

ẢNH HƯỞNG CỦA TỔ CHỨC CACHE ĐA CẤP VÀ MẠNG LIÊN KẾT ĐẾN HIỆU NĂNG CỦA CHIP ĐA LỖI

EFFECTS OF MULTI-LEVEL CACHE ORGANIZATION AND INTERCONNECT NETWORK ON PERFORMANCE OF MULTI-CORE CHIP

Hồ Văn Phi¹⁾, Hồ Khánh Lâm²⁾

¹⁾Trường Đại học Quy Nhơn; Email: hvphi@ftt.edu.vn

²⁾Trường Đại học sư phạm kỹ thuật Hưng Yên; Email: lamhokhanh@gmail.com

Tóm tắt: Ngày nay, công nghệ chip ASIC, PLD, và FPGA đã tạo cơ hội cho các nhà nghiên cứu để thiết kế chế tạo chip xử lý đa lõi. Trong khi đó, công nghệ chip đa lõi với tổ chức cache đa cấp và sự lựa chọn cấu hình mạng liên kết các lõi đảm bảo hiệu năng cao cho ứng dụng của chip đa lõi trong các hệ thống tính toán song song tốc độ cao là một xu hướng nghiên cứu và chế tạo hiện nay. Bài báo trình bày một trong những vấn đề được quan tâm trong công nghệ chip đa lõi là tổ chức cache, mạng liên kết các lõi xử lý trên chip, và đưa ra các tính toán đánh giá hiệu năng của tổ chức cache đa cấp thông qua xác định mức tăng tốc đạt được, và đề xuất giải pháp xác định trễ truyền thông của mạng liên kết các lõi xử lý – một nhân tố ảnh hưởng đến tốc độ thực hiện tính toán song song của chip đa lõi.

Từ khóa: Chip đa lõi; cache đa cấp; mạng liên kết trên chip; hiệu năng; trễ truyền thông

1. Đặt vấn đề

Với xu hướng phát triển của công nghệ vi xử lý đa lõi là tiếp tục tăng số lượng lõi trên một chip, nhưng cũng làm gia tăng tính phức tạp của các thành phần trên chip đa lõi xử lý đa lõi. Hiệu năng của hệ thống vi xử lý đa lõi phụ thuộc rất nhiều vào số lượng lõi, số luồng trong mỗi lõi, tổ chức cache, số cấp cache và cấu trúc mạng liên kết giữa các cấp cache trên chip [1].

Hiện nay, các nhà sản xuất đã sản xuất thương mại các chip đa xử lý đa lõi với số lõi là 2, 4, 6, 8 lõi. Các kiến trúc đa lõi thường sử dụng 2 cấp cache với L1 cache riêng cho mỗi lõi và L2 cache chia sẻ cho tất cả các lõi như: các bộ xử lý UltraSPARCT2 8-lõi, UltraSPARC T3 16-lõi, Rock 16-lõi của Sun; Core 2 duo 2-lõi, Core 2 quad 4-lõi của Intel. Cũng có một số chip đa xử lý đa lõi có 3 cấp cache với L1 cache riêng cho mỗi lõi, L2 cache riêng cho mỗi lõi hoặc chia sẻ cho 2 hay 4 lõi và L3 cache chia sẻ cho tất cả các lõi như: bộ xử lý Dunnington 6-lõi, Nehalem core i5 4-lõi, core i7 4 và 6-lõi và Xeon E7 10-lõi của Intel, Opteron 8-lõi của AMD... Tuy nhiên, cũng có một số chip chỉ sử dụng 2 cấp cache với L2 cache riêng cho từng lõi như: bộ xử lý Niagara 5-lõi của Sun, Tile64 64-lõi của Tileria [3]. Hầu hết các kiến trúc xử lý đa lõi trên sử dụng mạng liên kết trên chip theo các cấu hình: bus chia sẻ, crossbar-switched và 2Dmesh [1, 3, 4]. Các cấu hình liên kết này chỉ phù hợp cho các chip đa lõi có quy mô nhỏ, có độ trễ truyền thông cao và khả năng mở rộng thấp. Do đó, khi số lượng lõi trên chip tăng sẽ gây ra trễ truyền thông quá lớn, mức tăng tốc giảm gây ra nghẽn nút cổ chai làm suy giảm hiệu năng và khả năng mở rộng của bộ xử lý. Đây là thách thức lớn cho các nhà nghiên cứu và sản xuất chip đa lõi hiện nay [2].

Abstract: Today, chip ASIC, PLD and FPGA technology has created opportunities for researchers to design and manufacture processor chips. Meanwhile, the multi-core chip technology with a multi-level cache organization and a choice of network configuration connecting cores to ensure high performance for multi-core chip applications in the speed parallel computing systems is a trend of researching and manufacturing today. The article stated that one of the concerns in the multi-core chip technology is cache organization, core interconnect networks on chip, and offering calculations for performance evaluation of multi-level cache organization through determining acceleration rate achieving, and proposes solutions to determine communication overhead of a core processor interconnect network - a factor that affects the execution speed of parallel computing of multi-core chip.

Key words: Chip multi-core; multi-level cache; interconnect networks on chip; performance; communication overhead

Trong bài báo này chúng tôi tiến hành các tính toán, so sánh các hệ thống phân cấp cache và các cấu trúc mạng liên kết khác nhau trên chip để đánh giá ảnh hưởng của tổ chức cache đa cấp và mạng liên kết trên chip đa lõi đến hiệu năng của bộ xử lý, từ đó đề xuất một tổ chức cache và cấu hình mạng liên kết phù hợp cho kiến trúc đa lõi nhằm nâng cao hiệu năng của chip xử lý đa lõi.

2. Giải quyết vấn đề

2.1. Hiệu năng của tổ chức cache đa cấp

2.1.1. Cache riêng và cache chia sẻ

Hầu hết các chip vi xử lý đa lõi hiện nay thiết kế 2 cấp cache và cấp cache cuối cùng luôn là cache thống nhất, chia sẻ và thông minh. Kỹ thuật cache chia sẻ thông minh đảm bảo được tốc độ truyền dữ liệu giữa các lõi nhanh, tỷ số hiệu năng/chi phí cao hơn so với cache riêng, tăng hiệu quả sử dụng của cache chia sẻ, giảm dư thừa dữ liệu lưu trữ và giảm lưu lượng của bus bộ nhớ. Ngoài ra, tổ chức này còn có ưu điểm là đảm bảo được tính nhất quán cache, tiết kiệm băng thông bộ nhớ, đồng thời dung lượng của cache chia sẻ lớn làm tỷ số trượt cache (cache miss rate) giảm nhiều so với cache riêng và băng thông bus tăng làm tăng tốc độ truyền thông giữa cache và bộ nhớ chính. Tuy nhiên, cache chia sẻ cũng có nhược điểm là thời gian trùng cache (cache hit time) lớn hơn so với cache riêng [1, 3].

Khi số cấp cache trên chip là 3 cấp (L1, L2, L3 cache), để đánh giá hiệu năng của tổ chức cache riêng và chia sẻ ở cấp cache cuối, chúng tôi giả sử: bộ xử lý làm việc ở 2GHz.

- Kích thước L1 cache (L1 cache size) = 32KB, L1 hit

time = 1ns, L1 miss rate = 15%.

- Kích thước L2 cache (L2 cache size) = 256 KB, L2 hit time = 3ns, L2 miss rate = 30%,

- Đối với bộ xử lý có L3 cache riêng:

Kích thước L3 cache (L3 cache size) = 1MB, L3 hit time = 5ns, L3 miss rate = 40%, kích thước khối nhớ L3 cache (L3 cache block size) = 64B.

- Đối với bộ xử lý có L3 cache chia sẻ:

Kích thước L3 cache (L3 cache size) = 4MB, L3 hit time = 10ns, L3 miss rate = 20%, kích thước khối nhớ L3 cache (L3 cache block size) = 64B.

- Bộ nhớ chính:

Độ rộng bus bộ nhớ (memory bus wide) = 64bits, trễ truy cập bộ nhớ (memory access latency) = 40ns, tốc độ bus (bus speed) = 1000MHz hay chu kỳ bus (bus cycle) = 1ns.

Sử dụng các công thức trong [3], chúng tôi tính toán được thời gian truy cập bộ nhớ trung bình AMAT (Average memory access time):

$$\text{Transfer rate of memory bus} = \text{memory bus wide} / \text{bus cycle} \quad (1)$$

$$= 64\text{bits} / 1\text{ns} = 8\text{B} / \text{ns}$$

$$\text{Time to transfer one L3 cache block} \quad (2)$$

$$= \text{L3 cache block size} / \text{transfer rate of memory bus}$$

$$= 64\text{B} / (8\text{B} / \text{ns}) = 8\text{ns}$$

$$\text{L3 miss penalty} = \text{memory access latency} \quad (3)$$

$$+ \text{time to transfer one L3 cache block}$$

$$= 40\text{ns} + 8\text{ns} = 48\text{ns}$$

a. Đối với bộ xử lý có L3 cache riêng:

$$\text{AMAT1} = \text{L1 hit time} + \text{L1 miss rate} \times (\text{L2 hit time} \quad (4)$$

$$+ \text{L2 miss rate} (\text{L3 hit time} + \text{L3 miss rate} \times \text{L3 miss penalty}))$$

$$= 1 + 0,15 \times (3 + 0,3 \times (5 + 0,4 \times 48))$$

$$= 2,54\text{ns}$$

b. Đối với bộ xử lý có L3 cache chia sẻ:

$$\text{AMAT2} = \text{L1 hit time} + \text{L1 miss rate} \times (\text{L2 hit time} \quad (5)$$

$$+ \text{L2 miss rate} \times (\text{L3 hit time} + \text{L3 miss rate} \times \text{L3 miss penalty}))$$

$$= 1 + 0,15 \times (3 + 0,3 \times (10 + 0,2 \times 48))$$

$$= 2,33\text{ns}$$

Kết quả tính toán cho thấy rằng với chip xử lý đa lõi có 3 cấp cache, thời gian truy cập bộ nhớ trung bình (AMAT) trong chip có cache cấp cuối là cache chia sẻ nhỏ hơn so với cache cấp cuối là cache riêng.

2.1.2. Tốc độ thực hiện chương trình

Một chương trình ứng dụng được biên dịch để chạy trên một máy tính với các thông số được cung cấp bao gồm [3]:

Tổng số lệnh trong một chương trình là I, được xác định bằng số lệnh/chương trình (instructions per program).

Một lệnh trung bình cần một số chu kỳ đồng hồ để

thực hiện là $\text{CPI}_{\text{execution}}$, được xác định bằng số chu kỳ đồng hồ/lệnh (clock cycles per instruction)

Trung bình với một lệnh số chu kỳ đồng hồ tri hoãn bộ nhớ là MSPI được xác định bằng số chu kỳ đồng hồ tri hoãn bộ nhớ/lệnh (memory stalls clock cycle per instruction):

Thời gian của một chu kỳ đồng hồ là $C = 1 / \text{tốc độ đồng hồ}$ (1/clock rate), được xác định bằng giây/chu kỳ (seconds/cycle).

Để xác định thời gian thực hiện chương trình của các CPU, chúng tôi xét hai trường hợp: CPU có 3 cấp cache và CPU có 2 cấp cache với cấp cache cuối là cache chia sẻ:

Mặc định rằng một chương trình có các thông số sau: tổng số lệnh $I = 10000000$ lệnh, $\text{CPI}_{\text{execution}} = 2,5$ chu kỳ/lệnh, tốc độ đồng hồ CPU (CPU clock rate) = 2GHz. Giả sử trong 1000 lệnh tham chiếu bộ nhớ của CPU có 150 lệnh trượt L1 cache hay L1 miss rate = 0,15, thời gian trượt L1 cache (L1 hit time) = 2 chu kỳ đồng hồ; có 45 lệnh trượt L2 cache hay L2 miss rate = 0,3, L2 hit time = 6 chu kỳ đồng hồ; có 9 lệnh trượt L3 cache tham chiếu vào bộ nhớ chính hay L3 miss rate = 0,2; L3 hit time = 20 chu kỳ đồng hồ; L3 miss penalty = 96 chu kỳ đồng hồ; trong một lệnh thời gian truy cập bộ nhớ (memory accesses per instruction) = 1,5 chu kỳ đồng hồ.

Sử dụng các công thức trong [3], chúng tôi xác định được:

a. Đối với CPU có 3 cấp cache

$$\text{MSPI1} = \text{memory accesses per instruction} \times \text{L1 miss rate} \quad (6)$$

$$\times (\text{L2 hit time} + \text{L2 miss rate} (\text{L3 hit time} + \text{L3 miss rate} \times \text{L3 miss penalty}))$$

$$= 1,5 \times 0,15 \times (6 + 0,3 \times (20 + 0,2 \times 96))$$

$$= 4 \text{ chu kỳ đồng hồ}$$

$$\text{CPU1}_{\text{execution}} \text{ time} = I \times (\text{CPI}_{\text{execution}} + \text{MSPI1}) \times C \quad (7)$$

$$= (10^7) \times (2,5 + 4) \times (0,5 \times 10^{-9}) = 0,033\text{s}$$

b. Đối với CPU có 2 cấp cache

$$\text{MSPI2} = \text{memory accesses per instruction} \times \text{L1 miss rate} \quad (8)$$

$$\times (\text{L2 hit time} + \text{L2 miss rate} \times \text{L2 miss penalty}))$$

$$= 1,5 \times 0,15 \times (6 + 0,3 \times 96)$$

$$= 7,83 \text{ chu kỳ đồng hồ}$$

$$\text{CPU2}_{\text{execution}} \text{ time} = I \times (\text{CPI}_{\text{execution}} + \text{MSPI2}) \times C \quad (9)$$

$$= (10^7) \times (2,5 + 7,83) \times (0,5 \times 10^{-9}) = 0,052\text{s}$$

Mức tăng tốc thực hiện chương trình của CPU có 3 cấp cache so với 2 cấp cache là:

$$\text{Speedup} = \frac{\text{CPU2}_{\text{execution}} \text{ time}}{\text{CPU1}_{\text{execution}} \text{ time}} = \frac{0,052}{0,033} = 1,58 \quad (10)$$

Từ kết quả trên, chứng tỏ rằng khi số cấp cache tăng dẫn đến tốc độ thực hiện chương trình của CPU tăng tức là hiệu năng của bộ xử lý được nâng cao đáng kể.

2.2. Ảnh hưởng của mạng liên kết đến hiệu năng của chip vi xử lý đa lõi

Hiệu năng của chip vi xử lý đa lõi với đa cấp cache chỉ được nâng cao đáng kể khi chúng được sử dụng cho các ứng dụng song song. Các ứng dụng song song thường chia ra hai phần: phần thực hiện tuần tự vốn có và phần thực hiện song song. Khi đó, theo luật Amdahl [5, 6].

Mức tăng tốc tối đa khi bỏ qua trễ truyền thông được xác định:

$$S = \frac{1}{(1-f) + \frac{f}{n}} \quad (11)$$

Trong đó:

- f: tỷ lệ phần thực hiện song song.
- 1-f: tỷ lệ phần thực hiện tuần tự.

Mức tăng tốc S(p,n) khi tính đến trễ truyền thông được xác định:

$$S(p,n) = \frac{T_{seq}(p) + T_{par}(p)}{T_{seq}(p) + \frac{T_{par}(p)}{n} + T_O(p,n)} \quad (12)$$

Trong đó:

p: số lượng gói dữ liệu.

- n: số lõi xử lý.
- Tseq(p): thời gian thực hiện phần tuần tự.
- Tpar(p): thời gian thực hiện phần song song.
- TO(p,n): trễ truyền thông giữa các lõi xử lý.

Ở đây, chúng tôi sử dụng công thức (12) để phân tích và xác định trễ truyền thông cho các cấu hình mạng liên kết trên chip.

Khi n (số lõi trên chip) lớn, thì:

$$S(p,n) = \frac{T_{seq}(p) + T_{par}(p)}{T_{seq}(p) + T_O(p,n)} \quad (13)$$

Nếu phần thực hiện song song gồm các luồng chạy song song trên từng lõi xử lý thì mức tăng tốc phụ thuộc vào Tseq(p) và trễ truyền thông TO(p,n) giữa các luồng chạy trên các lõi. Lập trình song song phải đảm bảo sao cho Tseq(p) và TO(p,n) tối thiểu nhất.

Tổng thời gian thực hiện chương trình trên chip đa lõi với tổ chức cache nhiều cấp chính là thời gian thực hiện của CPU (CPU_{execution} time) = Tseq(p) + Tpar(p). Để xác định các thời gian này chúng tôi xét bộ xử lý đa lõi 3 cấp cache với L3 cache chia sẻ ở mục 2.1.2 theo kết quả (7) ta có:

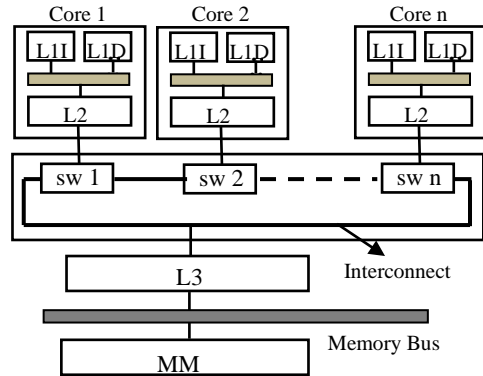
$$CPU_{execution} \text{ time} = T_{seq}(p) + T_{par}(p) = 0,033.$$

Để xác định ảnh hưởng của TO(p,n), chúng tôi cho rằng thời gian thực hiện tuần tự bằng 10% tổng CPU_{execution} time, do đó Tseq(p) = 0,003.

Khi đó:

$$S(p,n) = \frac{0,033}{0,003 + T_O(p,n)} \quad (14)$$

TO(p,n) phụ thuộc vào cấu hình mạng liên kết các lõi trên chip. Trong mục 2.1.2 chúng tôi đã lựa chọn tổ chức cache 3 cấp với L3 cache chia sẻ, trong đó các lõi liên kết với nhau thông qua các thành phần chuyển mạch kết nối với L3 cache như hình 1.



Hình 1. Chip đa lõi với tổ chức cache 3 cấp: L1I, L1D, L2 riêng lẻ cho mỗi lõi và L3 cache chia sẻ cho tất cả các lõi.

Để phân tích, chúng tôi chọn một số cấu hình mạng liên kết trên chip xử lý đa lõi phổ biến với các tham số đặc trưng như trong bảng 1.

Bảng 1. Các thông số của một số cấu hình mạng liên kết trên chip đa xử lý [1,3,4].

Mạng	Degree (d)	Aver. Hop count (H)	Bisection width (B)
Star	N	1	1
Ring	2	(N+1)/3	2
2DMesh	4	(2/3)N ^{1/2}	N ^{1/2}
2DTorus	4	(1/2)N ^{1/2}	2N ^{1/2}
3DMesh	6	N ^{1/3}	N ^{2/3}
3DTorus	6	(3/4)N ^{1/3}	2N ^{2/3}

Trong đó:

- N = n: tổng số nút chuyển mạch trong mạng liên kết (tổng số lõi trên chip).
- Degree (d): cấp độ của nút là số liên kết với một nút chuyển mạch.
- Aver. Hop count (H): khoảng cách định tuyến trung bình giữa các cặp nút chuyển mạch.
- Bisection width (B): độ rộng chia đôi số lượng các liên kết của lát cắt nhỏ nhất mà nó tách mạng thành hai nửa bằng nhau.

Chúng tôi đề xuất công thức tính trễ truyền thông trung bình trong chip vi xử lý đa lõi như sau:

$$T_O(p,n) = Hp_w (t_{syn} + t_{sw} + t_{Inter-switch}) \pi_{sw} \quad (15)$$

Trong đó:

- t_{Inter-switch}: trễ giữa các nút chuyển mạch.
- t_{sw}: trễ của nút chuyển mạch.
- t_{syn}: trễ cho đồng bộ truyền thông giữa các nút gửi và nhận gói dữ liệu.

Cho rằng $t_{sw} + t_{syn} = 1$ chu kỳ đồng hồ, trễ $t_{inter-switch} = 1$ chu kỳ đồng hồ.

- w : kích thước của gói dữ liệu (bits).
- p : số lượng gói dữ liệu.
- π_{sw} : xác suất định tuyến chuyển dữ liệu qua chuyển

mạch mạng liên kết đến lõi khác.

Trễ truyền thông giữa các nút chuyển mạch phải tính đến các xác suất định tuyến lưu lượng của từng lõi π_{core} và xác suất định tuyến đến cấp cache chia sẻ π_{cache} . Các xác suất này thỏa mãn điều kiện:

$$\pi_{core} + \pi_{cache} + \pi_{sw} = 1 \tag{16}$$

Để phân tích, chúng tôi chọn $\pi_{sw} = 0,2$ và $w = 8$ bits. Khi đó:

$$T_O = Hp(8)(2)(0,2) = 3,2Hp \tag{17}$$

Để xác định công thức tính độ trễ cho các cấu hình mạng liên kết, chúng tôi kết hợp công thức (17) với kết quả của H đã cho ở bảng 1. Ở đây không xét đến cấu hình mạng Star là vì mạng Star không phải là mạng phân cấp nên không phù hợp với liên kết các lõi trên chip.

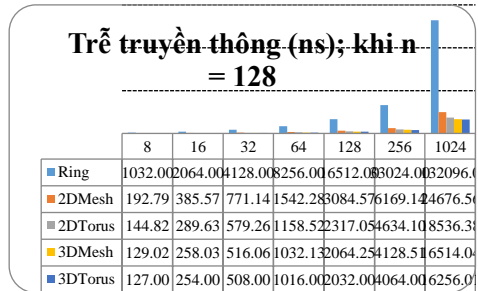
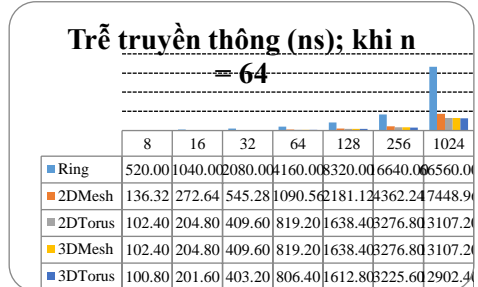
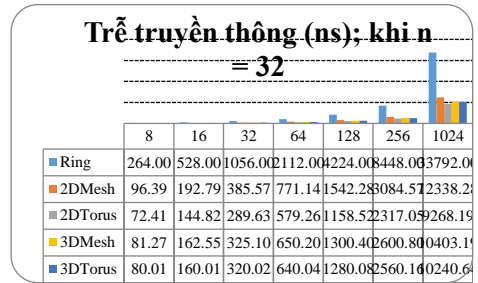
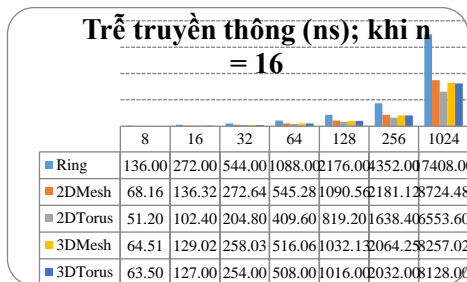
Khi đó, công thức trễ truyền thông trung bình cho các cấu hình mạng liên kết:

$$\begin{aligned} T_{ORing} &= \frac{3,2(n+1)}{3} p \approx (n+1)p; \\ T_{O2DMesh} &= \frac{6,4}{3} p\sqrt{n} \approx 2,13p\sqrt{n}; \\ T_{O2DTorus} &= \frac{3,2}{2} p\sqrt{n} = 1,6p\sqrt{n}; \\ T_{O3DMesh} &= 3,2p\sqrt[3]{n}; \\ T_{O3DTorus} &= \frac{12,6}{4} p\sqrt[3]{n} = 3,15p\sqrt[3]{n} \end{aligned} \tag{18}$$

3. Kết quả nghiên cứu và bình luận

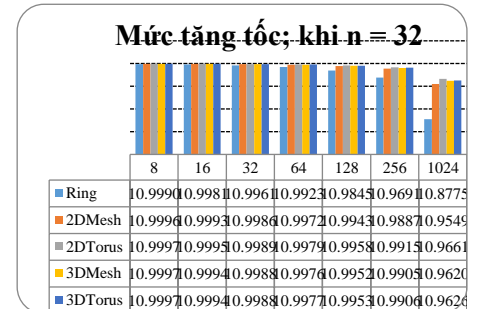
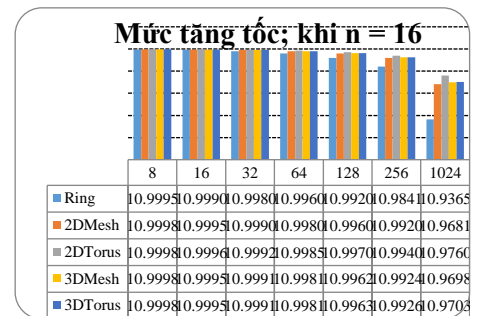
3.1. Kết quả nghiên cứu

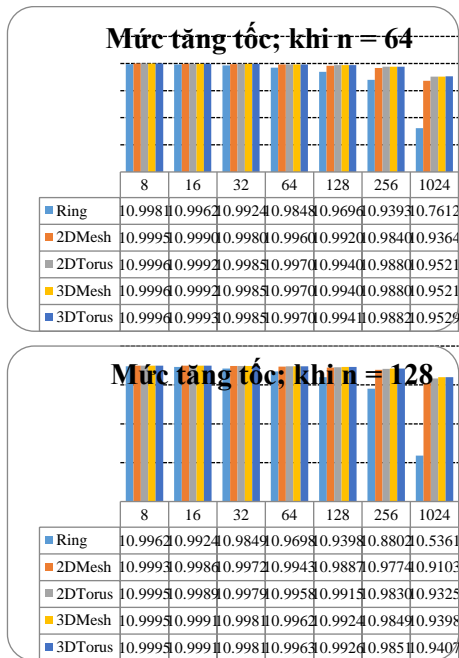
Từ các công thức (18) chúng tôi tiến hành tính toán, mô phỏng cho các trường hợp $n = 16, 32, 64, 128$ với $p = 8, 16, 32, 64, 128, 256, 1024$. Chúng tôi thu được các kết quả trễ truyền thông của các cấu hình mạng liên kết như trong hình 2.



Hình 2. Trễ truyền thông trung bình của các mạng liên kết Ring, 2DMesh, 2DTorus, 3DMesh, 3DTorus, với số lõi của trên chip: $n = 16; 32; 64; 128$.

Chúng tôi cũng kết hợp công thức (14) và (18) để tiến hành xác định mức tăng tốc cho các cấu hình mạng liên kết trên chip đa lõi. Các kết quả thu được trình bày như trong hình 3.





Hình 3. Mức tăng tốc cho các mạng liên kết Ring, 2DMesh, 2DTorus, 3DMesh, 3DTorus, trong các trường hợp số lõi của chip $n = 16, 32, 64, 128$.

3.2. Bình luận

Từ các kết quả trên, với số lõi n như nhau và số lượng gói dữ liệu p giống nhau, so sánh về truyền thông trung bình và mức tăng tốc xử lý của các cấu hình mạng liên kết trên chip, chúng tôi thấy rằng:

Khi số lõi trên chip $n < 64$ thì mạng liên kết 2DTorus cho về truyền thông trung bình nhỏ nhất và mức tăng tốc xử lý là lớn nhất.

Tuy nhiên, khi số lõi trên chip tăng $n \geq 64$ thì mạng

liên kết 3DTorus sẽ cho về truyền thông trung bình nhỏ nhất và mức tăng tốc xử lý là lớn nhất.

4. Kết luận

Các kết quả cho thấy rằng với chip vi xử lý đa lõi nên lựa chọn tổ chức cache 3 cấp với L3 cache chia sẻ cho các lõi và mạng liên kết các lõi trên chip 2DTorus ($n < 64$), mạng 3DTorus ($n \geq 64$) sẽ cho hiệu năng xử lý tốt nhất.

Đóng góp nghiên cứu của chúng tôi là thông qua việc nghiên cứu ảnh hưởng của tổ chức cache đa cấp và mạng liên kết đến hiệu năng của chip đa lõi, chúng tôi đã đề xuất mô hình tổ chức cache 3 cấp với cấp cuối là cache chia sẻ và các cấu hình mạng liên kết trên chip phù hợp với các kiến trúc vi xử lý đa lõi nhằm cải thiện hiệu năng của chip vi xử lý đa lõi.

Tài liệu tham khảo

- [1] S. Keckler, K. Olukotun, & H. P. Hofstee, "Multicore Processors and Systems", New York, NY: Springer, 2009.
- [2] John D. Owens, William J. Dally, Ron Ho, D.N. (Jay) Jayasimha, Stephen W. Keckler, Li-Shiuan Peh, "Research Challenges For on-Chip Interconnection Networks", IEEE Micro micr-27-05-owen.3d, 12/10/07, pp (96-108).
- [3] J.L. Hennessy and D.A. Patterson, "Computer Architecture: A Quantitative Approach", 5th edition, Elsevier Inc, 2011.
- [4] [Daniel Sanchez, George Michelogiannakis, And Christos Kozyrakis, "An Analysis of On-Chip Interconnection Networks for Large-Scale Chip Multiprocessors", ACM Transactions on Architecture and Code Optimization, Vol. 7, No. 1, Article 4, 2010.
- [5] Bashayer M. Al-Babtain, Fajer J. Al-Kanderi, Maha F. Al-Fahad, and Imtiaz Ahmad, "A Survey on Amdahl's Law Extension in Multicore Architectures", International Journal of New Computer Architectures and their Applications (IJNCAA) 3(3), 2013, pp (30-46).
- [6] Dong Hyuk Woo and Hsien-Hsin S. Lee, "Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era", IEEE Computer Society, 2008, pp (24-31).

(BBT nhận bài: 03/01/2014, phản biện xong: 20/03/2014)