

THUẬT TOÁN MODE GIẢI BÀI TOÁN LẬP LỊCH LUỒNG CÔNG VIỆC

An Algorithm MODE for Workflow Scheduling

Phan Thanh Toàn, Nguyễn Thế Lộc, Nguyễn Doãn Cường

Abstract: Cloud computing is a general term for the delivery of hosted services over the Internet which facilitates pay per model as per user demand and requirement. Cloud consist of a collection of servers which includes both computational and storage facility. The primary aim of cloud computing is to provide efficient access to remote and geographically distributed resources. There are many challenges in the cloud environment, one of them is workflow scheduling, although workflow scheduling problem has been discussed by many researchers, a handful efficient solutions have been proposed for Cloud computing. In this paper we propose MODE, a novel algorithm for workflow scheduling problem. Based on the Opposition-Based Differential Evolution method, our proposed algorithm not only ensures the fast convergence but also avoid being trapped on local extrema. Our experiments conducted in real scenarios by using simulation tools CloudSim show that MODE is superior to previously proposed algorithms, moreover the deviation between the solution found by MODE and the optimal solution is negligible.

Keyword: Workflow scheduling, Opposition-Based Differential Evolution, cloud computing, Differential Evolution.

I. GIỚI THIỆU

Bài toán lập lịch luồng công việc là một bài toán đã được nghiên cứu từ những năm 1950, và bài toán này đã được chứng minh thuộc lớp NP-Khó. Trong những năm gần đây đã có rất nhiều ứng dụng khoa học được mô hình hóa bởi dạng đồ thị luồng công việc như ứng dụng Montage [1], CyberShake [2],

Epigenomics [3], LIGO [4],... một trong những thách thức của bài toán lập lịch luồng công việc là phải hoàn thành luồng công việc với thời gian nhỏ nhất trong điều kiện giới hạn về nguồn tài nguyên. Sự phát triển của môi trường điện toán đám mây (Cloud Computing) đã tạo ra các cơ hội cho việc giải quyết bài toán lập lịch luồng công việc, với khả năng về tài nguyên dự phòng và luôn sẵn dùng sẽ giúp giảm bớt thời gian chờ đợi giữa các tác vụ trong luồng công việc, khả năng cung cấp tài nguyên theo nhu cầu khách hàng cũng là một lợi thế lớn của điện toán đám mây, khả năng này sẽ cho phép các ứng dụng dạng luồng công việc có thể bắt đầu được thực hiện ngay khi chỉ có một phần tài nguyên được đáp ứng mà không cần phải chờ đợi cho đến khi đủ tất cả các tài nguyên cần thiết. Ngoài ra khả năng co giãn trong môi trường điện toán đám mây cũng là một đặc trưng hữu ích với các ứng dụng dạng luồng công việc vì nó cho phép các hệ thống luồng công việc dễ dàng tăng, giảm các tài nguyên của ứng dụng theo nhu cầu. Bản chất của vấn đề lập lịch luồng công việc trong môi trường điện toán đám mây là gán các tác vụ của luồng công việc vào thực thi trên các tài nguyên của đám mây sao cho thời gian hoàn thành luồng công việc (Makespan) là nhỏ nhất.

Nội dung tiếp theo của bài báo gồm những phần chính như sau. Phần 2 trình bày một số công trình liên quan đến bài toán lập lịch luồng công việc. Phần 3 mô tả bài toán và trình bày mô hình toán học, sau đó phát biểu bài toán và chứng minh rằng nó thuộc lớp NP-Khó. Phần 4 giới thiệu thuật toán đề xuất - MODE. Để kiểm chứng hiệu năng của thuật toán MODE, phần 5 trình bày quá trình thực nghiệm trên một số luồng công việc khoa học mẫu trong môi trường đám mây thông qua công cụ mô phỏng CloudSim và gói thư viện Jswarm [5] và phân tích số liệu thu được, từ đó

đưa ra những nhận xét so sánh. Nội dung phần 6 trình bày kết luận và hướng nghiên cứu tiếp theo.

II. NHỮNG CÔNG TRÌNH LIÊN QUAN

Bài toán lập lịch luồng công việc đã được chứng minh là thuộc lớp NP-Khó [6] nghĩa là thời gian để tìm ra lời giải tối ưu tăng rất nhanh theo kích cỡ dữ liệu đầu vào, vì vậy đã có nhiều công trình nghiên cứu nhằm tìm ra lời giải gần đúng trong thời gian ngắn.

S. Sadhasivam đã đề xuất thuật toán lập lịch luồng công việc dựa trên sự cân bằng tải trong môi trường điện toán đám mây [7]. Thuật toán không chỉ đáp ứng các yêu cầu từ người sử dụng mà còn cung cấp khả năng sử dụng tài nguyên một cách hiệu quả. Đây là thuật toán theo hướng nâng cao hiệu quả dịch vụ dựa trên Meta-heuristic. R. Burya đã trình bày một cách tóm tắt về các chức năng của công cụ mô phỏng CloudSim [8] - môi trường mô phỏng cho phép cài đặt và thực nghiệm các thuật toán lập lịch luồng công việc trong môi trường điện toán đám mây. G. Guo-Ning đã đề xuất một thuật toán lập lịch luồng công việc dựa trên giải thuật di truyền [9]. trong đó đưa vào nhiều ràng buộc dịch vụ khác nhau như thời gian hoàn thành, băng thông, chi phí, độ tin cậy. Tác giả đã sử dụng kết hợp với giải thuật luyện thép sau pha lựa chọn, trao đổi chéo, đột biến nhằm tăng cường khả năng tìm kiếm cục bộ của giải thuật di truyền.

Các tác giả trong bài báo [10] đã đề xuất thuật toán EGA(Enhanced Genetic Algorithm) lập lịch bằng phương pháp di truyền. Trong công trình các tác giả sử dụng thuật toán Enhanced Max Min trong bước khởi tạo quần thể nhằm tìm ra các cá thể tốt cho quá trình tiến hóa.

S. Pandey [11] đã đề xuất thuật toán lập lịch luồng công việc PSO Heuristic (Particle Swarm Optimization Heuristic – PSO_H) trong môi trường điện toán đám mây dựa trên chiến lược tối ưu bầy đàn. R. Rajkumar đã đề xuất một thuật toán lập lịch phân cấp [12] và đưa vào các tham số dịch vụ khác nhau, chẳng hạn như thời gian đáp ứng. Thuật toán sử dụng tham số này như một quyền ưu tiên để lựa chọn các

tác vụ lập lịch. Q. Cao và các đồng nghiệp đã trình bày thuật toán lập lịch dựa trên giải thuật ABC (Activity Based Costing) [13]. Thuật toán này gán mức ưu tiên cho mỗi tác vụ trong luồng công việc theo các tham số về thời gian, không gian, các tài nguyên và chi phí, quá trình lập lịch sẽ sử dụng mức ưu tiên này để quyết định các tác vụ được chọn trong quá trình lập lịch.

S.Selvi và các cộng sự đã đề xuất thuật toán lập lịch luồng công việc trong môi trường điện toán lưới (Grid) [14], trong công trình tác giả đã vận dụng thuật toán tiến hóa vi phân (DE) vào giải bài toán lập lịch luồng công việc trên môi trường điện toán lưới nhằm cực tiểu thời gian hoàn thành luồng công việc (makespan), trong công trình tác giả đã chỉ ra giá trị Makespan tìm được bởi thuật toán đề xuất là nhỏ hơn so với thuật toán PSO. Q. XU và các cộng sự đã đề xuất thuật toán COODE [15] (Current Optimum Opposition-Based Differential Evolution) nhằm tìm giá trị tối ưu cho các hàm số dựa theo phương pháp tiến hóa vi phân đối xứng, trong công trình tác giả đã đề xuất công thức tìm điểm đối xứng của một điểm dựa theo giá trị tối ưu hiện tại nhằm thay đổi toán tử đột biến trong phương pháp tiến hóa vi phân và tác giả đã so sánh thuật toán COODE với các thuật toán DE và ODE, kết quả đã chỉ ra thuật toán đề xuất COODE tốt hơn các thuật toán đối sánh.

Các công trình hiện trên đều chưa hướng tới mục tiêu duy nhất là cực tiểu hóa thời gian hoàn thành của luồng công việc với các cấu trúc phức tạp và sự đa dạng về dữ liệu truyền qua lại giữa các tác vụ trong luồng công việc và chưa trình bày rõ ràng về mô hình toán học của bài toán lập lịch luồng công việc.

III. MÔ HÌNH LÝ THUYẾT

Hệ thống tính toán

Giả thiết cho trước hệ thống tính toán bao gồm:

- Tập hợp N máy chủ trong môi trường điện toán đám mây $S = \{S_1, S_2, \dots, S_N\}$.

- Luồng công việc cần thực hiện biểu diễn bởi đồ thị có hướng, không có chu trình $G=(V,E)$, mỗi đỉnh biểu thị một tác vụ, mỗi cạnh biểu diễn mối quan hệ cha-con giữa một cặp tác vụ.
- Tập các tác vụ $T=\{T_1, T_2, \dots, T_M\}$ với M là số lượng tác vụ.
- Khối lượng tính toán của tác vụ T_i ký hiệu là W_i , đo bằng đơn vị flop (floating point operations: phép tính trên số thực dấu phẩy động).
- Tốc độ tính toán của máy tính, đo bằng đơn vị flop/s (số phép tính thực hiện được trên giây), ký hiệu $P(\cdot)$, là hàm số được định nghĩa như sau:

$$P: S \rightarrow R^+$$

$$S_i \rightarrow P(S_i)$$

- Mọi cặp máy chủ (S_i, S_k) bất kỳ đều có đường truyền để trao đổi dữ liệu với nhau ($1 \leq i, k \leq N$).
- Băng thông của đường truyền, ký hiệu $B(\cdot)$, là tốc độ truyền dữ liệu giữa các máy chủ, đo bằng đơn vị bit trên giây (bps), là hàm số được định nghĩa như sau:

$$B: S \times S \rightarrow R^+$$

$$(S_i, S_k) \rightarrow B(S_i, S_k)$$

- Hàm băng thông $B(\cdot)$ tuân theo các ràng buộc sau:
 + $B(S_i, S_i) = \infty$: thời gian truyền từ một máy chủ tới chính nó bằng 0, nghĩa là nếu tác vụ cha và tác vụ con được bố trí trên cùng một máy chủ thì không mất thời gian để truyền dữ liệu giữa chúng vì dữ liệu đó được lưu trữ và sử dụng ngay tại chỗ.
 + $B(S_i, S_k) = B(S_k, S_i)$: kênh truyền hoạt động từ hai đầu với tốc độ tương đương nhau.
- Khối lượng dữ liệu cần truyền giữa hai tác vụ T_i và T_k , ký hiệu là D_{ik} , là các giá trị cho trước, $D_{ik} \neq 0$ khi và chỉ khi T_i là tác vụ cha của T_k , trong trường hợp ngược lại $D_{ik} = 0$.

Khái niệm lịch biểu

Một phương án xếp lịch F , còn gọi là lịch biểu F , được xác định bởi hai hàm (t_s, proc) trong đó

- $t_s: T \rightarrow R^+$; $t_s(T_i)$ là thời điểm mà tác vụ $n \in T$ bắt đầu được thực hiện

- $\text{proc}: T \rightarrow S$; $\text{proc}(T_i)$ là máy tính được phân công thực hiện tác vụ $T_i \in T$

Từ các giả thiết trên suy ra thời gian tính toán của tác vụ T_i là:

$$\frac{W_i}{P(\text{proc}(T_i))} \quad (i=1, \dots, M) \quad (1)$$

Thời gian truyền dữ liệu giữa tác vụ T_i và T_k là:

$$\frac{D_{ik}}{B(\text{proc}(T_i), \text{proc}(T_k))} \quad (i, k=1, \dots, M) \quad (2)$$

Makespan của lịch biểu F biểu diễn theo như sau:

$$\text{makespan}(F) = \max_{T_i \in T} \{t_f(T_i)\} - \min_{T_i \in T} \{t_s(T_i)\} \quad (3)$$

với $t_f(T_i)$ là thời điểm kết thúc và $t_s(T_i)$ là thời điểm bắt đầu thực hiện của tác vụ T_i

Mục tiêu của bài toán

Mục tiêu của bài toán là tìm lịch biểu F sao cho $\text{makespan}(F) \rightarrow \min$

Phát biểu và chứng minh độ phức tạp

Bài toán đang xét, từ đây ký hiệu là WSP - Workflow Scheduling Problem, được phát biểu như sau: *giả sử cho trước tài nguyên của đám mây bao gồm tập máy chủ S với tốc độ tính toán và băng thông được biểu diễn như trên. Giả sử tập tác vụ T được biểu diễn bởi đồ thị $G=(V,E)$ như trên. Hàm mục tiêu đặt ra là cực tiểu thời gian hoàn thành luồng công việc*

$$\text{makespan} \rightarrow \min$$

Bổ đề: WSP là bài toán thuộc lớp NP-Khó

Chứng minh:

Xét bài toán SCHED [16] sau đây: giả sử cho trước tập máy chủ và tập tác vụ gồm các thành phần như đã định nghĩa ở bài toán WSP trên đây, ngoài ra bổ sung thêm điều kiện là tập S gồm các máy chủ giống hệt nhau về tốc độ tính toán.

Sinnen đã chứng minh rằng bài toán SCHED thuộc lớp NP-Khó [16]. Quay về bài toán WSP đang xét, dễ thấy rằng bài toán SCHED chỉ là một trường hợp riêng của bài toán WSP khi bổ sung thêm điều kiện ràng buộc sau đây: $P(S_i) = P(S_j) \forall i, j = 1..N$.

Vì vậy rõ ràng bài toán WSP cũng thuộc lớp NP-Khó.

IV. GIẢI PHÁP ĐỀ XUẤT

Thuật toán tiến hóa vi phân dựa trên thông tin đối xứng (Opposition-based differential evolution – ODE) được đề xuất bởi R. Storn và K. Price [17], tương tự như các thuật toán tiến hóa khác, thuật toán ODE gồm hai bước chính: khởi tạo quần thể và sinh quần thể mới bằng cách áp dụng các toán tử di truyền như đột biến, trao đổi chéo và lựa chọn tự nhiên. Tuy nhiên thuật toán ODE nâng cao hiệu năng tìm kiếm trong hai bước này bằng cách tìm kiếm thêm lời giải trong phần đối xứng của không gian tìm kiếm hiện tại. Để giải quyết bài toán lập lịch luồng công việc đã đề xuất trong mô hình phần 3, chúng tôi sẽ đề xuất phương pháp tìm phần tử đối xứng trong quần thể và kết hợp với phương pháp bánh xe quay vòng dựa trên hạng cá thể nhằm tìm ra các cá thể tốt cho quá trình đột biến qua đó nâng cao hiệu năng của thuật toán.

Định nghĩa 1: giả sử x là số thực và $x \in [a, b]$, khi đó phần tử đối xứng của x kí hiệu là $\bar{x}$ và được tính như sau: $\bar{x} = a + b - x$ (3)

Định nghĩa 2: gọi $P(x_1, x_2, \dots, x_D)$ là một véc tơ D-chiều, với $x_i \in [a_i, b_i]$; $i=1, 2, \dots, D$. Khi đó véc tơ đối xứng của P kí hiệu là $\bar{P} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_D)$ và được tính như sau: $\bar{x}_i = a_i + b_i - x_i$ (4)

4.1. Biểu diễn cá thể

Mỗi cá thể trong quần thể được biểu diễn bởi một véc tơ có độ dài bằng số tác vụ trong luồng công việc. Giá trị tương ứng với mỗi vị trí i trong véc tơ biểu diễn số hiệu của máy chủ thực thi tác vụ i .

Ví dụ: xét luồng công việc với 5 tác vụ $T=\{T_1, T_2, \dots, T_5\}$, tập máy chủ gồm 3 máy $S=\{S_1, S_2, S_3\}$. Khi đó cá thể $x_i=(1, 2, 1, 3, 2)$ được biểu diễn như sau:

T_1	T_2	T_3	T_4	T_5
S_1	S_2	S_1	S_3	S_2

4.2. Phương pháp tính đối xứng cho cá thể

Trong phương pháp ODE ta cần phải tìm cá thể đối xứng cho mỗi cá thể trong quần thể hiện tại, trong

bài báo này chúng tôi đề xuất phương pháp tìm cá thể đối xứng như sau:

Gọi $a = \text{Max}\{P(S_i)\}$ và $b = \text{Min}\{P(S_i)\}$; $\forall i=1, 2, \dots, N$; với $P(S_i)$ là tốc độ tính toán của máy chủ S_i

Giả sử ta có cá thể $x_i = (S_{i\pi(1)}, S_{i\pi(2)}, \dots, S_{i\pi(M)}); S_{i\pi(j)} \in S, \forall j=1, 2, \dots, M$; cá thể đối xứng của x_i kí hiệu là $\bar{x}_i$ và $\bar{x}_i = (\bar{S}_{i\pi(1)}, \bar{S}_{i\pi(2)}, \dots, \bar{S}_{i\pi(M)})$ (5);

$$\bar{S}_{i\pi(j)} = a + b - S_{i\pi(j)}; \forall j = 1, 2, \dots, M$$

Sau đó sẽ gán các giá trị tương ứng với mỗi vị trí j trong véc tơ $\bar{x}_i$ bởi số hiệu máy chủ có tốc độ tính toán gần giá trị $\bar{S}_{i\pi(j)}$ nhất:

$$\bar{x}_{ij} \leftarrow k; |P(S_k) - \bar{S}_{i\pi(j)}| \leq |P(S_r) - S_{i\pi(j)}| \forall S_r \quad (6)$$

Algorithm: OP_Algorithm

Input: population $p = (x_1, x_2, \dots, x_{\text{PopSize}})$

Output: opposite of population OP

1. for $i=1$ to PopSize do
 2. $\bar{x}_i \leftarrow \text{opposite}(x_i)$; theo công thức (5)
 3. Gán số hiệu máy chủ cho các vị trí j của véc tơ $\bar{x}_i$ theo (6)
 4. end for
 - return OP
-

4.3. Phương pháp bánh xe quay vòng dựa trên hạng cá thể

Bánh xe quay vòng dựa trên hạng cá thể là một phương pháp lựa chọn cá thể cho các thế hệ kế tiếp. Trong phương pháp này mỗi cá thể được xếp hạng theo một hàm xác định, sau đó sẽ tính xác suất lựa chọn các cá thể theo hạng của chúng. Trong bài báo này chúng tôi đề xuất hàm tính hạng cho các cá thể như sau:

$$\text{rank}(\text{pos}) = 2 - \text{SP} + \left(2 \times (\text{SP} - 1) \times \frac{\text{pos} - 1}{\text{PopSize} - 1} \right) \quad (7)$$

Trong đó: $1.0 \leq \text{SP} \leq 2.0$; pos : vị trí cá thể cần tính hạng

Algorithm: RBRWS algorithm

Input: population $p = (x_1, x_2, \dots, x_{\text{PopSize}})$

Output: particle p_s

Begin

1. Sắp xếp các cá thể theo chiều tăng dần của hàm mục tiêu f_i
-

```

2. for i=1 to PopSize do
3.   pos[i] ← PopSize - 1
4. for i=1 to PopSize do
5.   calculate ranki by equation (7)
6. rand ← Random(0,SP)
7. s ← PopSize
8. while(rank[s] < rand && s>0) s-
   return xs
End.

```

4.4. Thuật toán MODE

Kết hợp các nội dung trên chúng tôi đề xuất thuật toán lập lịch luồng công việc trong môi trường điện toán đám mây dựa trên thông tin đối xứng và gọi là MODE như sau:

Algorithm **MODE** ()

Input: T, S, size of workload W[1×M], P[1×N], B[N×N], D[M×M], the constant K, the deviation ε, the number of particle NoP

Output: the best position *gbest*

```

1. Khởi tạo quần thể P gồm PopSize cá thể một cách ngẫu nhiên
2. OP ← OP_Algorithm ; tính quần thể đối xứng của quần thể hiện tại
3. Chọn PopSize cá thể tốt nhất từ P ∪ OP
4. while(Điều kiện lặp) do
5.   for i=1 to PopSize do
6.     Lựa chọn cá thể p1 theo thuật toán RBRWS
7.     Lựa chọn cá thể p2 theo thuật toán RBRWS
8.     F ← Random(1,0)
9.     vi ← pbest + F×(p1 - p2)
10.    Gán số hiệu máy chủ cho mỗi vị trí j của véc tơ vi theo (6)
11.    randi,j ← Random(0,1)
12.    Irand ← random(1,M)
13.    ui,j =  $\begin{cases} v_{i,j} & \text{nếu } rand_{i,j} \leq CR \text{ hoặc } i = I_{rand} \\ x_{i,j} & \text{nếu } rand_{i,j} \geq CR \text{ hoặc } i \neq I_{rand} \end{cases}$ 
14.    if (makespan(ui) < makespan(xi))
15.      xi ← ui
16.    end if
17.  end for
18.  rand ← Random(0,1)
19.  if(rand < Jr)
20.    OP ← OP_Algorithm
21.    Lựa chọn PopSize cá thể tốt nhất từ P ∪ OP
22.  end if
23. End while

```

Return *gbest*;

Bước khởi tạo: khởi tạo ngẫu nhiên quần thể P gồm PopSize cá thể, quần thể đối xứng OP của P theo (5), (6), chọn ra PopSize cá thể tốt nhất từ P ∪ OP.

Trong mỗi bước lặp chọn ra hai cá thể p₁, p₂ theo phương pháp bánh xe quay vòng dựa trên hạng các cá thể. Véc tơ đột biến của mỗi cá thể i được tính theo công thức: v_i ← pbest + F×(p₁ - p₂)

Trong đó: pbest là cá thể tốt nhất hiện tại

Sau bước đột biến toán tử trao đổi chéo được áp dụng cho mỗi cá thể x_i để sinh ra cá thể mới u_i

$$u_{i,j} = \begin{cases} v_{i,j} & \text{nếu } rand_{i,j} \leq CR \text{ hoặc } i = I_{rand} \\ x_{i,j} & \text{nếu } rand_{i,j} \geq CR \text{ hoặc } i \neq I_{rand} \end{cases}$$

Trong đó rand_{i,j} ∈ [0,1], I_{rand} ∈ [1,M], CR ∈ [0,1].

Toán tử lựa chọn được áp dụng để quyết định cá thể nào được sống sót cho thế hệ kế tiếp

$$x_i = \begin{cases} u_i, & \text{if makespan}(u_i) < \text{makespan}(x_i) \\ x_i, & \text{otherwise} \end{cases}$$

Sau khi trao đổi chéo và lựa chọn, tính quần thể đối xứng OP của P theo (5), (6) và chọn ra PopSize cá thể tốt nhất từ P ∪ OP.

Quá trình tiến hóa của quần thể sẽ được thực hiện qua các toán tử đột biến, trao đổi chéo và lấy đối xứng quần thể. Sau mỗi thế hệ chúng ta sẽ tính giá trị hàm mục tiêu (Makespan) của các phương án xếp lịch và đối sánh với giá trị tốt nhất trong cá thể gbest, nếu có một cá thể cho giá trị makespan tốt hơn thì cá thể đó sẽ thay thế cá thể gbest hiện tại.

V. KẾT QUẢ THỰC NGHIỆM

Để kiểm chứng thuật toán đề xuất MODE chúng tôi đã sử dụng công cụ mô phỏng Cloudsim [5] để tạo lập môi trường đám mây. Đối tượng so sánh là thuật toán tiến hóa mạnh như PSO Heuristic [14], và thuật toán Random [18], và EGA [10].

Các chương trình mô phỏng được viết bằng ngôn ngữ Java và chạy trên máy tính cá nhân với bộ vi xử lý Intel Core i5 2.2 GHz, RAM 4GB, hệ điều hành Windows 7 Ultimate.

5.1. Phân nhóm dữ liệu thực nghiệm

Dữ liệu sử dụng trong các thực nghiệm bao gồm :

- Dữ liệu về tốc độ tính toán của các máy chủ và băng thông giữa các máy chủ được lấy từ các công ty cung cấp dịch vụ cloud như Amazon [19].
- Dữ liệu luồng công việc được lấy từ các bộ dữ liệu thử nghiệm được xây dựng theo độ trừ mật khác nhau và các luồng công việc từ các ứng dụng thực tế như ứng dụng Montage [1].
- Dựa theo tính chất của môi trường điện toán đám mây, đây là một môi trường tính toán không đồng nhất, tốc độ tính toán các máy chủ và băng thông không đồng đều, đồng thời cũng dựa theo tính chất các luồng công việc, số lượng tác vụ, độ trừ mật của đồ thị luồng công việc chúng tôi đã tiến hành phân loại dữ liệu theo các nhóm với quan hệ về tốc độ tính toán các máy chủ và băng thông khác nhau, độ trừ mật của đồ thị luồng công việc cũng được thử nghiệm qua hệ số α khác nhau. Chi tiết các nhóm dữ liệu theo số lượng máy chủ N, số tác vụ M và hệ số α như sau:
- Nhóm 1: M=5, N=3; Nhóm 2: M=10, N=3, Nhóm 3: M=20, N=8, Nhóm 4: M=25, N=8
- Mỗi nhóm lại bao gồm ba thực nghiệm khác nhau về tỷ lệ số cạnh trên số đỉnh của đồ thị luồng công việc, ký hiệu là α và tính bởi công thức: $\alpha = \frac{|E|}{M \times (M-1)/2}$

5.2. Tham số cấu hình

Các tham số cấu hình của đám mây được thiết lập trong miền giá trị như sau:

- Tốc độ tính toán P của các máy chủ: từ 1 đến 250 (million instructions/s); khối lượng dữ liệu D giữa các tác vụ: từ 1 đến 1000 (MB); băng thông giữa các máy chủ B: từ 10 đến 100 (Mega bit/s).

- Hệ số vi sai F=0.5; hệ số trao đổi chéo CR=0.9; $J_r=0.3$; PopSize=50; α : từ 0.2 tới 0.7.

5.3. Quá trình tiến hành thực nghiệm

Để đánh giá hiệu năng của thuật toán đề xuất MODE chúng tôi đã tiến hành thực nghiệm và so sánh kết quả của MODE với các thuật toán tiến hóa mạnh hiện nay như PSO_H và Random. Dữ liệu thực nghiệm được lấy từ ứng dụng Montage và các các luồng công việc ngẫu nhiên theo độ trừ mật khác nhau, các tham số băng thông, tốc độ tính toán của máy chủ được thiết lập thống nhất cho tất cả các thực nghiệm. Với mỗi bộ dữ liệu chúng tôi tiến hành chạy chương trình 30 lần độc lập. Kết quả thực nghiệm được trình bày chi tiết trong bảng 1,2 và các hình 1,2, các hình đó đã chỉ ra kết quả so sánh giữa giá trị trung bình tính được bởi thuật toán MODE với các thuật toán đối sánh, trong hầu hết các trường hợp thuật toán MODE đều cho kết quả tốt hơn các thuật toán đối sánh, giá trị trung bình tìm được bởi MODE nhỏ hơn giá trị trung bình tìm được bởi PSO_H từ 8% - 29% và nhỏ hơn giá trị trung bình tìm được bởi thuật toán EGA từ 6% - 25%.

Hình 1,2 cũng so sánh giữa giá trị tốt nhất tìm được bởi thuật toán MODE với các thuật toán đối sánh, qua đó ta thấy giá trị tốt nhất tìm được bởi MODE nhỏ hơn giá trị tốt nhất tìm được bởi PSO_H từ 3% - 19%, chỉ duy nhất trong bộ dữ liệu thực nghiệm thứ 5 thuật toán MODE có giá trị tốt nhất nhỏ hơn giá trị tốt nhất tìm được bởi PSO_H là 0.25%, giá trị tốt nhất tìm được bởi MODE nhỏ hơn giá trị tốt nhất tìm được bởi EGA từ 2% - 20%. Cuối cùng các hình 1,2,3 chỉ ra kết quả so sánh giữa độ lệch chuẩn tìm được bởi thuật toán MODE với các thuật toán đối sánh, giá trị độ lệch chuẩn của MODE luôn nhỏ hơn độ lệch chuẩn của tất cả các thuật toán đối sánh, điều đó chứng tỏ thuật toán MODE có chất lượng lời giải tốt hơn các thuật toán đối sánh và độ ổn định trong các lần chạy cũng tốt hơn.

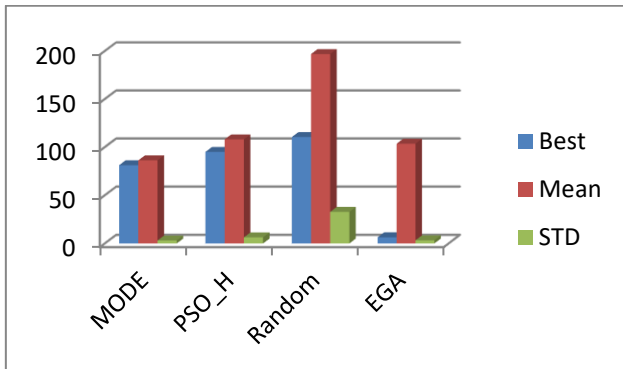
Bảng 1. Kết quả thực hiện các thuật toán với luồng công việc Montage (đơn vị tính: phút)

M	N	α	MODE			PSO_H			RANDOM			EGA		
			Best	Mean	STD	Best	Mean	STD	Best	Mean	STD	Best	Mean	STD

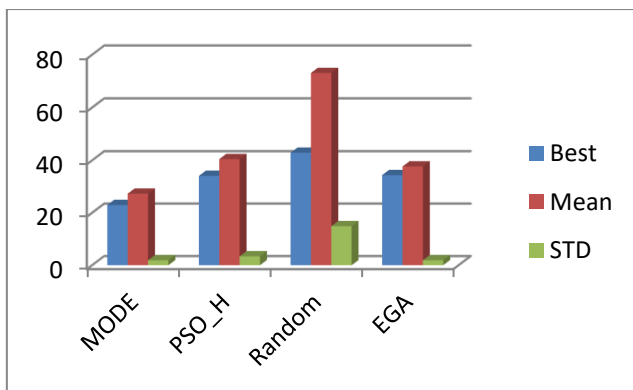
20	8	0.15	29.1	31.0	1.0	35.90	44.2	5.2	56.3	63.3	3.8	35.6	42.0	3.7
20	8	0.31	29.9	31.7	1.2	37.1	44.7	6.1	51.6	67.6	6.8	37.5	42.5	3.2
25	8	0.2	218.0	219.7	1.2	225.0	239.0	8.3	238.0	304.9	33.0	222.4	235.5	4.7
50	8	0.1	81.2	86.3	3.1	95.0	108.0	6.3	110.5	196.8	32.8	96.4	103.5	3.3

Bảng 2. Kết quả thực hiện các thuật toán với luồng công việc Epigenomics (đơn vị tính: giờ)

M	N	α	MODE			PSO_H			RANDOM			EGA		
			Best	Mean	STD	Best	Mean	STD	Best	Mean	STD	Best	Mean	STD
100	10	0.25	38.7	40.9	1.2	38.8	44.9	2.8	49.4	75.7	12.5	39.2	41.9	1.4
100	20	0.12	23.0	27.3	1.9	34.1	40.5	3.4	42.9	73.1	14.9	34.4	37.7	1.9



Hình 1. So sánh các thuật toán với M=50, N=8



Hình 2. So sánh các thuật toán với M=100, N=20

VI. KẾT LUẬN

Lập lịch luồng công việc là một vấn đề phức tạp nhưng rất quan trọng vì nó quyết định hiệu năng của đám mây điện toán. Để đạt được mục tiêu lập lịch là tối thiểu hóa thời gian thực thi, bài báo đã trình bày những kết quả chính sau đây:

- Đề xuất một mô hình lý thuyết cho bài toán lập lịch luồng công việc trong môi trường điện toán đám mây.
- Đề xuất phương pháp tính đối xứng cho các cá thể.

- Dựa trên phương pháp đó, bài báo đề xuất thuật toán MODE hoạt động theo thuật toán tiến hóa vì phân kết hợp với phương pháp đối xứng nhằm tìm ra phương án thực thi tốt nhất.
Tiếp theo chúng tôi dự định cải tiến phương pháp lựa chọn cá thể và cơ chế lai ghép nhằm tìm ra cá thể tốt hơn cho các thế hệ kế tiếp.

TÀI LIỆU THAM KHẢO

- [1] G. B. Berriman, et al, *Montage: A Grid Enabled Engine for Delivering Custom Science-Grade Mosaics On Demand*", in *SPIE Conference*, 2004.
- [2] P. Maechling, et al, *SCEC CyberShake Workflows, Automating Probabilistic Seismic Hazard Analysis Calculations*", Springer, 2006.
- [3] "USC Epigenome Center". <http://epigenome.usc.edu>. [Online]. <http://epigenome.usc.edu>
- [4] LIGO Project. LIGO - Laser Interferometer Gravitational Wave Observatory. [Online]. <http://www.ligo.caltech.edu>.
- [5] R. N. Calheiros, R. Ranjan, A. Beloglazov, Cesar A. F. De Rose, and R. Buyya, *CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms*, Software: Practice and Experience, volume 41, Number 1, Pages: 23-50, Wiley Press, USA, 2011
- [6] J.D. Ullman, *NP-complete scheduling problems*, Journal of Computer and System Sciences, pages 384-393, volume 10, issue 3, 1975
- [7] R. Rajkumar, T. Mala, Achieving Service Level Agreement in Cloud Environment using Job Prioritization in Hierarchical Scheduling, Proceeding of International Conference on Information System Design and Intelligent Application, 2012, vol 132, pp 547-554.
- [8] R. Burya, R. Calheiros, *Modeling and Simulation of Scalable Cloud Environment and the Cloud Sim Toolkit: Challenges and Opportunities*, IEEE publication 2009, pp1-11.
- [9] G. Guo-Ning and H. Ting-Lei, *Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment*, In Proceedings of International Conference on Intelligent Computing and Integrated Systems, 2010, pp. 60-63

- [10] S. Singh, M.Kalra, "Task scheduling optimization of independent tasks in cloud computing using enhanced genetic algorithm", *International Journal of Application or Innovation in Engineering & Management*, V.3, Issue 7, 2014.
- [11] S. Pandey, L. Wu¹, S. M. Guru, R. Buyya¹, *A Particle Swarm Optimization (PSO)-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments*, Proc. of 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pages 400-407, 2010
- [12] R. Rajkumar, T. Mala, *Achieving Service Level Agreement in Cloud Environment using Job Prioritization in Hierarchical Scheduling*, Proceeding of International Conference on Information System Design and Intelligent Application, 2012, vol 132, pp 547-554.
- [13] Q. Cao, W. Gong and Z. Wei, *An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing*, In Proceedings of Third International Conference on Bioinformatics and Biomedical Engineering, 2009, pp.1-3
- [14] S.Selvi, Dr. D.Manimegalai, Dr.A.Suruliandi, *Efficient Job Scheduling on Computational Grid with Differential Evolution Algorithm*, , International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April, 2011
- [15] Q. XU, L.WANG, HE. Baomin, N.WANG, *Modified Opposition-Based Differential Evolution for Function Optimization*, Journal of Computational Information Systems, pp. 1582-1591, 2011.
- [16] O. Sinnens, *Task Scheduling for Parallel Systems*, John Wiley & Sons, 2007, pp. 83
- [17] R. Storn and K. Price, *Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*, Journal of Global Optimization, 1997, pp. 341-359.
- [18] M. Mitzenmacher, E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press (2005)
- [19] J. V. Vliet, F. Paganelli, *Programming Amazon EC2*, O'Reilly Media, ISBN 1449393683, 2011

Nhận bài ngày:

SƠ LƯỢC VỀ TÁC GIẢ

PHAN THANH TOÀN



Sinh năm 1974 tại Thái Nguyên.

Tốt nghiệp đại học và thạc sĩ tại trường đại học Bách Khoa Hà Nội, nghiên cứu sinh năm 2012

tại học viện Khoa học công nghệ quân sự.

Hiện đang công tác tại trường đại học Sư Phạm Hà Nội

Lĩnh vực nghiên cứu: các phương pháp gần đúng giải bài toán lập lịch luồng công việc trong môi trường điện toán đám mây, xử lý song song và phân tán.

Điện thoại : 0912.069.762

E-mail: pttoan@hnue.edu.vn

NGUYỄN DOÃN CƯỜNG



Tốt nghiệp đại học 1989 1993 khoa Toán Tin đại học Sư Phạm Hà Nội, Tốt nghiệp thạc sĩ CNTT tại đại học Bách Khoa Hà Nội, tiến sỹ 2004 - 2007 viện nghiên cứu khoa học công nghệ Nhật Bản JAIST.

Lĩnh vực nghiên cứu hiện nay: mạng máy tính và truyền thông, xử lý song song và phân tán

Điện thoại : 0988.765.837

E-mail: locnt@hnue.edu.vn

NGUYỄN DOÃN CƯỜNG



Sinh năm 1977 tại Tuyên Quang.

Tốt nghiệp đại học tại học viện kỹ thuật quân sự, nghiên cứu sinh tại Trường Đại học Tổng hợp Kỹ thuật điện Quốc gia Saint-Peterburg - CHLB Nga 2006.

Hiện đang công tác tại : viên CNTT-viên Khoa học công nghệ quân sự.

Lĩnh vực công tác và hướng nghiên cứu: Công nghệ phần mềm, Data Mining and Knowledge Discovery, cơ sở dữ liệu lớn, cơ sở dữ liệu chuỗi thời gian, Bảo mật.

Điện thoại : 0976.210.686

E-mail: cuongvncntt@yahoo.com