI ÔN TẬP VÀ BÀI TẬP	14
TÀI LIỆU THAM KHẢO	15
Chương 2. TÍNH TOÁN SONG SONG	16
2.1 GIỚI THIỆU KIẾN TRÚC MÁY TÍNH	16
2.1.1 Mô hình kiến trúc máy tính của von Neumann	16
2.1.2 Phân loại kiến trúc máy tính	17
2.2 CÁC LOẠI KIẾN TRÚC BỘ NHỚ CỦA MÁY TÍNH SONG SONG	18
2.2.1 Máy tính song song với bộ nhớ chia sẻ chung	18
2.2.2 Máy tính song song với bộ nhớ phân tán	20
2.2.3 Máy tính song song với bộ nhớ lai (hybrid memory)	21
2.3 MÔ HÌNH LẬP TRÌNH SONG SONG	22
2.3.1 Mô hình bộ nhớ chia sẻ dựa vào luồng	22
2.3.2 Mô hình bộ nhớ phân tán	24
2.4 ĐỒNG BỘ HÓA TRONG TÍNH TOÁN SONG SONG	25
2.4.1 Barriers	25
2.4.2 Semaphore	25
2.4.3 Truyền thông điệp	26
2.5 ĐÁNH GIÁ HIỆU NĂNG TÍNH TOÁN SONG SONG	26

2.5.1	Sự tăng tốc và hiệu suất	26
2.5.2	Luật Gene Amdahl	27
	CÂU HỎI ÔN TẬP VÀ BÀI TẬP	29
	TÀI LIỆU THAM KHẢO	30
	PHẦN II. LẬP TRÌNH SONG SONG	31
	Chương 3. THIẾT KẾ GIẢI THUẬT SONG SONG	33
3.1	GIỚI THIỆU VỀ THIẾT KẾ GIẢI THUẬT SONG SONG	33
3.1.1	Mô hình tác vụ/kênh (task/channel)	33
3.1.2	Mô hình thiết kế giải thuật song song của Foster	34
3.1.2.1	Phân hoạch	34
3.1.2.2	Giao tiếp	36
3.1.2.3	Tổng hợp	37
3.1.2.4	Ảnh xạ	39
3.2	TÍNH SỐ PI	40
3.2.1	Giải thuật tính số PI	41
3.2.2	Giải thuật song song tính số PI	42
3.3	SẮP XẾP CHẶN LỀ	45
3.3.1	Giải thuật sắp xếp chẵn lẻ	47
3.3.2	Giải thuật song song sắp xếp chẵn lẻ	47
3.4	NHÂN MA TRẬN	50
3.4.1	Giải thuật nhân ma trận	51
3.4.2	Giải thuật song song nhân ma trận	51
	CÂU HỎI ÔN TẬP VÀ BÀI TẬP	57
	TÀI LIỆU THAM KHẢO	58
	Chương 4. LẬP TRÌNH SONG SONG VỚI BỘ NHỚ CHIA SẺ	59
4.1	SONG SONG HÓA TỰ ĐỘNG (AUTOMATIC PARALLELIZATION)	59
4.1.1	Song song tự động với GCC	60
4.2	LẬP TRÌNH SONG SONG VỚI OPENMP	61
4.2.1	Giới thiệu về OpenMP	61
4.2.2	Chương trình đơn giản Hello OpenMP	63
4.2.3	Phạm vi của biến (Scope of variable)	65
4.2.4	Các chỉ thị trong OpenMP	67
4.3	LẬP TRÌNH SONG SONG VỚI PTHREADS	73
4.3.1	Giới thiệu Pthread (POSIX thread)	73
4.3.2	Chương trình Pthreads “Hello World”	75

4.3.3 Quản lý các luồng	76
4.3.4 Bài toán nhân ma trận với vector	80
4.3.5 Biến mutex	81
CÂU HỎI ÔN TẬP VÀ BÀI TẬP	85
TÀI LIỆU THAM KHẢO	86
Chương 5. LẬP TRÌNH SONG SONG VỚI BỘ NHỚ PHÂN TÁN	87
5.1 GIỚI THIỆU MPI	88
5.1.1 MPI là gì?	88
5.1.2 Các khái niệm trong MPI	89
5.2 LẬP TRÌNH SONG SONG VỚI MPI	90
5.2.1 Cấu trúc chương trình MPI	90
5.2.2 Sáu hàm cơ bản MPI	90
5.2.3 Nhóm hàm cho giao tiếp với nhóm	96
5.3 CÀI ĐẶT GIẢI THUẬT SONG SONG VỚI MPI	102
5.3.1 Cài đặt giải thuật tính số PI với MPI	102
5.3.2 Cài đặt giải thuật sắp xếp chẵn lẻ với MPI	104
CÂU HỎI ÔN TẬP VÀ BÀI TẬP	108
TÀI LIỆU THAM KHẢO	110
PHẦN III. CÔNG CỤ PHÂN TÍCH VÀ ĐÁNH GIÁ HIỆU QUẢ	
CHƯƠNG TRÌNH SONG SONG	111
Chương 6. CÔNG CỤ PHÂN TÍCH VÀ ĐÁNH GIÁ HIỆU NĂNG	
CHƯƠNG TRÌNH SONG SONG	113
6.1 CÔNG CỤ TÍNH THỜI GIAN	113
6.1.1 Lệnh time	113
6.1.2 Hàm MPI_Wtime	114
6.1.3 Hàm gettimeofday	115
6.2 CÔNG CỤ KIỂM TRA BỘ NHỚ	116
6.3 CÁC CÔNG CỤ PROFILING	117
6.3.1 Công cụ gprof	118
6.3.2 ompP	121
6.3.3 mpiP	122
6.4 CÔNG CỤ PHÂN TÍCH HIỆU NĂNG TAU	124
CÂU HỎI ÔN TẬP VÀ BÀI TẬP	129
TÀI LIỆU THAM KHẢO	130

DANH MỤC HÌNH

Hình 1.1	Tính toán tuần tự	4
Hình 1.2	Tính toán song song	4
Hình 1.3	Các lĩnh vực ứng dụng tính toán song song	12
Hình 1.4	Các chuyên ngành ứng dụng tính toán song song	13
Hình 2.1	Mô hình kiến trúc máy tính von Neumann	16
Hình 2.2	Phân loại kiến trúc máy tính [1]	17
Hình 2.3	Kiến trúc chia sẻ bộ nhớ (UMA)	19
Hình 2.4	CC-UMA [4]	19
Hình 2.5	Kiến trúc chia sẻ bộ nhớ - NUMA	20
Hình 2.6	Bộ nhớ phân tán	21
Hình 2.7	Bộ nhớ lai	22
Hình 2.8	Mô hình luồng [2]	23
Hình 2.9	Mô hình truyền thông điệp [4]	24
Hình 2.10	Định luật Amdahl về khả năng tăng tốc khi song song hóa	28
Hình 3.1	Mô hình tác vụ/kênh	34
Hình 3.2	Mô hình thiết kế giải thuật song song của Foster	35
Hình 3.3	Tính số PI với phương pháp tích phân	40
Hình 3.4	Tính số PI với phương pháp rời rạc	41
Hình 3.5	Phân bổ (đều) tính toán song song số PI trên 5 bộ xử lý	43
Hình 3.6	Phân bổ (xen kẽ) tính toán song song số PI trên 5 bộ xử lý	44
Hình 3.7	Sắp xếp với giải thuật nổi bọt	46
Hình 3.8	Sắp xếp với giải thuật chẵn lẻ	47
Hình 3.9	Giải thuật song song chẵn lẻ (số lượng bộ xử lý bằng số phần tử)	49
Hình 3.10	Giải thuật song song chẵn lẻ (số lượng bộ xử lý nhỏ hơn số phần tử)	49
Hình 3.11	Nhân 2 ma trận	50
Hình 3.12	Phân hoạch vấn đề nhân 2 ma trận thành các tác vụ chi tiết	52
Hình 3.13	Các chiến lược tổng hợp	52
Hình 3.14	Tổng hợp của các ánh xạ khối 1 chiều, 2 chiều	53
Hình 3.15	Giải thuật Cannon cho nhân 2 ma trận	56
Hình 4.1	Đa luồng với fork-join của OpenMP	62

Hình 4.2	So sánh tiến trình và luồng [2]	74
Hình 4.3	Quá trình tạo luồng	78
Hình 4.4	Nối luồng giữa hai luồng master và worker	79
Hình 4.5	Ma trận nhân với vector	80
Hình 5.1	Mô hình truyền thông điệp	87
Hình 5.2	Truyền thông điệp với hàm MPI_Bcast	97
Hình 5.3	Truyền thông điệp với hàm MPI_Gather	97
Hình 5.4	Truyền thông điệp với hàm MPI_Allgather	98
Hình 5.5	Truyền thông điệp với hàm MPI_Scatter	98
Hình 5.6	Truyền thông điệp với hàm MPI_Alltoall	99
Hình 5.7	Hàm tổng hợp dữ liệu MPI_Reduce với phép MPI_SUM và root=0	100
Hình 5.8	Hàm tổng hợp dữ liệu MPI_Allreduce với phép MPI_SUM	101
Hình 6.1	Giao diện chính của công cụ paraprof	126
Hình 6.2	Dữ liệu chính của profiling	126
Hình 6.3	Thông tin về thời gian của các hàm trên 1 node	127
Hình 6.4	Giao diện chọn tập tin slog2 của JumShot	128
Hình 6.5	Danh sách các hàm được thực thi trên node0	128
Hình 6.6	Tracing của các hàm chạy trên node0	129

DANH MỤC BẢNG

Bảng 2.1	Sự tăng tốc và hiệu suất của chương trình song song	27
Bảng 3.1	Mã giả của giải thuật tính số PI	42
Bảng 3.2	Mã giả tính toán độc lập, song song trên từng bộ xử lý	45
Bảng 3.3	Giải thuật tuần tự nổi bọt	46
Bảng 3.4	Giải thuật tuần tự chẵn lẻ	48
Bảng 3.5	Giải thuật song song chẵn lẻ	50
Bảng 3.6	Giải thuật nhân 2 ma trận	51
Bảng 3.7	Giải thuật song song nhân 2 ma trận (ánh xạ khối 2 chiều)	55
Bảng 4.1	Chương trình với dữ liệu độc lập trong vòng lặp inner-loop	61
Bảng 4.2	Chương trình với dữ liệu phụ thuộc trong vòng lặp inner-loop	61
Bảng 4.3	Chương trình omp_hello.c	64
Bảng 4.4	Chương trình tính số PI tuần tự (pi.c)	66
Bảng 4.5	Chương trình tính số PI song song với 4 luồng (opm_pi.c)	67
Bảng 4.6	Đoạn code minh họa chỉ thị song song vùng lệnh	69
Bảng 4.7	Chương trình tính số PI song song với lệnh lặp for (opm_pi.c)	71
Bảng 4.8	Ví dụ chỉ thị SECTIONS để thực hiện song song 2 lệnh lặp for	72
Bảng 4.9	Chương trình Hello World với Pthreads (pt_hello.c)	75
Bảng 4.10	Nối luồng để đồng bộ hóa	79
Bảng 4.11	Giải thuật nhân ma trận với vector	80
Bảng 4.12	Hàm start_routine nhân ma trận vector	81
Bảng 4.13	Minh họa cách sử dụng biến mutex trong một chương trình đa luồng	82
Bảng 5.1	Kiểu dữ liệu cơ bản	89
Bảng 5.2	Cấu trúc chương trình MPI trong ngôn ngữ C	90
Bảng 5.3	Chương trình minh họa HelloWorld.c	92
Bảng 5.4	Chương trình minh họa sử dụng hàm MPI_Send	94
Bảng 5.5	Chương trình minh họa sử dụng hàm MPI_Recv	95
Bảng 5.6	Chương trình minh họa sử dụng hàm MPI_Bcast	97
Bảng 5.7	Phép toán tổng hợp MPI_Op	100
Bảng 5.8	Chương trình minh họa sử dụng hàm MPI_Reduce	101
Bảng 5.9	Chương trình minh họa tính số PI (pi.c)	102

Bảng 5.10	Chương trình minh họa sắp xếp chẵn lẻ (even-odd-sort.c)	105
Bảng 6.1	Chương trình mpi_wtime.c	115
Bảng 6.2	Chương trình minh họa hàm gettimeofday()	115
Bảng 6.3	mypro.c lỗi truy cập và không giải phóng vùng nhớ	117
Bảng 6.4	mpip_test.c	123

THUẬT NGỮ

Agglomeration	tổng hợp
Balancing	cân bằng tải
Block	khối
Channel	kênh
Cloud	đám mây
Collective communication	giao tiếp nhóm
Communication	giao tiếp
Distributed memory	bộ nhớ phân tán
Granularity	mức độ chi tiết
Grid	lưới
High performance computing	tính toán hiệu năng cao
Hyperthreading	siêu phân luồng
I/O	vào/ra
Inner product	tích vô hướng (tích trong)
Mapping	ánh xạ
MIMD (Multiple-Instruction- Multiple -Data)	đa lệnh, đa dữ liệu
MISD (Multiple-Instruction-Single-Data)	đa lệnh, đơn dữ liệu
Mode	chế độ
MPI (The Message Passing Interface)	truyền thông điệp
MPMD (Multiple Program Multiple Data)	đa dữ liệu, đa chương trình
Multi-core	đa nhân
OpenMP (Open Multiprocessing)	mô hình đa xử lý
Outer product	tích có hướng (tích ngoài)
Parallel computing	tính toán song song
Parallel programming	lập trình song song
Partition	phân hoạch
PC cluster	nhóm máy tính
Point-to-point	điểm nối điểm
Process	tiến trình
Profiling	tập tin log khi chạy chương trình

Shared memory	bộ nhớ chia sẻ
SIMD (Single-Instruction-Multiple-Data)	đơn lệnh, đa dữ liệu
SISD (Single-Instruction-Single-Data)	đơn lệnh, đơn dữ liệu
SPMD (Single Program Multiple Data)	đa dữ liệu, đơn chương trình
Synchronization	đồng bộ hóa
Task	tác vụ
Thread	luồng
Type	kiểu

PHẦN I

GIỚI THIỆU

- TÍNH TOÁN SONG SONG
- KIẾN TRÚC TÍNH TOÁN SONG SONG
- MÔ HÌNH TÍNH TOÁN SONG SONG
- ĐỒNG BỘ HÓA
- PHÂN TÍCH HIỆU SUẤT

Chương 1

GIỚI THIỆU

Trong những năm qua, tính toán song song được sử dụng phổ biến để giải quyết các vấn đề lớn và phức tạp. Để làm quen với tính toán song song, chúng tôi trước tiên giới thiệu tổng quát các khái niệm cơ bản về tính toán song song. Chúng tôi tập trung giới thiệu ngắn gọn về định nghĩa, các tài nguyên, hình thức tính toán song song, nhu cầu và các ứng dụng của tính toán song song trong thực tiễn.

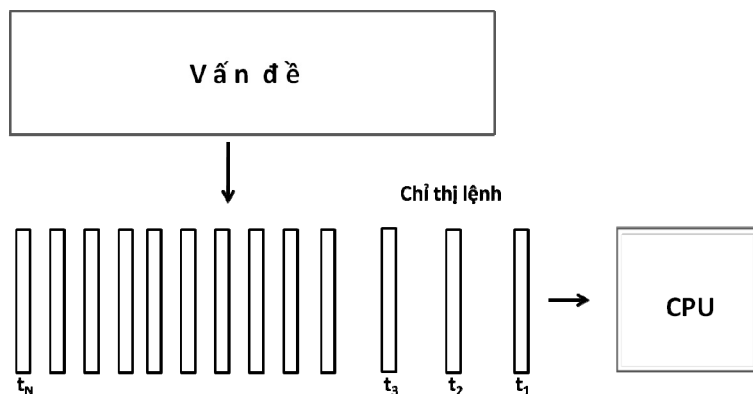
1.1 TÍNH TOÁN SONG SONG

Ngày nay, ngành khoa học máy tính giải quyết các vấn đề lớn, phức tạp, đòi hỏi dung lượng bộ nhớ phải lớn, cần nhiều thời gian tính toán để hoàn thành. Một chương trình tuần tự trên một máy có bộ xử lý đơn, không khả thi cho xử lý các vấn đề này. Trong khi đó, chương trình song song sử dụng đồng thời nhiều bộ xử lý để giải quyết một vấn đề, đã trở thành mô hình thông trị, cho phép tiết kiệm thời gian, chi phí, khi xử lý các vấn đề lớn và phức tạp.

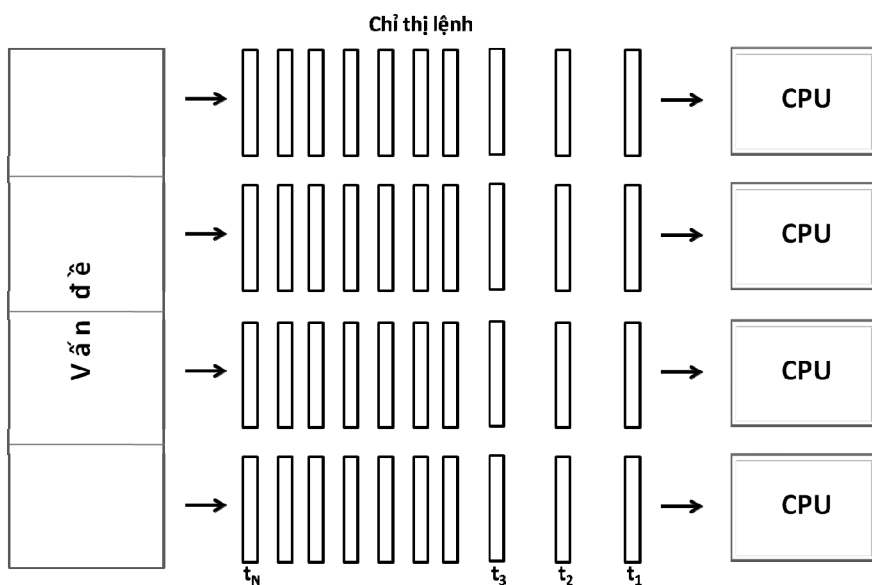
1.1.1 Tính toán song song là gì?

Tính toán song song là một hình thức tính toán trong đó nhiều phép tính được thực hiện đồng thời, hoạt động trên nguyên tắc là những vấn đề lớn đều có thể chia thành nhiều phần nhỏ hơn, tương đối độc lập với nhau, sau đó được giải quyết đồng thời (song song) trên máy tính có nhiều bộ xử lý hoặc trên nhiều máy tính.

Theo truyền thống, phần mềm máy tính được viết cho tính toán tuần tự. Để giải quyết một vấn đề, một giải thuật được xây dựng và thực thi tuần tự theo các dòng lệnh. Những câu lệnh này lại được thực thi trên một bộ xử lý trung tâm (CPU - Central Processing Unit) của một máy tính. Vấn đề được phân tách thành dãy các chỉ thị lệnh rời rạc. Theo trình tự, chỉ có một chỉ thị lệnh được thực thi trong một khoảng thời gian, sau khi chỉ thị lệnh này kết thúc, thì chỉ thị lệnh tiếp theo mới được thực hiện, cứ thế tiếp diễn đến khi chỉ thị lệnh cuối cùng kết thúc. Khi đó vấn đề được giải quyết xong. Hình 1.1 minh họa tính toán tuần tự.



Hình 1.1 Tính toán tuần tự



Hình 1.2 Tính toán song song

Tính toán song song sử dụng đồng thời các tài nguyên tính toán để giải quyết một vấn đề. Việc này được hoàn thành bằng cách tách vấn đề lớn cần giải quyết thành nhiều vấn đề nhỏ độc lập, sau đó mỗi vấn đề nhỏ được tách ra thành dãy các chỉ thị lệnh rời rạc, cuối cùng các dãy chỉ thị lệnh của từng vấn đề nhỏ được thực thi đồng thời trên những bộ xử lý khác nhau. Hình 1.2 minh họa tính toán song song.

Tài nguyên tính toán được dùng trong tính toán song song có thể là các dạng khác nhau như:

1. một máy tính đơn có nhiều bộ xử lý,
2. một siêu máy tính có nhiều bộ đa xử lý,
3. nhiều máy tính được kết nối mạng,
4. phần cứng chuyên biệt, chẳng hạn như các bộ đồng xử lý Field-Programmable Gate Array (FPGA), bộ xử lý đa năng của card đồ họa General-Purpose Computing on Graphics Processing Units (GPGPU) ,
5. hoặc có thể là sự kết hợp của những dạng trên.

Tính toán song song giải quyết một vấn đề lớn, phức tạp có thể là một trong các dạng sau đây:

1. phân hoạch vấn đề lớn thành các vấn đề nhỏ độc lập để xử lý một cách đồng thời các vấn đề nhỏ,
2. thực thi nhiều chỉ thị lệnh ở bất kỳ thời điểm,
3. xử lý vấn đề trên nhiều tài nguyên tính toán nhanh hơn so với chỉ sử dụng tài nguyên tính toán đơn.

Tính toán song song có thể thực hiện dưới nhiều hình thức khác nhau như: song song cấp bit, song song cấp lệnh, song song dữ liệu, và song song tác vụ.

Giải thuật song song thường rất khó thiết kế hơn so với những giải thuật tuần tự. Trong thực tế, không dễ dàng phân hoạch hiệu quả vấn đề lớn thành các vấn đề nhỏ độc lập để xử lý một cách đồng thời. Hơn nữa, thực thi song song các vấn đề nhỏ phát sinh sự tương tranh trong truy xuất dữ liệu, sự phụ thuộc vào dữ liệu và xử lý, đồng bộ hóa xử lý, tạo ra nhiều lỗi tiềm ẩn.

Trong trường hợp lý tưởng, khả năng tăng tốc từ tính toán song song là tuyến tính so với số lượng thành phần xử lý. Tuy nhiên, rất ít khi giải thuật song song có thể đạt được sự tăng tốc tối ưu này. Hầu hết các giải thuật song song chỉ đạt đến gần mức tuyến tính khi số thành phần xử lý là nhỏ, và tiệm cận đến một giá trị không đổi khi số lượng các thành phần xử lý tăng lên.

Khả năng tăng tốc cao nhất có thể đạt được của một chương trình khi được song song hóa trên nền tảng tính toán song song, tuân theo định luật Gene Amdahl đề xuất trong những năm 1960. Nếu tồn tại một bộ phận nhỏ của giải thuật không thể được song song (thực thi tuần tự) thì kéo theo hạn chế khả năng tăng tốc tổng thể của việc song song hóa giải thuật. Chúng tôi sẽ trình bày chi tiết về định luật Gene Amdahl trong chương tiếp theo.

1.1.2 Tại sao cần tính toán song song?

Tính toán song song cho phép tiết kiệm thời gian và chi phí. Theo suy diễn một cách lý tưởng, nếu sử dụng càng nhiều tài nguyên tính toán cho một công việc thì sẽ càng rút ngắn thời gian hoàn thành, có thể tiết kiệm chi phí về thời gian. Máy tính song song có thể được xây dựng từ các thành phần giá rẻ.

Tính toán song song cho phép giải quyết các vấn đề lớn và phức tạp. Trong thực tiễn, có nhiều vấn đề rất lớn và phức tạp mà chúng ta không thể xử lý chúng trên một máy tính đơn, đặc biệt là hạn chế về bộ nhớ của máy tính. Chẳng hạn như dự báo thời tiết, thăm dò khai thác mỏ, mô phỏng hệ phức hợp, công cụ tìm kiếm Web, đòi hỏi khoảng 10^{15} FLOPS (Floating-point Operations Per Second) và 10^{15} Bytes tài nguyên của máy tính.

Một tài nguyên tính toán đơn chỉ có thể làm một việc tại một thời điểm. Nhiều tài nguyên tính toán có thể được làm nhiều thứ cùng một lúc, có khả năng cung cấp đồng thời. Ví dụ, như Grid Access (www.accessgrid.org) cung cấp một mạng lưới cộng tác toàn cầu, mà ở đó mọi người từ khắp nơi trên thế giới có thể gặp gỡ và tiến hành làm việc trong môi trường "ảo".

Tính toán có thể được thực hiện từ các nguồn tài nguyên phi cục bộ, chẳng hạn như sử dụng tài nguyên tính toán trên một mạng diện rộng, thậm chí Internet, trong khi các tài nguyên tính toán cục bộ bị giới hạn. Ví dụ: SETI@home (setiathome.berkeley.edu) có hơn 1,3 triệu người dùng và 3,2 triệu máy tính, gần như trên tất cả các quốc gia trên thế giới (tính đến tháng 7 năm 2012). Hay Folding@home (folding.stanford.edu) có hơn 450 000 CPU trên toàn thế giới (tính đến tháng 7 năm 2011).

Để cải thiện thời gian xử lý của tính toán tuần tự, chúng ta cần phải phát triển những máy tính tuần tự ngày càng nhanh hơn. Tuy nhiên, điều này tồn tại nhiều hạn chế về mặt vật lý và thực tiễn đặt ra. Trước tiên là tốc độ truyền dẫn của một máy tính tuần tự phụ thuộc trực tiếp vào tốc độ di chuyển dữ liệu giữa các phần cứng. Giới hạn tuyệt đối là tốc độ ánh sáng (30 cm/nano giây) và giới hạn việc truyền tải của dây đồng (9 cm/nano giây). Tăng tốc độ đòi hỏi phải ngày càng thu hẹp khoảng cách giữa thành phần xử lý nguyên tố (processing element). Ngoài ra, việc này còn gặp giới hạn về kích thước thu nhỏ. Công nghệ vi xử lý cho phép tích hợp số lượng ngày càng tăng của các đèn bán dẫn (transistor) được đặt trên một con chip. Tuy nhiên, ngay cả với các thành phần phân tử hoặc nguyên tử, vẫn có một giới hạn về kích thước thu nhỏ sẽ đạt được của các bộ xử lý. Phát triển cải thiện tốc độ của bộ xử lý đơn thì đắt tiền hơn so với sử dụng một số lượng lớn các bộ xử lý thông thường, để đạt được cùng một hiệu suất giải quyết vấn đề. Kiến trúc máy tính hiện nay đang ngày càng phát triển, dựa vào phần cứng song song, để cải thiện hiệu suất: nhiều đơn vị thực thi (multiple execution units), chỉ thị lệnh dạng ống dẫn (pipelined instructions), đa nhân (multi-core).

1.2 NHU CẦU TÍNH TOÁN SONG SONG

Trong thực tiễn, nhiều chuyên ngành phải giải quyết các bài toán rất lớn, phức tạp, cần nhiều tài nguyên tính toán và thời gian để hoàn thành. Tính toán song song trở thành mô hình thống trị trong các chuyên ngành này.

1.2.1 Tính toán song song cho khám phá tri thức và khai mỏ dữ liệu

Trong những năm 1990, cuộc cách mạng kỹ thuật số cho phép số hóa thông tin dễ dàng và chi phí thấp, thêm vào đó là sự phát triển của công nghệ thông tin bao gồm cả phần cứng lẫn phần mềm, công nghệ truyền thông, web, internet đã góp phần đưa máy tính vào các sinh hoạt thường nhật của con người. Hệ quả là một khối lượng lớn dữ liệu được sinh ra và lưu trữ trong các cơ sở dữ liệu. Từ năm 1999, Giáo sư P. Lyman và các cộng sự của ông ở Đại học Berkeley đã tiến hành thống kê dữ liệu được sinh ra hằng năm trên toàn cầu (tham khảo ở địa chỉ <http://www.sims-berkeley.edu/research/projects/how-much-info-2003>), dữ liệu toàn cầu tăng 5 Exabytes (5×10^{18} bytes). Vấn đề đặt ra là làm sao chúng ta có thể rút trích tri thức quan trọng từ các kho dữ liệu khổng lồ. Các tri thức phục vụ cho các tổ chức, cơ quan, công ty bao gồm việc phát hiện quan trọng trong khoa học, các dự báo chính xác về thời tiết và các thảm họa tự nhiên, những tri thức cho phép ta xác định được nguyên nhân và phương pháp điều trị các căn bệnh hiểm nghèo, dự báo dịch hại, etc.

Sự ra đời của công nghệ khám phá tri thức và khai mỏ dữ liệu [6] trong những năm gần đây nhằm đáp ứng nhu cầu cần thiết của các tổ chức, cơ quan, công ty về phát hiện tri thức từ các kho dữ liệu khổng lồ. Các ứng dụng thành công của công nghệ khai mỏ dữ liệu có thể tìm thấy trong rất nhiều lĩnh vực như: tiếp thị, ngân hàng, bảo hiểm, y tế, sinh học, môi trường, phát hiện gian lận, tìm kiếm thông tin, lọc thư rác, phân loại văn bản.

Xét ứng dụng về tìm kiếm hay phân lớp ảnh được mô tả trong [5]. Tập ảnh với khoảng 14 triệu ảnh, 20 ngàn lớp đối tượng (người, xe đạp, xe ô tô, cây, etc). Vấn đề là chúng ta làm sao xây dựng mô hình phân lớp đối tượng (gán nhãn) tự động cho từng ảnh với lỗi thấp nhất. Do dữ liệu ảnh là dữ liệu phi cấu trúc, công việc đầu tiên cần giải quyết là trích chọn từ ảnh các đặc trưng (tổ chức đồ, biểu diễn đặc trưng không đổi, etc) và tìm cách biểu diễn ảnh (thường là dạng véc tơ). Các đặc trưng này cho phép phân biệt tốt nhất giữa các lớp đối tượng. Tuy nhiên không có bất kỳ một đặc trưng nào là tốt cho tất cả các trường hợp. Chính vì lý do đó, người ta thường kết hợp nhiều đặc trưng để có kết quả xử lý tốt nhất. Thời gian cho rút trích từng đặc trưng hơn một tháng trên một máy tính cá nhân, sử dụng hàng trăm GB bộ nhớ. Do đó cần phải phát triển các giải thuật cho phép thực hiện tính toán song song trên máy tính nhiều bộ xử lý hay trên nhiều máy tính để tăng tốc quá trình xử lý này. Hơn nữa, các đặc trưng được rút trích một cách độc lập, điều này cho phép dễ dàng thực hiện song song. Sau bước tiền xử lý trên, tập ảnh được biểu diễn dưới

dạng một bảng có 14 triệu dòng và 100 ngàn cột với 20 ngàn lớp đối tượng khác nhau. Bước tiếp theo là xây dựng mô hình học phân lớp 20 ngàn đối tượng từ bảng dữ liệu vừa thu được ở bước tiền xử lý. Mô hình học phân lớp và nhanh nhất hiện nay cho vấn đề này là máy học véc tơ hỗ trợ (SVM [7]). Tuy nhiên, để huấn luyện một mô hình phân lớp SVM, chúng ta mất hơn một năm thực hiện trên một máy tính có bộ xử lý đơn. Do đó, người ta thường tập trung phát triển các giải thuật học song song để xử lý đồng thời các vấn đề nhỏ (được phân rã một cách tương đối độc lập từ vấn đề lớn), trên máy tính có nhiều bộ xử lý hay nhóm các máy tính.

Xử lý song song là nhu cầu cần thiết, nhằm đáp ứng ràng buộc về thời gian của các ứng dụng về phát hiện tri thức từ các kho dữ liệu khổng lồ.

1.2.2 Tính toán song song trong sinh tin học

Tin sinh học (bioinformatics [8], [9]) là một lĩnh vực khoa học sử dụng các công nghệ của các ngành tin học, toán học ứng dụng, thống kê, khoa học máy tính, trí tuệ nhân tạo, hóa học và hóa sinh (biochemistry) để giải quyết các vấn đề sinh học. Tin sinh học phát triển các giải thuật, lý thuyết và các kỹ thuật thống kê và tính toán để giải quyết các bài toán bắt nguồn từ nhu cầu quản lý và phân tích dữ liệu sinh học. Trong khi đó, sinh học tính toán (computational biology) thiên về kiểm định các giả thuyết (hypothesis) được đặt ra của một vấn đề trong sinh học nhờ máy tính thực nghiệm trên dữ liệu mô phỏng, với mục đích chính là phát hiện tri thức về sinh học, chẳng hạn như dự đoán mối quan hệ tương tác giữa các protein, dự đoán cấu trúc bậc 2 phân tử của protein, etc. Các nghiên cứu trong ngành sinh học tính toán bao gồm:

- bắt cặp trình tự (sequence alignment),
- bắt cặp cấu trúc protein (protein structural alignment),
- dự đoán cấu trúc protein (protein structure prediction),
- dự đoán biểu hiện gen (gene expression),
- tương tác protein - protein (protein-protein interactions),
- và mô hình hóa quá trình tiến hoá.

Những mối quan tâm chính trong các dự án tin sinh học và sinh học tính toán là việc sử dụng các công cụ toán học để trích rút các thông tin hữu ích từ các kho dữ liệu rất lớn được thu nhận từ các kỹ thuật sinh học.

Những bài toán đặc trưng trong sinh học tính toán bao gồm việc lắp ráp (assembly) những trình tự DNA chất lượng cao từ các đoạn ngắn DNA thu nhận được từ kỹ thuật xác định trình tự DNA (shotgun sequencing), và việc dự đoán qui luật gen (gene regulation) với dữ liệu từ các mRNA, microarray hay khối phổ (mass spectrometry).