

# MODE: HƯỚNG TIẾP CẬN MỚI CHO VIỆC THỰC THI LUỒNG CÔNG VIỆC

Phan Thanh Toàn<sup>1</sup>, Nguyễn Thế Lộc<sup>2</sup>, Nguyễn Doãn Cường<sup>3</sup>, Trần Đăng Hưng<sup>2</sup>

<sup>1</sup>Khoa Sư Phạm Kỹ Thuật, Trường Đại Học Sư Phạm Hà Nội

<sup>2</sup>Khoa Công Nghệ Thông Tin, Trường Đại Học Sư Phạm Hà Nội

<sup>3</sup>Viện công nghệ thông tin, Viện Khoa Học và Công Nghệ Quân Sự

**Tóm tắt:** Lập lịch luồng công việc là một vấn đề quan trọng trong thời đại điện toán đám mây. Về cơ bản vấn đề này liên quan đến việc tìm kiếm tài nguyên và phân bổ các tác vụ dựa trên tài nguyên phù hợp. Lập lịch luồng công việc đóng một vai trò quan trọng trong quản lý hệ thống. Việc lập lịch hợp lý có ảnh hưởng đáng kể lên hiệu năng của đám mây. Bài báo này đề xuất một kỹ thuật lập lịch mới gọi là MODE chạy trong môi trường điện toán đám mây. Giải thuật này được xây dựng dựa trên việc nghiên cứu chi tiết và phân tích của quá trình tiến hóa khác biệt và áp dụng ưu điểm của tiến hóa khác biệt và loại trừ nhược điểm của nó. Chúng tôi đã đề xuất một giải thuật tiến hóa khác dựa trên Modified Opposition để lập lịch phân luồng nhiệm vụ trong môi trường điện toán đám mây để thời gian thực thi là nhỏ nhất.

**Từ khóa:** Workflow scheduling, Opposition-Based Differential Evolution, cloud computing, Differential Evolution.

## I. GIỚI THIỆU

Điện toán đám mây là môi trường phân tán không đồng nhất với sự liên kết của rất nhiều máy chủ ảo và vật lý trên môi trường mạng. Các tài nguyên phần cứng, phần mềm được cung cấp một cách linh động theo nhu cầu của người dùng. Luồng công việc (workflow) là một chuỗi có thứ tự các tác vụ (task) có thể được thực hiện đồng thời hay

tuần tự, dữ liệu ra của tác vụ này là dữ liệu vào của tác vụ kế tiếp. Rất nhiều ứng dụng trong các lĩnh vực khoa học đều yêu cầu phải xử lý một lượng lớn dữ liệu dưới dạng luồng công việc, dẫn tới nhu cầu lập lịch thực thi luồng công việc sao cho hiệu quả nhất. Trong môi trường điện toán đám mây bản chất của vấn đề này là tìm phương án ánh xạ những tác vụ của luồng công việc tới các máy chủ sao cho thời gian xử lý toàn bộ luồng công việc 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-đầy đủ. Phần 4 giới thiệu thuật toán đề xuất - MODE.

## II. NHỮNG CÔNG TRÌNH LIÊN QUAN

Bài toán lập lịch luồng công việc đã được chứng minh là thuộc lớp NP-đầy đủ [2] nghĩa là thời gian để tìm ra lời giải tối ưu là rất lớn, 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 [3]. 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 [4]

Tác giả liên lạc: Phan Thanh Toàn,  
email: ptoan@hnue.edu.vn

Đến tòa soạn: 14/3/2016, chỉnh sửa: 28/4/2016, chấp  
nhận đăng: 30/5/2016.

- 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 [5], 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.

L. Guo đã trình bày một mô hình cho bài toán lập lịch luồng công việc trong môi trường điện toán đám mây [6] và đề xuất một thuật toán lập lịch luồng công việc dựa trên chiến lược tối ưu bày đàn, kết hợp với luật SPV (Smallest Position Value) nhằm rời rạc hóa các giá trị thực của véc tơ dịch chuyển và véc tơ vị trí của các cá thể trong quần thể. Tác giả đã thực nghiệm và chỉ ra thuật toán đề xuất có chất lượng lời giải tốt hơn các thuật toán CM-PSO (Crossover and Mutation PSO) và L-PSO (Local search Particle Swarm Optimization). S. Pandey [7] 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 bày đàn. R. Rajkumar đã đề xuất một thuật toán lập lịch phân cấp [8] 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) [9]. 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) [10], trong công trình tác giả đã vận dụng thuật toán tiến hóa vi phân (Differential Evolution - 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 [11] (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.

### III. MÔ HÌNH LÝ THUYẾT

Biểu diễn luồng công việc bởi đồ thị  $G=(V,E)$ , với

- $V$  là tập đỉnh, mỗi đỉnh biểu diễn cho một tác vụ trong luồng công việc.
- $T = \{T_1, T_2, \dots, T_M\}$  là tập  $M$  tác vụ
- $E$  biểu diễn sự phụ thuộc dữ liệu giữa các tác vụ. Cạnh  $(T_i, T_j) \in E$  thì tác vụ  $T_i$  là tác vụ cha của tác vụ  $T_j$ , dữ liệu sinh ra bởi  $T_i$  sẽ được sử dụng bởi  $T_j$ .
- Tập máy chủ  $S = \{S_1, S_2, \dots, S_N\}$ .  $N$  là số máy chủ.
- Mỗi tác vụ  $T_i$  có thể thực hiện tại một máy chủ bất kỳ  $S_j \in S$
- Khối lượng tính toán của tác vụ  $T_i$  kí hiệu là  $W_i$  (flop-floating point operations).
- $P(S_j)$ : là tốc độ tính toán của máy chủ  $S_j$  (MI/s : million instructions/second).
- Bảng thông  $B(S_i, S_j)$  giữa máy chủ  $S_i$  và  $S_j$  được biểu diễn bởi hàm  $B(): S \times S \rightarrow R^+$ . Bảng thông thỏa mãn các điều kiện:  $B(S_i, S_i) = \infty$  and  $B(S_i, S_j) = B(S_j, S_i)$  (Đơn vị đo là Mbps)
- $D_{ij}$ : khối lượng dữ liệu truyền từ tác vụ  $T_i$  đến tác vụ  $T_j$ . (Đơn vị đo là Mbit)

Mỗi phương pháp lập lịch được biểu diễn bởi hàm  $f(): T \rightarrow S$ ;  $f(T_i)$  là máy chủ thực thi tác vụ  $T_i$ .

Từ các giả thiết trên ta có :

- Thời gian thực hiện tác vụ  $T_i$  là :

$$\frac{W_i}{P(f(T_i))} \quad (1)$$

- Thời gian truyền dữ liệu giữa tác vụ  $T_i$  và  $T_j$  là

$$\frac{D_{ij}}{B(f(T_i), f(T_j))} \quad (2)$$

### Phát biểu bài toán.

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 thực thi luồng công việc:*

$$makespan \rightarrow \min$$

trong đó thời gian thực thi (ký hiệu là makespan) là đại lượng được tính từ lúc tác vụ đầu tiên bắt đầu được khởi động cho tới khi tác vụ cuối cùng được hoàn thành.

**Bổ đề:** WSP là bài toán thuộc lớp NP-Complete

### Chứng minh:

Xét bài toán SCHED [12] 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-đầy đủ [12]. 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:

$$P(S_i) = P(S_j) \quad \forall i, j = 1, \dots, N.$$

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

## 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 [13], 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)

### A. 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  $x_1 = (1, 2, 1, 3, 2)$  được biểu diễn như sau:

| $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ |
|-------|-------|-------|-------|-------|
| $S_1$ | $S_2$ | $S_1$ | $S_3$ | $S_2$ |

### B. 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

$$\bar{x}_i \text{ và } \bar{x}_i = (\bar{S}_{i\pi(1)}, \bar{S}_{i\pi(2)}, \dots, \bar{S}_{i\pi(M)})$$

$$\bar{S}_{i\pi(j)} = a + b - S_{i\pi(j)}; \forall j = 1, 2, \dots, M \quad (5)$$

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:

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

#### 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.  $\bar{x}_i \leftarrow \text{opposite}(x_i)$ ;
- $\{\bar{x}_i$  được tính 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
5. return OP

### C. 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:

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

Trong đó:  $1.0 \leq SP \leq 2.0$ ; 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.  $\text{pos}[i] \leftarrow \text{PopSize} - 1$
4. for  $i=1$  to PopSize do
5. calculate  $\text{rank}_i$  by equation (7)
6.  $\text{rand} \leftarrow \text{Random}(0, SP)$
7.  $s \leftarrow \text{PopSize}$
8. while( $\text{rank}[s] < \text{rand} \ \&\& \ s > 0$ )  $s = s - 1$
- return  $x_s$  End.

### 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  $W[1 \times M]$ ,  $P[1 \times N]$ ,  $B[N \times N]$ ,  $D[M \times M]$ , the constant K, the deviation  $\varepsilon$ , 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  $\leftarrow$  OP\_Algorithm ; tính quần thể đối xứng của quần thể hiện tại
3. Chọn PopSize cá thể tốt nhất từ  $P \cup OP$
4. while(Điều kiện lặp)do
5. for  $i=1$  to PopSize do
6. Lựa chọn cá thể  $p_1$  theo thuật toán RBRWS
7. Lựa chọn cá thể  $p_2$  theo thuật toán RBRWS
8.  $F \leftarrow \text{Random}(1, 0)$
9.  $v_i \leftarrow pbest + F \times (p_1 - p_2)$
10. Gán số hiệu máy chủ cho mỗi vị trí  $j$  của véc tơ  $v_i$  theo (6)
11.  $\text{rand}_{i,j} \leftarrow \text{Random}(0, 1)$
12.  $I_{\text{rand}} \leftarrow \text{random}(1, M)$
13.  $u_{i,j} = \begin{cases} v_{i,j} & \text{nếu } \text{rand}_{i,j} \leq CR \text{ hoặc } i = I_{\text{rand}} \\ x_{i,j} & \text{nếu } \text{rand}_{i,j} \geq CR \text{ hoặc } i \neq I_{\text{rand}} \end{cases}$
14. if ( $\text{makespan}(u_i) < \text{makespan}(x_i)$ )
15.  $x_i \leftarrow u_i$
16. end if
17. end for
18.  $\text{rand} \leftarrow \text{Random}(0, 1)$
19. if( $\text{rand} < J_r$ )
20. OP  $\leftarrow$  OP\_Algorithm
21. Lựa chọn PopSize cá thể tốt nhất từ  $P \cup OP$
22. end if
23. End while
24. 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  $P \cup OP$ .

Trong mỗi bước lặp chọn ra hai cá thể  $p_1, p_2$  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:  $v_i \leftarrow pbest + F \times (p_1 - p_2)$

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  $rand_{i,j} \in [0,1]$ ,  $I_{rand} \in [1,M]$ ,  $CR \in [0,1]$ .

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

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

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  $P \cup OP$ .

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 [1] 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 [15].

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.

### A. 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 [16].
- 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 [17].
- Dựa theo tính chất của môi trường điện toán đám mây, đây là một môi trường tính toán không đồng nhất, năng lực 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 năng lực 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  $\alpha$  khác nhau. Chi tiết các nhóm dữ liệu như sau theo số lượng máy chủ N, số tác vụ M và hệ số  $\alpha$  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  $\alpha$  và tính bởi công thức:

### B. Tham số cấu hình

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

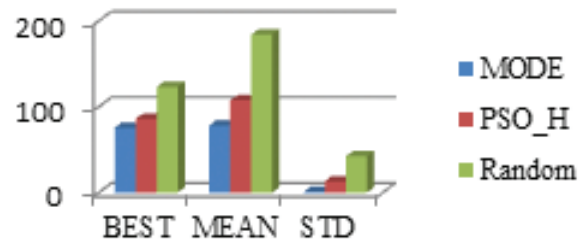
- Tốc độ tính toán P của các máy chủ: từ 1 đến 250 (million instructions/s); khối lượng dữ liệu D giữa các tác vụ: từ 1 đến 10000 (Mega bit); băng thông giữa các máy chủ B: từ 10 đến 100 (Mega bit/s).
- Hệ số vi sai  $F=0.5$ ; hệ số trao đổi chéo  $CR=0.9$ ;  $J_r=0.3$ ;  $PopSize=50$ ;  $\alpha$ : từ 0.2 tới 0.7.

### C. Quá trình tiến hành thực nghiệm

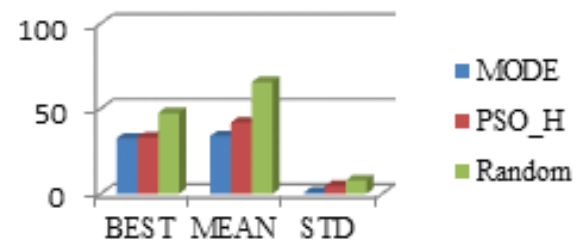
Để đánh giá hiệu năng của thuật toán đề xuất MODE chúng tôi đã tiến hành thực nghiệm và so sánh kết quả của MODE với các thuật toán tiến hóa mạnh hiện nay như PSO\_H và Random. Dữ liệu thực nghiệm được lấy từ ứng dụng Montage và các các luồng công việc ngẫu nhiên theo độ trừ mật khác nhau, các tham số bằng 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 I và các Hình 1, 2, 3, 4, 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 tốt hơn giá trị trung bình tìm được bởi PSO\_H từ 6% - 88% và tốt hơn giá trị trung bình tìm được bởi thuật toán Random từ 39% - 85%.

Hình 1, 2, 3, 4 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 tốt hơn giá trị tốt nhất tìm được bởi PSO\_H từ 2% - 19%, chỉ duy nhất trong bộ dữ liệu thực nghiệm thứ nhất thuật toán MODE có giá trị tốt nhất lớn hơn giá trị tốt nhất tìm được bởi PSO\_H là 3%, giá trị tốt nhất tìm được bởi MODE tốt hơn giá trị tốt nhất tìm được bởi Random từ 20% - 40%. Cuối cùng các Hình 1, 2, 3, 4 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 tốt hơn PSO\_H từ 6% - 88% và tốt hơn Random từ 82% -

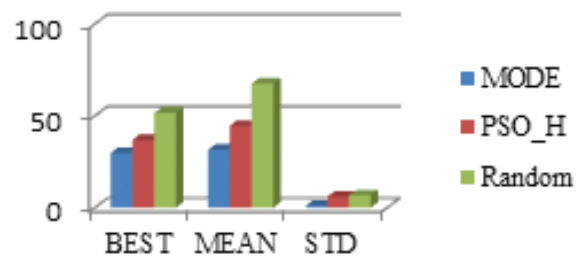
97%, điều đó chứng tỏ thuật toán MODE có chất lượng lời giải tốt hơn các thuật toán đối sánh và độ ổn định trong các lần chạy cũng tốt hơn.



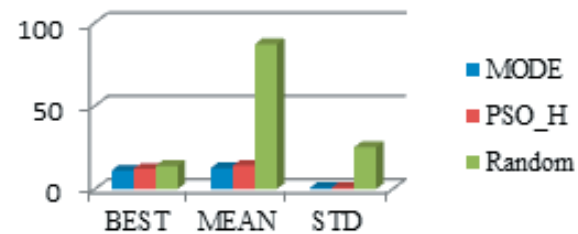
Hình 1. So sánh các thuật toán với M=10, N=5



Hình 2. So sánh các thuật toán với M=20, N=3



Hình 3. So sánh các thuật toán với M=20, N=8



Bảng I. Kết quả thực hiện các thuật toán

| M  | N | $\alpha$ | MODE |      |      | PSO_H |       |      | RANDOM |       |      |
|----|---|----------|------|------|------|-------|-------|------|--------|-------|------|
|    |   |          | Best | Mean | STD  | Best  | Mean  | STD  | Best   | Mean  | STD  |
| 10 | 3 | 0.26     | 16.9 | 17.3 | 0.4z | 16.4  | 20.4  | 2.4  | 21.4   | 28.6  | 3.2  |
| 10 | 5 | 0.26     | 75.5 | 78.2 | 0.9  | 86.0  | 107.5 | 13.2 | 123.3  | 184.1 | 42.4 |
| 20 | 5 | 0.15     | 27.5 | 29.3 | 1.4  | 29.6  | 41.0  | 5.0  | 45.8   | 59.0  | 6.8  |
| 20 | 3 | 0.31     | 32.5 | 33.9 | 0.8  | 33.2  | 41.9  | 4.6  | 47.4   | 65.6  | 7.8  |
| 20 | 8 | 0.31     | 29.9 | 31.7 | 1.2  | 37.1  | 44.7  | 6.1  | 51.6   | 67.6  | 6.8  |
| 50 | 8 | 0.3      | 11.1 | 12.6 | 0.8  | 12.1  | 14.0  | 0.9  | 13.9   | 87.1  | 25.2 |

## 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 vi 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]. 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*, vol. 41, no. 1, pp. 23-50, 2011
- [2]. J.D. Ullman, NP-complete scheduling problems, *Journal of Computer and System Sciences*, vol. 10, no. 3, pp.384-393, 1975
- [3]. Dr. Sudha Sadhasivam, R. Jayarani, Dr. N. Nagaveni, R. Vasanth Ram, Design and Implementation of an efficient Twolevel Scheduler for Cloud Computing Environment, *In Proceedings of International Conference on Advances in Recent Technologies in Communication and Computing*, 2009
- [4]. R. Burya, R. Calheiros, Modeling and Simulation of Scalable Cloud Environment and the Cloud Sim Toolkit: Challenges and Opportunities, IEEE publication 2009, pp1-
- [5]. 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*, pp. 60-63, 2010.
- [6]. L. Guo, Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm, *Journal of networks*, vol. 7, no. 3, pp. 547-552, 2012.
- [7]. 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)*, pp. 400-407, 2010.
- [8]. 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*, vol 132, pp 547-554, 2012.
- [9]. 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*, pp.1-3, 2009.
- [10]. 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.
- [11]. 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.

- [12]. O. Sinnén, *Task Scheduling for Parallel Systems*, John Wiley & Sons, pp. 83, 2007.
- [13]. R. Storn and K. Price, Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, pp. 341-359, 1997.
- [14]. J. Kennedy, R.C. Eberhart, Particle swarm optimization, *Proc. of IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.
- [15]. M. Mitzenmacher, E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press (2005).
- [16]. J. V. Vliet, F. Paganelli, *Programming Amazon EC2*, O'Reilly Media, 2011.
- [17]. <http://montage.ipac.caltech.edu>.

#### MODE: A NOVEL APPROACH FOR EXECUTING DATA WORKFLOW

**Abstract:** Workflow scheduling is a big issue in the era of Cloud computing. Basically it is the issue related to the discovering resources and allocating tasks on suitable resources. Workflow scheduling plays a vital role in the system management. Proper scheduling can have significant impact on the performance of the Cloud. In this paper, a novel workflow scheduling called MODE, running on the cloud computing environments, is proposed. The algorithm is built through a comprehensive study and analysis of Differential Evolution strategy with applying the advantages of the Differential Evolution and covers its disadvantages. We propose a Modified Opposition-Based Differential Evolution algorithm for scheduling workflow tasks in the cloud environments so that the execution time of the workflow (makespan) is minimized.



#### 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



#### Nguyễn Thế Lộc

Sinh năm 1972, tại Hà Nội.

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 sĩ 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](mailto:locnt@hnue.edu.vn)



#### Nguyễn Doãn Cường

Sinh năm 1977 tại Tuyên Quang.

Tốt nghiệp đại học tại học viện kĩ 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 : viện CNTT-viện Khoa học công nghệ quân sự

Điện thoại : 0976.210.686

E-mail : [cuongvncntt@yahoo.com](mailto:cuongvncntt@yahoo.com)

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.



#### Trần Đăng Hưng

Sinh năm 1979, tại Hà Tĩnh. Tốt nghiệp Khoa Toán-Tin, Trường ĐH Sư Phạm Hà Nội năm 2001; tốt nghiệp Thạc sĩ tại Trường ĐH Công nghệ, ĐHQGHN năm 2006. Tốt nghiệp Tiến sĩ tại Viện Khoa học và Công nghệ tiên tiến Nhật Bản năm 2009.

Lĩnh vực nghiên cứu: Khai phá dữ liệu, Học máy, Tin-sinh học.

Điện thoại: 0904 68 72 82

Email: [hungtd@hnue.edu.vn](mailto:hungtd@hnue.edu.vn)