

LẬP LỊCH LÀM VIỆC CHO CÁC TÁC VỤ KHAI PHÁ DỮ LIỆU TRONG MÔI TRƯỜNG LƯỚI DỮ LIỆU

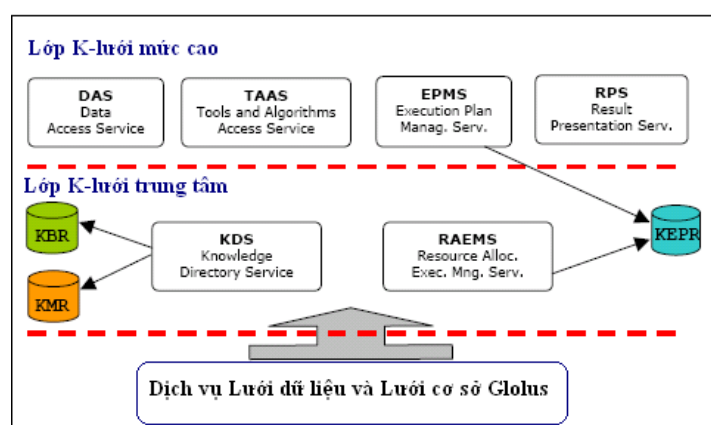
Đoàn Văn Ban (Viện Công nghệ thông tin - Viện KH&CN Việt Nam)
Vũ Đức Quảng (Trường ĐH Quảng Nam)

1. Giới thiệu

Trong những năm gần đây, chúng ta đã chứng kiến sự bùng nổ cả về số lượng lẫn kích thước của các kho dữ liệu điện tử. Điều này đem đến cho các nhà nghiên cứu cơ hội để phát triển hiệu quả các kỹ thuật khai phá để khám phá và trích rút tri thức từ khối lượng thông tin khổng lồ đó. Hơn nữa, do kích thước của dữ liệu lớn và thường được phân tán ngẫu nhiên. Nếu như chúng ta xem các thuật toán khai phá dữ liệu là thường xuyên được thực hiện, chúng ta có thể kết luận rằng lưới là nền tảng cơ sở cho việc kiểm khai một dịch vụ hiệu năng cao cho quá trình khai phá tri thức phân tán (DKD-Distributed Knowledge Discovery).

Môi trường lưới có thể cung cấp khả năng phân bố tài nguyên, khả năng xử lý cộng tác và khả năng phân tích khai phá dữ liệu với khối lượng lớn dữ liệu được đưa ra và được lưu trữ. Vì các ứng dụng DKD đòi hỏi dữ liệu đặt trung, một trong những yêu cầu của môi trường lưới DKD là quản lý việc lưu trữ và việc truyền tải tài nguyên một cách hiệu quả.

Trong báo cáo này, kiến trúc quản lý dữ liệu dựa trên các hệ thống lưu trữ và các dịch vụ quản lý siêu dữ liệu. Dịch vụ lưới dữ liệu được xây dựng bên trên các dịch vụ Globus cơ sở [8] và đơn giản hóa việc quản lý các tính toán truy cập các nguồn dữ liệu lớn và phân tán. Số các thành phần tính toán được xem xét với số lượng hữu hạn và các tính năng của chúng đã được biết. Lưới dữ liệu cung cấp bộ quản lý khối lượng công việc cần làm (WorkLoad manager) có nhiệm vụ định nghĩa các công việc với các yêu cầu liên quan và lập lịch làm việc cho chúng trong môi trường Lưới. Mô hình lưới dữ liệu trình bày ở trên đưa ra các yêu cầu đối với việc thực thi của một lưới DKD, ở đây, dữ liệu đòi hỏi có thể được lấy từ nhiều nguồn. Ngoài ra, các dịch vụ cơ sở của lưới dữ liệu có thể được sử dụng và được mở rộng để thực thi các dịch vụ lưới mức cao hơn liên quan đến quá trình khai phá tri thức từ các kho dữ liệu phân tán. Hạ tầng cơ sở lưới chuyên dụng như thế được gọi là *lưới tri thức* [8], kiến trúc này được thiết kế để phù hợp với các kỹ thuật lưới ở mức thấp hơn và với các kỹ thuật lưới dữ liệu. Kiến trúc lưới tri thức có thể được chia thành 2 lớp: Lớp K-Lưới trung tâm và Lớp K-Lưới mức cao.



Hình 1. Kiến trúc lưới tri thức

Trên cơ sở kiến trúc Lưới tri thức, các dịch vụ trọng tâm của nó, ví dụ: Dịch vụ thư mục tri thức (KDS) và dịch vụ phân phối tài nguyên và quản lý thực thi (RAEM - Resource Allocation and Execution Management). KDS mở rộng dịch vụ kiểm soát và khai phá thông tin (MDS-Monitoring and Discovery Services) Globus[7] và có nhiệm vụ duy trì một mô tả của tất cả dữ liệu và các công cụ được sử dụng trong Lưới tri thức. Siêu dữ liệu được quản lý bởi KDS được trình bày trong các tài liệu XML, được lưu trữ trong kho siêu dữ liệu tri thức (KMR-Knowledge Metadata Repository).

Báo cáo này phân tích sâu một số vấn đề đã gặp trong thiết kế và thi hành một chiến lược lập lịch cho bộ định giá của kiến trúc lưới tri thức. Với cách giải quyết này, bộ lập lịch cần sử dụng một mô hình thực thi phức hợp để xem xét trạng thái hiện tại của lưới, xác định vị trí các nguồn dữ liệu và cách thức thực thi tác vụ. Thông tin về các chi phí thực thi là cần thiết đối với bộ định giá để có thể đánh giá khả năng của các hoạt động quản lý tập dữ liệu (ví dụ: việc di chuyển hay phân hoạch một tập dữ liệu) và để định cấu hình thời gian tải các công cụ DM nhằm đạt được hiệu suất mong muốn (ví dụ: bằng việc cho một phép phân tích có chi phí cao thực hiện song song). Tuy nhiên, chi phí thực thi của các công cụ DM không chỉ phụ thuộc vào kích thước dữ liệu mà còn phụ thuộc vào các tham số khai phá được xác định bởi người dùng.

Xét ví dụ về khai phá luật kết hợp (ARM-Association Rule Mining): Độ phức tạp của ARM không chỉ phụ thuộc vào kích thước của dữ liệu đầu vào mà còn phụ thuộc vào *ngưỡng hỗ trợ* và *ngưỡng tin cậy*. Hơn nữa, sự tương quan của các *mục* (item) xuất hiện trong tập dữ liệu ảnh hưởng lớn đến số lượng và độ dài tối đa của các luật được tìm thấy bởi công cụ ARM. Bởi thế, thật khó để dự đoán việc cải tiến thực thi dựa vào chi phí vào/ra và kích thước dữ liệu đầu vào. Để giải quyết các vấn đề này, ta đưa vào trong dịch vụ KDS gồm cả thông tin động liên quan các thực thi khác nhau của các tác vụ DM trên các nguồn dữ liệu khác nhau. Thông tin này có thể được thêm vào giống như siêu dữ liệu bổ sung được kết hợp với tập dữ liệu. Vì thế, KDS được mở rộng không chỉ có siêu dữ liệu tĩnh được sử dụng để xem xét dữ liệu hay các công cụ được sử dụng trong lưới tri thức mà thông tin động còn được sử dụng để xác định làm thế nào để cấu hình và chạy các công cụ đó. Siêu dữ liệu động tập trung vào thông tin kiểm tra việc chạy các phần mềm khác nhau trước đây trên các tập dữ liệu xác định. Siêu dữ liệu động có thể được kết hợp với các tập dữ liệu để biết thông tin về các chi phí thực thi trước đây, thông tin này chỉ hữu dụng khi một tập dữ liệu chỉ định đã được phân tích ít nhất một lần, ví dụ, các yêu cầu ARM (bao gồm *ngưỡng hỗ trợ* và *ngưỡng tin cậy*) tương tự nhau được đưa ra để xem xét trên lưới tri thức. Tuy nhiên siêu dữ liệu này có thể không sẵn có, trong trường hợp một tập dữ liệu mới được đưa ra phân tích lần đầu dẫn đến thiếu vắng thông tin về các chi phí thực thi, dịch vụ lưới RAEM sẽ đưa ra các cách giải quyết lập lịch một cách mù quáng. Để giải quyết được vấn đề này, ta sử dụng phương pháp lấy mẫu như là một cách thức để thu được tri thức về các chi phí thực thi của các tác vụ DM. Phương pháp này có khả năng trích rút tri thức đúng đắn từ một mẫu của tập lớn dữ liệu [4]. Tuy nhiên, thực tế tri thức khai phá được không phụ thuộc tuyến tính vào kích thước mẫu, việc xác định bao nhiêu dữ liệu cần được sử dụng là không thể. Do đó chúng tôi đề xuất một hướng sử dụng khác của phương pháp lấy mẫu: khi mà không có tri thức sẵn có về chi phí của một phép phân tích được chỉ định, ta thực hiện phép phân tích này trên một mẫu nhỏ của tập dữ liệu để dự đoán chi phí thực thi thực tế của nó trên tập dữ liệu đầy đủ. Việc xác định đúng kích thước của mẫu là mẫu chốt để trích rút tri thức chính xác, tuy nhiên, ta không thể áp dụng để đánh giá chi phí thực thi. Cùng với các chi phí thực thi, với phương pháp lấy mẫu ta có thể ước lượng được kích thước của các kết quả khai phá được, số lần vào/ra dữ liệu và dung lượng bộ nhớ chính yêu cầu. Các chi phí sẽ cung cấp các mô hình thực thi cụ thể

cho bộ lập lịch lưới tri thức để dự đoán chi phí truyền thông cần thiết, hiệu quả của việc phân bổ tài nguyên và ích lợi có thể nhận được từ việc thực thi song song.

2. Lập lịch phân tán trong môi trường lưới tri thức

Một bộ định giá lưới hoạt động như sau:

- + Khai phá một số tài nguyên phù hợp với các yêu cầu tối thiểu cho việc thực thi.
- + Kiểm tra các giấy phép trong việc chọn một công việc để giải quyết trên các tài nguyên đó.
- + Chọn các nguồn tài nguyên phù hợp nhất với các yêu cầu thực thi ứng dụng và lập lịch làm việc.

Các giải thuật lập lịch có thể được phân thành 2 dạng: Lập lịch làm việc động và lập lịch làm việc tĩnh. Dựa vào đặc điểm của các công việc khai phá dữ liệu, thường có tác động lẫn nhau, chiến lược lập lịch tốt nhất nên được sử dụng trong thiết kế bộ lập lịch lưới tri thức là chiến lược lập lịch làm việc động. Báo cáo này cũng đề cập đến việc đánh giá tính khả thi và lợi ích của việc sử dụng bộ lập lịch trực tuyến mức thấp (local scheduler) tập trung của một tổ chức AO (AO có một vài nhóm máy tính được kết nối tạo thành Lưới).

2.1. Các tính toán cần thiết cho việc lập lịch làm việc

Gọi một tác vụ khai phá dữ liệu là t_i được định nghĩa đầy đủ trong phạm vi phép phân tích DM yêu cầu, tập dữ liệu phân tích D_i (có kích thước $|D_i|$) và các tham số người dùng u_i (được xác định trước). Tác vụ t_i trích rút một mô hình tri thức từ tập dữ liệu D_i . Đặt $\alpha_i(D_i)$ là dữ liệu được trích rút bởi t_i , kích thước của nó là $|\alpha_i(D_i)|$. Cuối cùng, mô hình tri thức được trích rút phải được chuyển đến một vị trí xác định trước.

Trước khi nghiên cứu chi tiết giải thuật sắp xếp và môi trường lưới, ta giả sử rằng:

- Một bộ lập lịch tập trung điều khiển việc sắp xếp các tác vụ DM trên nhiều máy tính khác nhau của Lưới AO trong môi trường lưới. AO gồm có một tập $M = \{m_1, \dots, m_{|M|}\}$ máy tính, p_j là thừa số thực thi ứng với máy tính m_j . Thừa số thực thi cho biết tốc độ của các máy tính trong lưới. Trong báo cáo này chúng tôi không xem xét tính đa nhiệm của nút và không đưa vào tham số truyền tải của máy tính ở bên ngoài có thể hưởng đến các thừa số thực thi.

Ngoài ra, các máy tính được sắp xếp như một tập các nhóm $CL = \{cl_1, \dots, cl_{|CL|}\}$, mỗi nhóm cl_j bao gồm một tập các máy tính rời nhau trong M máy có kết nối với nhau bởi một mạng có tốc độ truyền thông cao. Cụ thể, $cl_j = \{m_1^j, m_2^j, \dots, m_{|cl_j|}^j\}$, mỗi cl_j là có khả năng là một máy điều khiển việc thực thi song song một phân tích DM đã định. Các thừa số thực thi của một nhóm cl_j là p_j , p_j bằng với thừa số thực thi của máy chạy chậm nhất trong nhóm.

- Mã chương trình (tuần tự hay song song) thực thi công việc DM được xem là sẵn có ở mỗi nút của lưới. Vì thế, vấn đề sắp xếp chính là việc chọn lựa tác vụ nào nên được gán cho máy nào thực hiện là phù hợp nhất, liên quan đến thời gian truyền thông cần thiết cho việc di chuyển dữ liệu vào/ra và các thời điểm máy tính rỗi và các liên kết đang sẵn sàng.

- Dựa trên nền tảng của phương pháp pháp lấy mẫu, có khả năng ước lượng e_i là chi phí tính toán tuần tự để thực hiện tác vụ t_i trên tập dữ liệu D_i với các tham số người dùng u_i . Đặt $e_{ij} = p_j * e_i$ là thời gian thực thi thực tế của tác vụ t_i trên máy m_j ; Khi một phép phân tích được thực hiện song song trên một nhóm cl_j , nếu như việc cân bằng tải được giải quyết tốt, tác vụ t_i có thể

được thực thi song song với tốc độ gần như hoàn hảo. Đặt e_{ij} là thời gian thực thi của tác vụ t_i trên một nhóm cl_j bằng $\max_{m'_i \in cl_j} (e_{i'} / |cl_j|) + ovh = \max_{m'_i \in cl_j} ((p_i * e_i) / |cl_j|) + ovh$. Điều kiện ovh là mô hình tạp phí của việc thực thi song song và tính không đồng nhất của nhóm. Xét trường hợp, khi một nhóm là đồng nhất và e_i là đủ lớn thì ovh là luôn luôn nhỏ.

- Một tập dữ liệu D_i có thể được tập trung hay được phân tán. Trong trường hợp các tập dữ liệu được phân bố không cố định. Vì thế, một tập dữ liệu chỉ được di chuyển khi điều đó có ích cho việc rút ngắn thời gian hoàn thành công việc. Chẳng hạn, một tập dữ liệu tập trung được lưu trữ tại vị trí h có thể được di chuyển sang vị trí j và chi phí di chuyển phụ thuộc vào dải thông mạng trung bình b_{hj} giữa hai vị trí đó. Khi đó, D_i có thể di chuyển với chi phí là $|D_i|/b_{hj}$

Việc di chuyển dữ liệu giữa các vị trí được chuyển ra ngoài bởi bộ quản lý nhân bản của các dịch vụ lưới mức thấp. Các truy cập tới một tập dữ liệu trong tương lai có thể có được lợi thế do các bản sao khác nhau được phổ biến trong lưới. Vì thế, lúc tác vụ t_i cần được sắp xếp, ta phải xem xét, đối với mỗi máy tính, ta phải lựa chọn bản sao tập dữ liệu nào có lợi nhất cần chuyển đi hay được truy cập.

2.2. Mô hình chi phí

Giả sử rằng mỗi tập dữ liệu đầu vào được lưu trữ trên một máy tính đơn m_h , khi mô hình tri thức được trích rút ra phải được di chuyển đến một máy tính m_k . Dựa vào cách giải quyết được đưa ra bởi bộ lập lịch, các tập dữ liệu có thể được di chuyển đến các máy khác và vì thế được nhân bản hay có thể được phân hoạch cho các máy tính khác nhau của một nhóm máy để thực hiện song song. Vì vậy, bộ lập lịch phải đưa ra bảng báo cáo về một số bản sao (nhân bản hay phân tán) của một tập dữ liệu có thể tồn tại trên nhiều máy tính của lưới AO của nó.

Thực thi tuần tự: Giả sử tập dữ liệu đầy đủ được lưu trữ trên một máy đơn $m_h \in M$. Tác vụ t_i được thực thi tuần tự bằng việc chạy một đoạn mã trên máy tính m_j với thời gian thực hiện là e_{ij} . Ta cũng xem xét các truyền thông cần thiết để di chuyển D_i từ máy m_h đến máy m_j và các truyền thông khác để di chuyển kết quả $|\alpha_i(D_i)|$ đến máy m_k . Tổng số thời gian thực thi sẽ là:

$$E_{ij} = \frac{|D_i|}{b_{hj}} + e_{ij} + \frac{|\alpha_i(D_i)|}{b_{jk}}$$

Thực thi song song: Tác vụ t_i được thực thi song song bởi việc chạy một đoạn mã trên một nhóm cl_j với thời gian thực thi là e_{ij} . Ta cũng phải xem các truyền thông cần thiết để di chuyển D_i từ máy m_h đến cl_j và để di chuyển kết quả $|\alpha_i(D_i)|$ đến máy m_k . Tổng thời gian thực hiện sẽ là:

$$E_{ij} = \sum_{m'_i \in cl_j} \frac{|D_i| / |cl_j|}{b_{ht}} + e_{ij} + \sum_{m'_i \in cl_j} \frac{|\alpha_i(D_i)| / |cl_j|}{b_{ik}}$$

Các chi phí truyền thông liên quan là bằng 0 nếu tập dữ liệu này đã được phân tán và được định vị trên các máy tính của nhóm cl_j .

Xét giải thuật song song, chúng ta cùng phân phối các yêu cầu và lập lịch cho tất cả các máy tính của nhóm, ta giả sử rằng các tiến trình song song được kết hợp với nhau. Một mô hình thực thi khác nên được sử dụng nếu chúng ta sử dụng giải thuật DM phân bố bất đồng bộ hơn, trước tiên các tính toán phụ thuộc được thực thi trên các phân hoạch tập dữ liệu riêng biệt và rồi các kết quả khác nhau của phép phân tích khai phá phân tán được tập hợp lại để thu được kết quả cuối cùng

Các thước đo thực thi: E_{ij} và E_{ij} là thời gian thực thi tổng cộng mong muốn của tác vụ t_i khi không có sự truyền tải hiện diện trong hệ thống. Khi mà sự truyền tải hiện hữu trong các máy tính và trong mạng, việc lập lịch sẽ trì hoãn thời điểm bắt đầu và thời gian hoàn thành một tác vụ. Trong phần sau chúng ta sẽ phân tích thời gian hoàn thành thực tế của một tác vụ trong trường hợp thực hiện tuần tự. Tương tự các phân tích cũng được làm trong trường hợp song song.

Đặt C_{ij} là thời gian thực mà tất cả sự truyền thông và tính toán tuần tự đòi hỏi để thực thi hoàn thành tác vụ t_i . Để xác định C_{ij} ta cần xác định thời điểm bắt đầu các truyền thông và thực hiện tính toán. Đặt s_{hj} là thời điểm bắt đầu truyền thông cần thiết để di chuyển dữ liệu đầu vào từ máy h đến máy j , đặt s_j là thời gian bắt đầu việc thực thi tuần tự tác vụ t_i trên máy j và đặt s_{jk} là thời gian bắt đầu truyền thông cần thiết để di chuyển mô hình tri thức kết quả được trích rút ra từ máy j đến máy k . Từ các định nghĩa trên ta có:

$$C_{ij} = (s_{hj} + \frac{|D_i|}{b_{hj}}) + \delta_1 + e_{ij} + \delta_2 + \frac{|\alpha_i(D_i)|}{b_{jk}} = s_{hj} + E_{hj} + \delta_1 + \delta_2$$

Trong đó: $\delta_1 = s_j - (s_{hj} + \frac{|D_i|}{b_{hj}}) \geq 0$ và $\delta_2 = s_{jk} - (s_j + e_{ij}) \geq 0$

Vì vậy, nếu A_i là thời gian đến của tác vụ t_i và t_i là tác vụ duy nhất trong quá trình thực hiện của hệ thống thì thời gian hoàn thành của tác vụ trên máy m_j là $\overline{C_{ij}} = A_i + E_{ij}$

Giả sử rằng m_j là máy được chọn bởi giải thuật lập lịch để thực thi tác vụ t_i . Đặt $C_i = C_{ij}$ và $\overline{C_i} = \overline{C_{ij}}$. Đặt T là tập tất cả các tác vụ được lập lịch. Các đơn vị thời gian cho việc lập lịch hoàn thành được xác định là $\max_{t_i \in T} (C_i)$ và dùng để đo toàn bộ lượng dữ liệu đưa vào hệ thống.

3. Phương pháp dự đoán thực thi

Trước khi nghiên cứu chiến lược sắp xếp công việc dựa vào mô hình chi phí, chúng tôi xét tính khả thi của phương pháp chọn mẫu trong việc dự đoán việc thực thi của tác vụ DM.

Với phương pháp chọn mẫu, chúng ta có thể xác định vấn đề để rút ngắn thời gian hoàn thành của một tác vụ phức tạp, điều đó có lợi cho việc di chuyển dữ liệu và thực thi tác vụ đó trên một máy xác định ở xa với thời gian nhanh nhất, hay phân tán dữ liệu để thực thi tác vụ song song. Trong báo cáo này, chúng tôi không đề cập đến độ chính xác của tri thức được trích rút từ một tập dữ liệu được lấy mẫu, mà chỉ quan tâm đến độ chính xác của các dự đoán thực thi. Vì thế, chúng ta sẽ phân tích các yêu cầu bộ nhớ và thời gian thực thi của giải thuật DM được chỉ định như hàm kích thước mẫu khai thác được tức là khả năng thực hiện của giải thuật. Từ việc nghiên cứu tính khả thi, đối với mỗi giải thuật, các hàm đưa ra các tiêu chí đánh giá thu được từ phương pháp chọn mẫu, trả lại thời gian thực thi dự đoán, yêu cầu bộ nhớ cần thiết cho việc chạy các phương pháp khai phá tương tự khác trên tập dữ liệu đầy đủ.

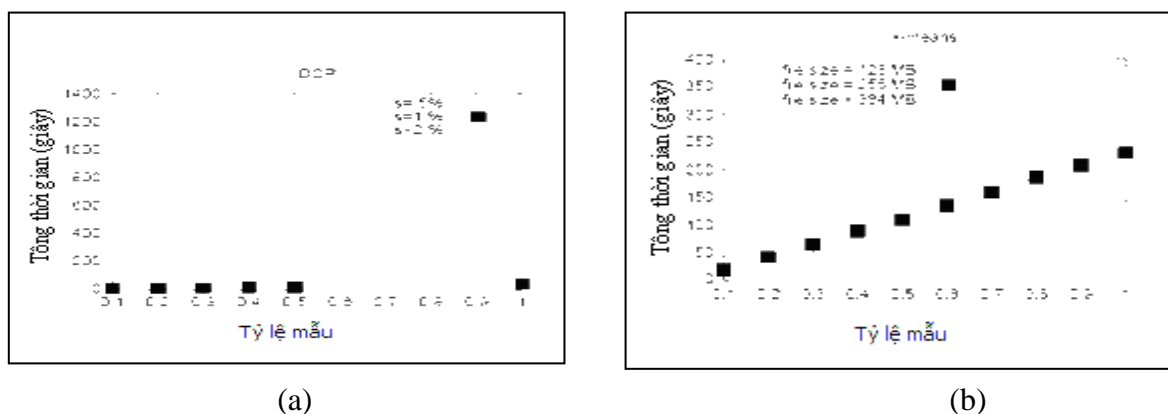
Giả sử một tác vụ cho trước t_i được thực thi đầu tiên trên mẫu \widehat{D}_i của D_i trên máy m_j , đặt \widehat{e}_{ij} là thời gian thực thi tác vụ này và $\widehat{e}_i = \widehat{e}_{ij} / p_j$ là thời gian thực thi chuẩn hoá đối với mẫu này. Phương pháp chọn mẫu là có thể thực hiện được nếu hàm $F()$ có dạng như sau: $e_i = F(\widehat{e}_i, D_i, \widehat{D}_i)$. Trong trường hợp đơn giản hàm F là một hàm tuyến tính của mẫu với hệ số tỷ lệ $s = |D_i| / |\widehat{D}_i|$ và ta có thể viết $e_i = \widehat{e}_i + (1-s)\beta$ trong đó β là hệ số góc đường cong.

Chúng tôi phân tích 2 giải thuật: DCP - giải thuật DM tối ưu đầy đủ cho việc phát hiện tập mục phổ biến và giải thuật K-means - giải thuật phân nhóm phổ biến và thực thi 2 giải thuật này trên tập dữ liệu nhân tạo bằng cách thay đổi kích thước các mẫu đưa ra.

Các kết quả của các thực nghiệm cho biết: cả hai giải thuật DCP và K-means tỷ lệ tuyến tính đối với kích thước mẫu khi các tham số của người sử dụng hay tập dữ liệu đầu vào là cố định. Hình 2.(a) cho thấy thời gian hoàn thành của giải thuật DCP trên tập dữ liệu với kích thước trung bình là 40MB giống như một hàm kích thước mẫu của tập dữ liệu đầu vào, với các tham số được đưa vào bởi người dùng (độ hỗ trợ tối thiểu). Trong hình 2.(b) thời gian hoàn thành của giải thuật K-means được đưa ra với tập dữ liệu khác, với tham số người dùng đó là số các nhóm tìm kiếm. Các kết quả thu được trên các tập dữ liệu khác nhau với các tham số khác là gần giống nhau.

Như những gì được chỉ ra từ các hình 2.(a,b), trong cả 2 trường hợp hệ số góc β phụ thuộc vào các tham số người dùng u_i (hình 2.(a)) và phụ thuộc vào tập dữ liệu đầu vào D_i (hình 2.(b)). Để có thể sử dụng phương pháp chọn mẫu như là một phương thức để dự đoán việc thực thi chúng ta cần xác định β phụ thuộc như thế nào vào u_i và D_i

Nghiên cứu về tính phụ thuộc trong giải thuật giúp ta có thể xác định đặc điểm của tập dữ liệu từ đó cho phép xác định được các tham số mà β phụ thuộc vào - dựa vào tập các tham số người dùng u_i . Ví dụ, với giải thuật K-means, β phụ thuộc vào kích thước của dữ liệu đầu vào $|D_i|$. Còn với giải thuật DCP, việc thực thi của giải thuật phụ thuộc nhiều vào số các tập mục được tìm thấy và độ hỗ trợ của các tập mục này hơn là phụ thuộc vào kích thước tập dữ liệu. Nếu như có thể tiến hành xây dựng độc lập một mô hình thực thi, tập trung vào việc xác định giá trị β đối với các giải thuật khác nhau, các đặc điểm tập dữ liệu khác nhau và các tập tham số khác nhau, sau đó, trong lúc bộ lập lịch trực tuyến hoạt động, chúng ta sử dụng các giá trị tìm thấy để dự đoán chi phí thực hiện thực tế. Lập luận tương tự cho các thước đo đánh giá việc thực thi khác chẳng hạn thời gian chiếm giữ bộ nhớ chính hay hoạt động vào ra.



Hình 2. Thời gian thực thi của giải thuật DCP (a) và kích thước kết quả đầu ra khác nhau (b).

Ở góc độ phân tích này chúng ta có thể kết luận rằng phương pháp chọn mẫu có thể được sử dụng như một phương pháp dự đoán thực thi hiệu quả cho nhiều trường hợp.

4. Lập lịch làm việc trực tuyến cho các tác vụ DM

Bộ sắp xếp trực tuyến tập trung dựa vào phương pháp MCT (thời gian hoàn thành nhỏ nhất) [5,3], lập lịch cho các tác vụ DM trên lưới tri thức nhỏ. Bộ sắp xếp này không xét đến hoạt

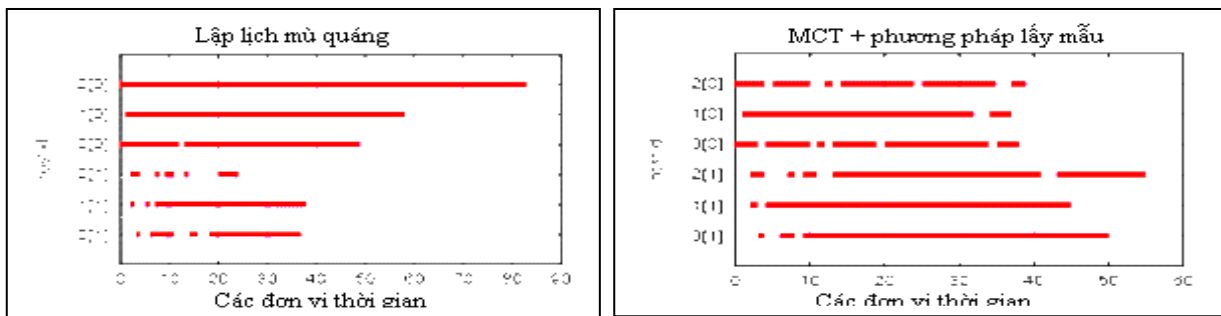
động đa nhiệm của nút trong Lưới, chịu trách nhiệm chọn lựa lịch biểu cho các di chuyển tập dữ liệu và các tính toán bao gồm việc thực thi một tác vụ t_i đã định, cũng như thời điểm bắt đầu thực thi của các tác vụ và kiểm tra sự hoàn thành của chúng.

Phương pháp sắp xếp dựa vào MCT rất là đơn giản. Mỗi thời điểm một tác vụ t_i được đưa ra xem xét, bộ sắp xếp định giá *thời gian sẵn sàng kỳ vọng* của mỗi máy và liên kết truyền thông, thời gian sẵn sàng kỳ vọng là một giá trị ước lượng - thời điểm sớm nhất một tài nguyên được chỉ định đã sẵn sàng sau khi thực thi các công việc được gán với nó trước đó. Để có được ước lượng này phải dựa vào cả các thời gian ước lượng và thời gian thực thi thực tế của tất cả các tác vụ mà được gán tài nguyên trước đây. Để cập nhật các thời gian sẵn sàng của tài nguyên, khi mà các di chuyển dữ liệu hay các thao tác tính toán bao gồm việc thực thi hoàn thành tác vụ t_i , một báo cáo được chuyển đến bộ sắp xếp. Bộ sắp xếp sau đó đánh giá tất cả các cách thực thi có thể đối với tác vụ t_i và chọn lựa một trong số các cách đó để rút gọn thời gian hoàn thành tác vụ này.

Chú ý, bộ sắp xếp dựa vào MCT có thể đem lại các cách giải quyết lập lịch một cách chính xác chỉ khi thời gian thực thi mong muốn của một tác vụ đã được biết. Khi không có một dự đoán thực thi nào sẵn có cho tác vụ t_i , đầu tiên bộ sắp xếp sinh và lập lịch cho tác vụ \hat{t}_i tức là thực thi tác vụ t_i trên một mẫu tập dữ liệu \hat{D}_i . Tuy nhiên, *thời gian thực thi kỳ vọng* của tác vụ mẫu \hat{t}_i là không được biết, bộ sắp xếp giả sử rằng thời gian thực thi ở mức trung bình và bằng một hằng số (nhỏ) cho trước. Ngoài ra, bộ sắp xếp dựa vào MCT không thực thi thử để tối ưu thời gian hoàn thành tác vụ \hat{t}_i mà chỉ đơn giản là gán \hat{t}_i cho máy quản lý tập dữ liệu vào D_i thực hiện, vì thế không có các di chuyển dữ liệu được yêu cầu khi thực thi tác vụ. Khi tác vụ \hat{t}_i hoàn thành, bộ sắp xếp biết được các dự đoán thực thi của tác vụ hiện tại t_i và sử dụng thông tin này để tối ưu việc sắp xếp công việc tiếp theo và việc lập lịch của nó.

Mô hình mô phỏng và tính khả thi của phương pháp lập lịch

Để đánh giá bộ lập lịch trực tuyến dựa vào MCT dùng để khai thác mẫu giống như một kỹ thuật dự đoán thực thi, ta đi xây dựng một biểu đồ mô phỏng cho phép chúng ta so sánh phương pháp của ta với chiến lược sắp xếp mù quáng.

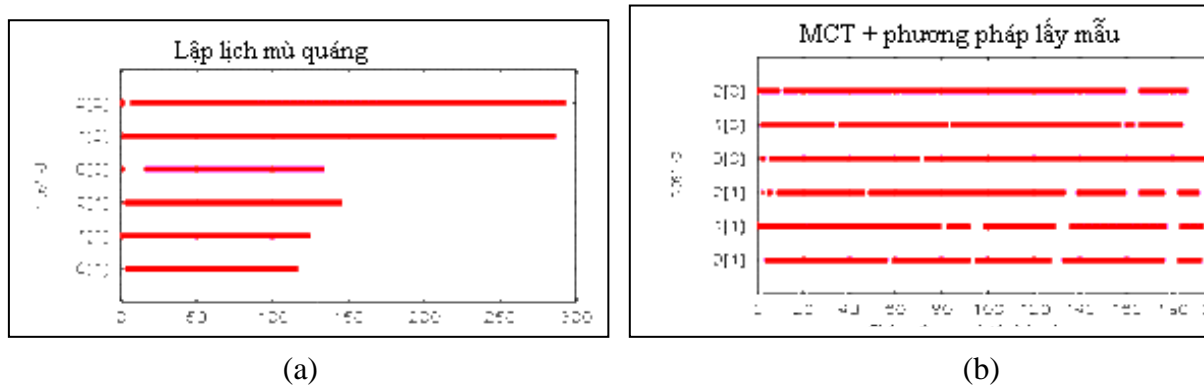


(a)

(b)

Hình 3. Biểu đồ biểu diễn thời gian bận (trong khoảng thời gian 100 giây) của 6 máy trong trường hợp chỉ có 10 trong số 100 tác vụ là thường xuyên thực hiện:

(a) phương pháp lập lịch mù quáng, (b) Phương pháp lập lịch dựa vào MCT kết hợp với phương pháp chọn mẫu.



Hình 4. Biểu đồ biểu diễn thời gian bận (trong khoảng thời gian 100 giây) của 6 máy khi 60 trong 100 tác vụ là thực hiện thường xuyên:

(a) phương pháp lập lịch mù quáng, (b) Phương pháp lập lịch dựa vào MCT kết hợp với phương pháp chọn mẫu.

Một môi trường lưới, bao gồm có 2 nhóm, mỗi nhóm 3 có máy. Mỗi nhóm được kết nối với nhau bởi một mạng ethernet có tốc độ truyền cao, một kết nối mạng WAN có tốc độ chậm tồn tại giữa 2 nhóm này. Hai nhóm máy này là đồng nhất (cùng chủng loại) nhưng các máy của nhóm 1 chạy nhanh hơn gấp 2 lần các máy của các nhóm khác. Để đặt các tham số mô phỏng, chúng ta lần lượt đo độ rộng trung bình các dải thông b_{WAN} và b_{LAN} của mạng WAN và mạng LAN.

Chúng ta giả sử rằng các tác vụ DM được lập lịch chuyển đến theo từng lô. Các chi phí thực thi là ngẫu nhiên, nhưng x% trong số đó là các tác vụ được thực hiện thường xuyên (1000s là thời gian thực thi liên tục trung bình trên máy có tốc độ chậm nhất), trong khi (100-x)% trong số đó là các tác vụ ít thường xuyên hơn (50s là thời gian thực thi liên tục trung bình trên máy chậm nhất).

Các tập dữ liệu D_i có kích thước trung bình (50MB) được đặt ngẫu nhiên trên các máy thuộc 2 nhóm. Mô hình mô phỏng đầu tiên là tập trung chủ yếu vào việc kiểm tra ích lợi của hướng tiếp cận của ta trước khi thi hành nó bên trong dịch vụ RAEM của lưới tri thức. Mục đích của chúng ta là đánh giá chất lượng sắp xếp công việc của phương pháp lập lịch dựa vào MCT kết hợp với phương pháp lấy mẫu trong giới hạn đơn vị thời gian. Để xác định việc sắp xếp tối ưu này, chúng ta giả sử biết được chi phí chính xác của tác vụ mẫu. Chúng ta có thể thấy ngay được những ích lợi của hướng tiếp cận lập lịch làm việc dựa vào MCT kết hợp với phương pháp lấy mẫu so với chiến lược lập lịch thông thường.

Hình 3. và hình 4. biểu diễn thời gian bận của các máy. Máy i của nhóm j được biểu thị bởi nhãn $i[j]$, khi nhóm 0 chậm hơn các nhóm khác và không có tập dữ liệu được di chuyển bởi chiến lược lập lịch mù quáng, các đơn vị thời gian trên các máy tính chậm hơn lại có hiệu suất làm việc nhiều hơn. Chiến lược lập lịch dựa vào MCT kết hợp với phương pháp lấy mẫu, mặc dù, giai đoạn đầu chi phí tính toán cao hơn do dựa theo phương pháp chọn mẫu, tuy nhiên, trong giai đoạn sau nó hoạt động hợp lý hơn.

5. Kết luận

Trong báo cáo này chúng ta đã áp dụng chiến lược tự tìm tòi MTC trực tuyến cho việc lập lịch làm việc để thực thi các tác vụ khai phá dữ liệu trên một tổ chức cục bộ của lưới tri thức. Các giải pháp lập lịch được đưa ra dựa trên các cơ sở của thước đo thực thi và các mô hình thực thi dựa trên thông tin được tập hợp trong các lần thực thi trước đây, kết hợp với sử

dụng phương pháp lấy mẫu để dự đoán chi phí thực thi. Ban đầu, chúng ta cũng có được một số kết quả khả thi trong việc ứng dụng phương pháp lập lịch này so với một số phương pháp khác. Các kỹ thuật sắp xếp và lập lịch sẽ được thừa kế bởi một bộ sắp xếp trực tuyến tập trung, nó là một phần của một bộ lập lịch mức cao hơn.

Với bộ sắp xếp trực tuyến, chỉ nghiên cứu trong giới hạn là không cho phép đa tác vụ ở các nút và lập lịch làm việc cho các tác vụ theo lô. Hướng nghiên cứu tiếp theo sẽ giải quyết các vấn đề này, chẳng hạn, bộ sắp xếp có thể chọn để thực thi đồng thời một tính toán và một tác vụ I/O trên cùng một máy. Bên cạnh, chúng tôi sẽ bổ sung thêm một số ràng buộc mang tính kinh tế cho việc lập lịch làm việc, chẳng hạn như giới hạn về ngân quỹ hay giới hạn thời gian thực hiện.

Mặt hạn chế của kỹ thuật của chúng ta là phải tiêu tốn chi phí cho việc chọn mẫu, dù thế phương pháp chọn mẫu vẫn được công nhận là một kỹ thuật khả thi so với một số kỹ thuật khác. Tất nhiên, các mô hình tri thức trích rút được bởi các tác vụ chọn mẫu có thể trong một số trường hợp là hữu ích đối với người dùng, người có thể đưa ra quyết định trên các cơ sở các kết quả chọn mẫu để bỏ qua hay tiếp tục thực thi trên tập dữ liệu đó. Trên một phương diện khác, khi mà các kết quả thu được bởi phương pháp chọn mẫu biểu diễn đúng một mô hình tri thức không hoàn chỉnh được trích rút từ một phân hoạch của tập dữ liệu, chúng ta có thể bỏ qua và không lưu giữ lại các kết quả đó. Ngoài ra, chúng ta có thể khai thác các giải thuật khai phá dữ liệu khác cũng phù hợp trong môi trường phân tán, ở đây các phân tích DM độc lập được thực hiện trên các phân hoạch dữ liệu khác nhau và rồi các kết quả rời rạc được tập hợp lại. Theo hướng tiếp cận này, tri thức được trích rút từ mẫu \hat{D}_i có thể được giữ lại và rồi sau đó kết hợp với một mẫu thu được từ việc thực thi tác vụ trên các tập dữ liệu đầu vào còn lại $D \setminus \hat{D}_i$.

Tóm tắt

Khối lượng các tập dữ liệu được sử dụng cho khai phá dữ liệu đang ngày càng đồ sộ và được lưu trữ khắp nơi. Vì thế, quá trình khai phá tri thức phân tán cần một khối lượng lớn dữ liệu và với sự trợ giúp của nhiều máy tính. Môi trường lưới là một nền tảng cơ sở cho việc triển khai một hệ thống dịch vụ khai phá dữ liệu với hiệu suất cao. Nội dung của báo cáo này đề cập đến các dịch vụ then chốt của một hệ thống lưới. Đặc biệt, chúng ta tập trung đến việc thiết kế và thực thi việc phân phối tài nguyên, dịch vụ quản lý thực thi cung cấp thông tin về các vị trí nguồn dữ liệu và nhu cầu tài nguyên của các tác vụ khai phá dữ liệu. Báo cáo này giới thiệu cách giải quyết lập lịch làm việc và phân công công việc được đưa ra dựa trên độ đo chi phí thực thi, các mô hình khai thác tri thức liên quan đến các thực thi trước đây và sử dụng phương pháp lấy mẫu để thu được ước lượng thực thi liên quan đến hoạt động khai phá dữ liệu.

Summary

The datasets used for data mining are increasing and physically distributed. So, the distributed knowledge discovery process is both data and computational intensive, the Grid is a natural platform for developing a higher performance data mining service. The key content of this paper focuses on the core services of such a Grid infrastructure. In particular, we concentrate our attention on the design and implementation of specialized Resource Allocation and Execution Management services aware of data source locations and resource needs of data mining tasks. Allocation and scheduling decisions are taken on the basis of performance cost metrics and models that exploit knowledge about previous executions, and use sampling to acquire estimate about execution behavior.

Tài liệu tham khảo

- [1]. Vũ Đức Quảng (2007), *Khai phá dữ liệu, luật kết hợp và thuật toán khai phá luật kết hợp song song*, luận văn Thạc sỹ, Đại học Sư Phạm Hà Nội .
- [2]. A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke (2001), *The Data Grid: towards an architecture for the distributed management and analysis of large scientific datasets*, Journal of Network and Computer Applications, (23):p.187–200.
- [3]. D. Talia and M., Cannataro. *Knowledge grid: An architecture for distributed knowledge discovery*, Comm. of the ACM, 2002 (to appear).
- [4]. H. J. Siegel and Shoukat Ali, *Techniques for Mapping Tasks to Machines in Heterogeneous Computing Systems*, Journal of Systems Architecture, (46):627–639, 2000.7
- [5]. M. J. Zaki, S. Parthasarathy, W. Li, and M. Ogihara, *Evaluation of sampling for data mining of association rules*, In 7th International Workshop on Research Issues in Data Engineering (RIDE—in conjunction with ICDE), pages 42–50, 1997.
- [6]. M. Maheswaran, A. Shoukat, H. J. Siegel, Siegel, D. Hensgen, and R. F. Freund, *Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems*, In 8th Heterogeneous Computing Workshop (HCW '99), 1999.
- [7]. R. Baraglia, D. Laforenza, S. Orlando, P. Palmerini, and R. Perego, *Implementation issues in the design of I/O intensive data mining applications on clusters of workstations*, In Proc. of the 3rd Workshop on High Performance Data Mining, Cancun, Mexico. Springer-Verlag, 2000.
- [8]. S. Fitzgerald, I. Foster, C. Kesselman, G. von Laszewski, W. Smith, and S. Tuecke, *A directory service for configuring high-performance distributed computations*, In Proc. 6th IEEE Symp. on High Performance Distributed Computing, pages 365–375. IEEE Computer Society Press, 1997.
- [9]. S. Orlando, P. Palmerini, and R. Perego, *Enhancing the Apriori Algorithm for Frequent Set Counting*, In Proc. of 3rd Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK 01) - Munich, Germany. LNCS Springer-Verlag, 2001.