Thuật toán lập lịch luồng công việc trong môi trường điện toán đám mây dựa trên chiến lược tối ưu bày đàn

A Particle Swarm Optimization-Based Workflow Scheduling Algorithm in the Cloud Environment

Phan Thanh Toàn, Nguyễn Thế Lộc, Nguyễn Doãn Cường, Đỗ Như Long

Abstract: *Workflow is the series of tasks that are necessary to complete a goal. Workflow scheduling, the most important problem which the cloud controllers deal with, focuses on mapping and managing the execution of tasks on servers so that the expenses is the minimum. In this paper, we build a workflow scheduling framework which run on the cloud computing environments. In order to solve the mentioned problem, we propose a PSO-based algorithm for scheduling workflow tasks in the cloud environments so that the total cost is minimized.*

Keywords: *workflow scheduling, workflow applications, cloud computing.*

I. GIỚI THIỆU

Điện toán đám mây là sự tích hợp của nhiều công nghệ thuộc lĩnh vực công nghệ thông tin và truyền thông, môi trường điện toán đám mây cho phép người sử dụng truy cập một cách thuận tiện, nhanh chóng đến các tài nguyên tính toán (máy chủ, thiết bị lưu trữ, các ứng dụng, các dịch vụ…), điện toán đám mây là một mô hình phân tán không đồng nhất với sự tập hợp của nhiều máy tính làm việc trên môi trường mạng internet nhằm tận dụng sức mạnh chung của hệ thống trong các ứng dụng lớn. Một trong số các ứng dụng phổ biến nhất trong môi trường điện toán đám mây là bài toán luồng công việc (từ đây viết tắt là workflow), hiệu năng của các trung tâm điện toán phụ thuộc rất nhiều vào việc sắp xếp các tác vụ trong luồng để thực thi trên các máy tính trong môi trường đám mây nhằm hoàn thành luồng công việc một cách "tối ưu" nhất.

Nội dung của bài báo gồm những phần chính sau đây. Phần I giới thiệu bối cảnh thực tế tại trung tâm điện toán đám mây nơi cung cấp dịch vụ workflow. Phần II trình bày một số công trình liên quan và các hạn chế, Phần III phát biểu bài toán và xây dựng mô hình toán học bài toán tối thiểu chi phí thực thi luồng công việc trong môi trường điện toán đám mây. Phần IV giới thiệu thuật toán đề xuất. Phần V để kiểm chứng hiệu năng của thuật toán đề xuất, chúng tôi đã thực hiện các thực nghiệm trên những ứng dụng workflow trong môi trường đám mây thông qua công cụ mô phỏng CloudSim [1]. Các kết quả được thu thập và so sánh với giải thuật PSO Heuristic [2] và 2 giải thuật lập lịch cơ bản là giải thuật Random [3,4] và Round Robin [5].

II. NHỮNG CÔNG TRÌNH LIÊN QUAN

Bài toán lập lịch luồng công việc đã được chứng minh là thuộc lớp NP-đầy đủ [6] nghĩa là thời gian để tìm ra lời giải tối ưu là rất lớn, vì vậy đã có nhiều giải thuật metaheuristic được nghiên cứu nhằm tìm ra lời giải gần đúng trong thời gian ngắn. S. Parsa [7] đã đề xuất một thuật toán lập lịch nhằm tối thiểu thời gian thực thi trong môi trường lưới tính toán Grid. J.M. Cope và đồng nghiệp đã phân tích hiệu năng của giải thuật FRMTL và FRMAS [8] trong môi trường lưới tính toán TeraGrid. A. Agarwal đã đề xuất thuật toán

tham lam [9] trong đó mỗi tác vụ được gán một thứ tự ưu tiên dựa vào khối lượng công việc của tác vụ, mỗi máy chủ cũng được gán một thứ tự ưu tiên theo tốc độ xử lý của máy chủ sau đó gán các tác vụ vào các máy chủ theo các thứ tự ưu tiên đã tính toán. Cách làm này có nhược điểm là khiến những tác vụ có mức ưu tiên thấp phải chờ đợi lâu và bỏ qua yếu tố tốc độ truyền dữ liệu giữa các máy chủ trong đám mây.

Một số tác giả khác như M.Wieczorek [10] đã nghiên cứu và đề xuất thuật toán lập lịch thực thi luồng công việc theo thuật toán di truyền (Genetic Algorithm - GA), tuy nhiên các nghiên cứu [11,12] đã nhận định rằng phương pháp PSO (Particle Swarm Optimization - Tối ưu bày đàn) có ưu thế hơn so với phương pháp GA khi giải bài toán lập lịch luồng công việc trong những môi trường tính toán phân tán như Lưới (Grid Computing) hay Đám mây (Cloud Computing). Theo hướng đó, S. Pandey [12] đã đề xuất thuật toán theo phương pháp PSO nhằm cực tiểu hóa chi phí thực thi luồng công việc. Thay vì tìm phương án có tổng chi phí thực thi tại các máy chủ là bé nhất, S. Pandey lại định nghĩa hàm mục tiêu để tìm phương án có chi phí thực thi của máy chủ tốn kém nhất (máy có tổng chi phí lớn hơn mọi máy khác) là nhỏ nhất so với các phương án khác. Cách làm này có xu hướng "cào bằng" nghĩa là thiên về các lời giải có chi phí thực thi của các máy chủ là xấp xỉ nhau. Chúng tôi nhận thấy, qua lý thuyết và các thực nghiệm kiểm chứng, cách làm này thường khiến chương trình sớm hội tụ về những giá trị cực tiểu địa phương thay vì tìm ra cực trị toàn cục.

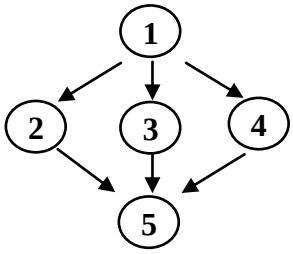
III. BÀI TOÁN TỐI THIỂU HÓA CHI PHÍ

Chúng tôi phát biểu bài toán như sau: giả sử cần sắp xếp lịch biểu cho một luồng công việc gồm m tác vụ (task), trong một môi trường điện toán đám mây với các yêu cầu như sau :

- Luồng công việc gồm có M tác vụ
- Các tác vụ có quan hệ thứ tự trước sau
- Mỗi tác vụ cần một khối lượng dữ liệu vào và sẽ xuất ra một lượng dữ liệu xác định
- Luồng công việc chỉ có duy nhất một tác vụ gốc
- Môi trường thực thi luồng công việc là môi trường điện toán đám mây với N máy chủ tính toán và K máy chủ lưu trữ
- Mỗi tác vụ có thể được thực thi trên một máy chủ bất kì và chỉ được thực thi trên một máy duy nhất
- Chi phí của mỗi máy chủ thực thi và máy chủ lưu trữ dữ liệu đều tính theo một đơn giá do nhà cung cấp dịch vụ ấn định

Ký hiệu

- Tập các máy chủ $S = \{S_1, S_2,\ldots,S_N\}$
- Luồng công việc được biểu diễn bởi một đồ thị $G=(V, E)$, với $V =\{T_1, T_2,\ldots,T_M\}$ là tập các đỉnh, mỗi đỉnh biểu thị một tác vụ, E là tập cạnh thể hiện mối quan hệ giữa các tác vụ. Cạnh $e =(T_i, T_j) \in E$ có nghĩa là tác vụ T_i là cha của tác vụ T_j, nó sẽ được thực hiện trước, sau đó chuyển cho tác vụ T_j một khối lượng dữ liệu làm đầu vào cho tác vụ T_j



Hình 1: Đồ thị luồng công việc với 5 tác vụ

- Khối lượng tính toán (Workload) của T_i kí hiệu là W_i, đơn vị đo là flop (*floating point operations*- phép tính trên số thực dấu phảy động)

- Tốc độ tính toán (đơn vị flop/s) của máy S_i được ký hiệu bởi $P(S_i)$

- Đơn giá cước tính toán (đơn vị dolar/flop) của máy S_i được ký hiệu là $E(S_i)$

- Mỗi cặp máy chủ S_i, S_j ($1 \le i,j \le N$) đều có một kênh truyền kết nối chúng, băng thông của kênh truyền kí hiệu là B và là một hàm số:

$$B : SxS \rightarrow R^+$$

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

Hàm B thỏa mãn các điều kiện sau:

- $B(S_i, S_i) = \infty$ (chi phí truyền tại chỗ bằng 0)
- $B(S_i, S_k) + B(S_k, S_j) \leq B(S_i, S_j)$
- $B(S_i, S_j) = B(S_j, S_i)$

Tập giá trị của hàm băng thông B( ) giữa các máy chủ được cung cấp bởi nhà cung cấp dịch vụ và được biểu diễn dưới dạng Bảng 1

Bảng 1. *Băng thông giữa các máy chủ*

B	***1***	***2***	***...***	***N***
1	B(1,1)= ∞	B(1,2)	...	B(1,N)
2	B(2,1)	B(2,2)= ∞	...	...
...	...	...	...	...
N	B(N,1)	B(N,2)	...	B(N,N)= ∞

- Khối lượng dữ liệu truyền từ T_i tới T_k được kí hiệu là D_{ik} được cho bởi Bảng 2.

Bảng 2. *Khối lượng dữ liệu giữa các tác vụ*

D	***1***	***2***	***...***	***M***
1	D_{11}	D_{12}	...	D_{1N}
2	D_{21}	$D_{22}= \infty$	...	...
...	...	...	...	...
M	D_{N1}	D_{N2}	...	$D_{NN} = \infty$

- Đơn giá cước truyền thông (đơn vị là dolar/bit) giữa 2 máy được kí hiệu là C và là một hàm số

$C : SxS \rightarrow R^+$

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

Hàm C( ) thỏa mãn điều kiện: $C(S_i, S_k) = C(S_k, S_i)$

- Mỗi phương án xếp lịch thực thi luồng công việc là một ánh xạ f

$f : T \rightarrow S$

$T_i \rightarrow f(T_i)$

$f(T_i)$ là máy chủ chịu trách nhiệm thực thi tác vụ T_i. Từ đó suy ra:

- Thời gian tính toán của tác vụ T_i là: $\dfrac{W_i}{P(f(T_i))}$ (1)

  (i=1,2, ... M)
- Chi phí tính toán của tác vụ T_i là: $W_i \times E(f(T_i))$ (2)
- Thời gian truyền dữ liệu giữa tác vụ T_i và tác vụ con T_k là

$$\frac{D_{ik}}{B(f(T_i), f(T_k))} \quad (3)$$

- Chi phí truyền thông giữa tác vụ T_i và tác vụ con T_k là: $D_{ik} \times C(f(T_i), f(T_k))$ (4)

Chi phí thực thi T_i trên máy chủ $f(T_i)$ bằng chi phí tính toán cộng với tổng chi phí truyền thông giữa các tác vụ T_j với T_i trong đó các tác vụ T_j là các tác vụ cha của tác vụ T_i. Chi phí thực thi toàn bộ luồng công việc là tổng chi phí thực thi tất cả các tác vụ trong luồng.

Chúng ta đặt hàm mục tiêu là:

$$\sum_{i=1}^{M}(W_i \times E(f(T_i))) + \sum_{i=1}^{M}\sum_{k=1}^{M}(D_{ik} \times C(f(T_i), f(T_k))) \rightarrow Min \quad (5)$$

V. THUẬT TOÁN ĐỀ XUẤT

Dựa trên mô hình toán học đã đề xuất ở mục III nêu trên, chúng tôi đề xuất một giải thuật tìm kiếm theo kiểu bày đàn PSO. PSO là chiến lược tìm kiếm tiến hóa đề xuất bởi R. Eberhart và J. Kennedy [13], trong đó mỗi cá thể luôn có xu hướng dịch chuyển tới vị trí tốt hơn dựa vào kinh nghiệm của cá nhân và kinh nghiệm của cả quần thể. Giải thuật Scheduling_Heuristic của chúng tôi được mô tả chi tiết như sau.

1) Trước hết chúng ta biểu diễn mỗi cá thể trong quần thể bởi 2 thành phần cơ bản là véc tơ vị trí và véc tơ dịch chuyển. Véc tơ vị trí có số chiều bằng số tác vụ trong luồng công việc và được mô tả như một cấu trúc dữ liệu bảng băm:

Hashtable<Task, Integer> position

Véc tơ dịch chuyển cũng được biểu diễn như một cấu trúc dữ liệu bảng băm:

Hashtable<Task, Double> velocity

Ví dụ: xét luồng công việc gồm 5 tác vụ T1, T2, T3, T4, T5 và 3 máy chủ PC1, PC2, PC3 khi đó một cá thể được mã hóa như sau :

Bảng 3. *Minh họa cách mã hóa cá thể*

T_1	T_2	T_3	T_4	T_5
PC_1	PC_2	PC_1	PC_3	PC_2

Nghĩa là tác vụ T1, T3 được thực hiện bởi PC1, còn tác vụ T2, T5 được thực hiện bởi PC2 và tác vụ T4 thực hiện trên PC3.

2) Mã hóa tác vụ: mỗi tác vụ được xác định bởi 3 đại lượng: (i) số lệnh cần thực hiện (ii) kích thước dữ liệu đầu ra của tác vụ và (iii) danh sách các tác vụ phụ thuộc, danh sách các tác vụ này được biểu diễn như một cấu trúc danh sách ArrayList

3) Biểu diễn các dữ liệu về chi chí thực thi các tác vụ trên các máy chủ, chi phí truyền thông giữa các máy chủ và khối lượng dữ liệu vào/ra giữa các tác vụ

- Chi phí thực thi các tác vụ trên các máy chủ được biểu diễn như một ma trận và ta sử dụng cấu trúc bảng băm như sau để lưu trữ chi phí thực thi một tác vụ trên một máy chủ.

Hashtable<Task,Hashtable<Host,Double>>
TH_matrix;

Bảng 4*. Chi phí thực thi T_i trên máy PC_j*

		PC1	PC2	PC3
TP[5 x3]	T_1	1.23	1.12	1.15
	T_2	1.17	1.17	1.28
	T_3	1.13	1.11	1.11
	T_4	1.26	1.12	1.14
	T_5	1.19	1.14	1.22

- Chi phí truyền thông giữa các PC được biểu diễn như một ma trận và ta cũng sử dụng một cấu trúc bảng băm như sau để lưu trữ :

Hashtable<Host,Hashtable<Host,Double>>
HH_matrix;

Bảng 5*. Chi phí truyền thông giữa các PCj*

		PC1	PC2	PC3
PP[3x3]	PC_1	0	0.17	0.21
	PC_2	0.17	0	0.22
	PC_3	0.21	0.22	0

Trong đó:

PP[i,j]=chi phí truyền thông giữa PC_i và PC_j (đơn vị dolar/Mb)

- Dữ liệu vào/ra giữa các tác vụ trong luồng công việc được biểu diễn bởi một ma trận và ta sử dụng cấu trúc dữ liệu bảng băm như sau để lưu trữ :

Hashtable<Task, Hashtable<Task,Double>>
edge_weight

Bảng 6. *Kích thước input/output của task i*

		total data			total data
$DS_{T2,T3,T4}$ [2x2]	Input	10	DS_{T5} [2x2]	Input	30
	Output	10		Output	60

Trong đó Input là dữ liệu vào, Output là dữ liệu ra của các tác vụ (đơn vị MB)

Thuật toán đề xuất như sau:

Input:
- Luồng công việc gồm n tác vụ
- Chi phí thực thi các tác vụ trên các máy chủ (bảng 4)
- Chi phí truyền thông giữa các máy chủ (bảng 5)
- Khối lượng dữ liệu vào/ra giữa các tác vụ (bảng 6)

Output: phương án lập lịch cực tiểu hóa chi phí thực thi luồng công việc theo hàm mục tiêu ở công thức (5)

Algorithm Scheduling_Heuristic

1. Tính ma trận chi phí thực thi các task trên các host
2. Tính ma trận chi phí truyền thông giữa các host
3. Tính ma trận khối lượng dữ liệu vào/ra giữa các task
4. Khởi tạo danh sách các task sẵn sàng là danh sách tất cả các task
5. Khởi tạo danh sách các task chưa lập lịch là danh sách tất cả các task
6. repeat

7. foreach t_i in readyTasks do

8. Gán t_i cho thực hiện bởi PC_j theo thuật toán PSO_Algorithm

9. end for

10. Chờ xử lý công việc (phụ thuộc dữ liệu đầu vào và đầu ra giữa task cha-con).

11. Cập nhật lại các task ở trạng thái "ready"

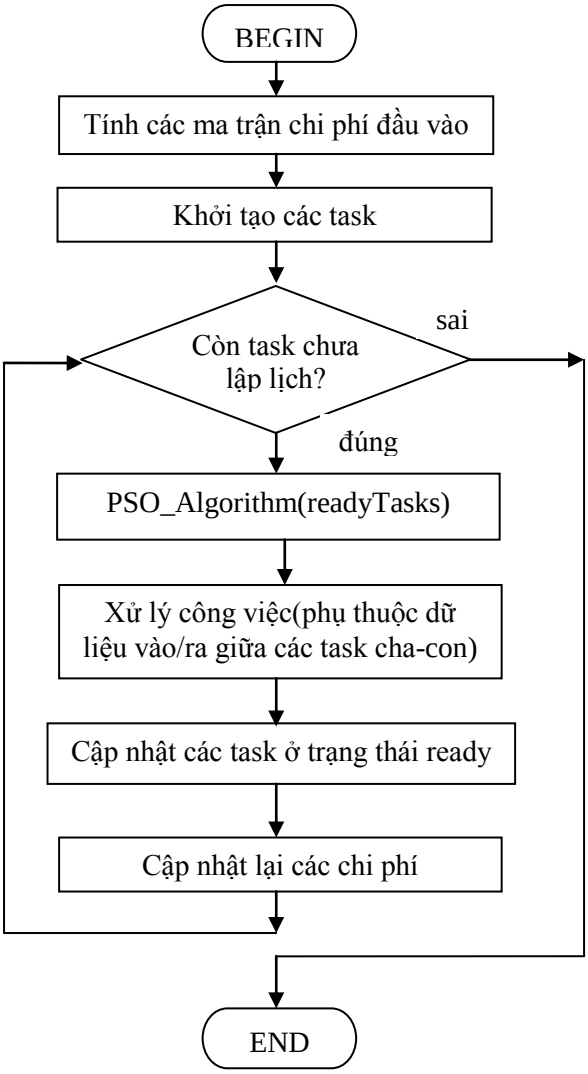
12. Cập nhật lại chi phí giao tiếp giữa các tài nguyên theo trạng thái mạng hiện tại.

13. Tính toán PSO($\{t_i\}$).

14. Until không có task chưa được lập lịch.

end procedure

Sơ đồ thuật toán Scheduling_Heuristic



Thuật toán thực hiện tính toán các ma trận chi phí thực thi của các tác vụ trên các host, ma trận chi phí truyền thông giữa các host và ma trận dữ liệu vào/ra giữa các tác vụ cha-con, sau đó khởi tạo ngẫu nhiên quần thể với số cá thể xác định, và thực hiện việc gán các tác vụ vào các host sau đó sẽ cực tiểu hóa các chi phí theo hàm mục tiêu đặt ra.

VI. KẾT QUẢ THỰC NGHIỆM

Để thực hiện mô phỏng việc lập lịch workflow trong môi trường đám mây, chúng tôi cài đặt giải thuật **Scheduling_Heuristic** bằng ngôn ngữ Java với sự trợ giúp của gói thư viện JSwarm [1,14] và công cụ mô phỏng CloudSim [1]. Hình 2 cho thấy kết quả thực nghiệm được so sánh giữa giải thuật **Scheduling_Heuristic** với 3 giải thuật: PSO_Heuristic [2], Random [3,4], RoundRobin [5] với dữ liệu tính toán dưới đây.

***Bảng** 7. Ma trận dữ liệu TP, PP, DS cho bộ dữ liệu Test*

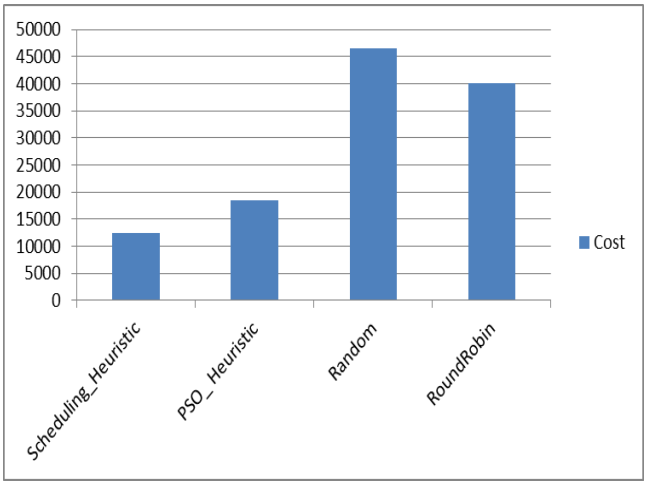
TP[5x3]		PC1	PC2	PC3
	T_1	0.1*25	0.2*25	0.3*25
	T_2	0.1*25	0.2*25	0.3*25
	T_3	0.1*25	0.2*25	0.3*25
	T_4	0.1*25	0.2*25	0.3*25
	T_5	0.1*25	0.2*25	0.3*25
PP[3x3]		PC1	PC2	0.3*25
	PC_1	0	0.1	0.1
	PC_2	0.1	0	0.1
	PC_3	0.1	0.1	0
$DS_{T2,T3,T4}$ [2x2]		Data Size	DS_{T5} [2x2] =	DataSize (MB)
	Input	10	Input	30
	Outp	10	Output	60

Giá trị ở bảng trên được lấy từ bảng giá sử dụng dịch vụ của Amazon EC2 [15] cho các tài nguyên trong phạm vi 1.1$ - 1.28$/giờ

Các tham số giải thuật :
Số cá thể = 25; số thế hệ = 30; số lần lặp = 30;
Trọng số quán tính w = 0.85 và hệ số gia tốc C1 = 1.5 và C2 = 0.5

Bảng 8. *Kết quả tính toán chi phí thực thi workflow sau 30 lần chạy*

Lần lặp	Scheduling _Heuristic	PSO_ Heuristic	Random	Round Robin
30	12.500	18.500	46500	40000



Hình 2. *Chi phí luồng công việc tìm được qua các giải thuật*

Nhận xét

Thực nghiệm được tiến hành trên những số liệu thực tế về chi phí xử lý dữ liệu và chi phí truyền thông tin giữa các vị trí địa lý khác nhau. Những số liệu này được thu thập và cung cấp bởi công ty Amazon [15], nhìn chung kết quả thực nghiệm đã khẳng định hiệu quả vượt trội của giải thuật đề xuất so với các giải thuật đối sánh.

Cụ thể, giải thuật đề xuất Scheduling_Heuristic cho kết quả phụ thuộc vào việc thiết lập các hệ số quán tính w, hệ số gia tốc C_1, C_2. Trong bài báo này chúng tôi đã sử dụng các trọng số quán tính w = 0.85, hệ số gia tốc C_1 = 1.5 và C_2 = 0.5, kết quả được thử nghiệm với số cá thể là 25, số lần lặp là 30 lần, như bảng kết quả đã chỉ ra chi phí của luồng công việc tính toán được bởi thuật toán Scheduling_Heuristic có giá trị thấp nhất so với các thuật toán Random, Round Robin và PSO_Heuristic.

V. KẾT LUẬN

Bài báo đã xây dựng một mô hình toán học cho bài toán luồng công việc trong môi trường điện toán đám mây nhằm cực tiểu hóa chi phí thực thi luồng công việc, đây là yêu cầu hết sức cần thiết trong môi trường điện toán đám mây vì điện toán đám mây là một môi trường dịch vụ tích hợp được cung cấp bởi các nhà cung cấp dịch vụ và người sử dụng phải trả chi phí cho các dịch vụ mà họ sử dụng. Đồng thời chúng tôi đã đề xuất một giải thuật lập lịch heuristic dựa trên chiến lược tối ưu bày đàn và cài đặt thử nghiệm trên môi trường mô phỏng cloudSim, kết quả chỉ ra việc tính toán chi phí tốt hơn các thuật toán đã có như Random hay Round Robin và PSO_Heuristic.

TÀI LIỆU THAM KHẢO

[1] Công cụ mô phỏng CloudSim http://www.cloudbus.org/cloudsim/

[2] S. PANDEY, L.WU, S.GURU, AND R.BUYYA, *A Particle Swarm Optimization (PSO)-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments*, The 24th IEEE International Conference on. Advanced Information Networking and Applications AINA, Australia, April, 2010.

[3] M.MICHAEL, E.UPFAL, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, April 2005. Cambridge University Press

[4] DON FALLIS. *The Reliability of Randomized Algorithms*, British Journal for the Philosophy of Science 51:255–271.

[5] SILBERSCHATZ, ABRAHAM, GALVIN, B.GAGNE, GREG, *Process Scheduling. Operating System Concepts* John_Wiley_&_Sons (Asia), pp. 194. ISBN 978-0-470-23399-3.

[6] J.D.ULLMAN, *NP-complete scheduling problems*, Journal of Computer and System Sciences, Volume 10, Issue 3, 1975

[7] S. PARSA, R. E. MALEKI, *RASA. A New Task Scheduling Algorithm in Grid Environment*, International Journal of Digital Content Technology and its Applications, Vol. 3, No. 4, 2009

[8] J.M. COPE, N. TREBON, H.M. TUFO, P.BECKMAN, *Robust data placement in urgent computing environments*, IEEE International Symposium on Parallel & Distributed Processing, IPDPS 2009

[9] A. AGARWAL, S. JAIN, *Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment*, International Journal of Computer Trends and Technology (IJCTT), vol. 9, 2014

[10] M.WIECZOREK, *Marek Scheduling of Scientific Workflows in the ASKALON Grid Environment*, ACM SIGMOD Record Journal, Vol. 34, Issue 3, 2005.

[11] A. SALMAN, *Particle swarm optimization for task assignment Problem*, Microprocessors and Microsystems, 2002.

[12] S. PANDEY, A. BARKER, K. K. GUPTA, R. BUYYA, *Minimizing Execution costs when using globally distributed cloud services*, 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.

[13] J. KENNEDY, R. EBERHART, *Particle Swarm Optimization*. IEEE International Conference on Neural Networks, ICNN.1995

[14] Thư viện JSwarm http://jswarm-pso.sourceforge.net.

[15] http://aws.amazon.com/ec2

Ngày nhận bài : 15/11/2014

SƠ LƯỢC VỀ TÁC GIẢ

PHAN THANH TOÀN

Sinh năm 1974 tại Thái Nguyên.

Tốt nghiệp đại học và thạc sĩ tại Trường ĐH Bách Khoa Hà Nội, nghiên cứu sinh năm 2012 tại học viện Khoa học công nghệ quân sự.

Hiện đang công tác tại Trường ĐH 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 THẾ LỘC

Sinh năm 1972, tại Hà Nội.

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

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 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.

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

ĐỖ NHƯ LONG

Sinh năm 1984 tại Sơn Tây, Hà Nội

Tốt nghiệp đại học tại Trường ĐH Sư phạm Hà Nội, tốt nghiệp thạc sĩ tại Trường ĐH Công nghệ - ĐH Quốc Gia Hà Nội

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

Lĩnh vực nghiên cứu : An Ninh Mạng, Bảo mật trong hệ thống mạng không dây

Điện thoại : 0983.120.984

E-mail: Longdn@hnue.edu.vn