

به نام خدا



درس معماری کامپیوتر
نیم سال دوم ۰۲-۰۳
استاد: دکتر حسین اسدی

دانشکده مهندسی کامپیوتر

تمرین سری نهم

- پرسش‌های خود را در صفحه quera مربوط به تمرین مطرح نمایید.
- سوالات نظری را حتماً به صورت انفرادی و سوالات عملی را می‌توانید در گروه‌های دو نفر تحویل دهید.
- پاسخ‌ها را به صورت تاپی بنویسید.
- اسکرین‌شات‌ها، عکس‌ها و فایل‌های مربوط به سوال عملی را در فایل فشرده مربوطه در cw و quera قرار دهید. هر گونه عدم تطابق بین دو تمرین آپلود شده در دو سایت منجر به از دست رفتن نمره تمرین مربوطه می‌شود.
- پی دی اف قسمت تئوری را در سامانه cw و quera بارگذاری کنید.
- هر دانشجو می‌تواند حداکثر سه تمرین را با دو روز تأخیر بدون کاهش نمره ارسال نماید.

تمارین تئوری

۱. (آ) فرض کنید یک واحد حافظه نهان^۱ به اندازه ۳ و کاملاً خالی در اختیار دارید، دسترسی‌های حافظه زیر به ترتیب از راست به چپ برای شما ارسال می‌شود با الگوریتم FIFO آن‌ها را درون حافظه نهان قرار دهید و تعداد miss‌های حافظه را محاسبه کنید.

4, 3, 2, 1, 4, 3, 5, 4, 3, 2, 1, 5

(ب) حال فرض کنید یک حافظه به اندازه ۴ در اختیار دارید، مجدداً دسترسی‌های بخش الف به حافظه رخ می‌دهند و باید آن‌ها را با همان الگوریتم FIFO در حافظه خود قرار دهید. تعداد miss‌های حافظه را به دست آورید.

(ج) به پدیده‌ای که مشاهده کردید Bélády's anomaly می‌گویند، در مورد آن تحقیق کنید.

۲. در مورد الگوریتم‌های Pseudo LRU و Clock تحقیق کنید و نحوه کارکرد آن‌ها را توضیح دهید. سپس به صورت خلاصه توضیح دهید که چرا به نظر شما استفاده از خود LRU در سخت افزار واقعی کار منطقی نیست.

۳. یک پردازنده، حافظه‌ای byte-addressable دارد که آدرس‌های آن ۳۲ بیتی هستند. یک حافظه نهان به اندازه‌ی ۵۱۲ کیلوبایت داریم که اندازه‌ی هر بلوک آن ۳۲ بایت است. فرض کنید این حافظه نهان از نوع two-way set associative است. به سوالات زیر پاسخ دهید:

(آ) این حافظه نهان در مجموع چند بلوک دارد؟

(ب) چند set دارید؟

(ج) فیلدهای tag و index و offset چندبیتی هستند؟

۴. یک سیستم از حافظه ۸ گیگابایتی با کلمات ۶۴ بیتی استفاده می‌کند. هر بلوک حافظه ۱۶ کلمه را در خود قرار می‌دهد. اگر قرار باشد از یک حافظه نهان نگاشت مستقیم^۲ متشکل از ۱۲۸ بلوک استفاده کنیم، نحوه آدرس‌دهی را مشخص کنید. اگر به جای این حافظه نهان از یک حافظه نهان 4-way set associative استفاده کنیم، آدرس‌دهی به چه صورت خواهد بود؟

۵. کد زیر را در نظر بگیرید:

```

1   for (i = 0; i < 64; i++) {
2       for (j = 0; j < 64; j++) {
3           sum1 += a[i][j];
4       }
5       for (j = 0; j < 32; j++){
6           sum2 += b[i][2*j];
7       }
8   }
```

در صورتی که خط‌های ۳ و ۵ در حافظه نهان باشند ۱۰ چرخه و سایر خطوط به ۴ چرخه زمانی نیاز دارند. اگر در حافظه نهان MISS اتفاق بیفتد ۴۰ چرخه برای انتقال داده به حافظه نهان نیاز است. فرض کنید حافظه نهان در ابتدا خالی باشد.

(آ) در صورت استفاده از حافظه نهان 2-way associative با سیاست جایگذاری LRU شامل ۱۶ جایگاه و با بلوک‌های ۱۶ بیتی چند چرخه برای اجرا لازم است؟

(ب) چه نوع محلیتی در کد بالا مشهودتر است؟ توضیح دهید.

(ج) تغییر کدام ویژگی‌ها حافظه نهان می‌تواند باعث بهبود سرعت اجرا شود؟

¹cache

²Direct-Map

۶. یک حافظه اصلی به بزرگی ۲۵۶K کلمه و یک حافظه نهان به بزرگی ۴ بلوک ۴ کلمه‌ای موجود است. فرض کنید از روش نگاشت مستقیم استفاده می‌کنیم در این صورت با فرض خالی بودن حافظه نهان، نرخ برخورد^۳ در انتهای صدور آدرس‌های ذیل از چپ به راست کدام است؟

170, 257, 168, 246, 176, 175, 176, 177, 175, 176, 177, 175, 176, 177, 176, 175, 174, 173, 172,
171, 170, 169, 168, 167, 168, 165, 164

³hit rate

تمارین عملی

۱. تمرین عملی اول

در این تمرین قصد داریم با انجام چند آزمون با شبیه‌ساز Gem5 و نحوه کار با این شبیه‌ساز آشنا شویم. برای کار با این شبیه‌ساز می‌توانید از ماشین مجازی که لینک آن در صفحه درس در CW قرار داده شده است و یا build کردن شبیه‌ساز Gem5 روی سیستم خودتان استفاده کنید. (در صورتی که روش دوم را انتخاب می‌کنید، توصیه می‌شود از یک سیستم عامل Unix-Based استفاده کنید.)

صورت تمرین

در این تمرین شما باید دو نمونه کد جمع و دو نمونه کد ضرب ماتریس را به وسیله شبیه‌ساز Gem5 اجرا کنید و نتایج آن‌ها را بررسی نمایید.

۱. **جمع دو ماتریس:** دو کد Adder1 و Adder2 را که در فایل add1.c و add2.c قرار دارند، در نظر بگیرید. این دو تابع که دو روش مختلف برای جمع دو ماتریس است را از لحاظ locality مقایسه کنید و قبل از انجام شبیه‌سازی، پیش‌بینی کنید که کدام عملکرد بهتری دارد. همچنین پس از اجرای شبیه‌سازی با بررسی فایل stats.txt عملکرد این دو کد را از لحاظ عملکرد L1DCache مقایسه کنید.

۲. **ضرب دو ماتریس:** کد ضرب ماتریس را در فایل mul.c در نظر بگیرید. روش کار این کد و عملکرد آن را با شبیه‌سازی در Gem5 بررسی کنید. همچنین با توجه به وجود حافظه L1DCache در شبیه‌سازی که انجام می‌دهید، یک پیاده‌سازی جدیدی برای ضرب دو ماتریس ارائه دهید. سپس کد جدید را با Gem5 شبیه‌سازی کنید. در نهایت فایل stats.txt را برای دو کد بررسی، تحلیل و مقایسه کنید.

راهنمایی

- در هنگام شبیه‌سازی در صورتی که کد کامپایل شده مناسب معماری X86 است، از X86TimingSimpleCpu استفاده کنید و اگر معماری سیستم شما ARM است، از ARMTimingSimpleCpu استفاده کنید.
- در شبیه‌سازی که انجام می‌دهید از دو لایه حافظه نهان استفاده کنید، به صورتی که حافظه نهان لایه اول برای داده و دستورالعمل‌ها به صورت جداگانه و حافظه نهان لایه دوم به صورت مشترک برای هر دو باشد.
- برای مشاهده ملموس تغییر نتایج در بین نمونه کدهای مختلف، لازم است که به اندازه ماتریس‌های و همچنین حافظه نهان توجه کنید.
- توصیه می‌شود در کدی که شبیه‌سازی می‌کنید خروجی در stdout نداشته باشید.

گزارش

گزارش شما باید شامل تمام کدهایی که به زبان C پیاده‌سازی کرده‌اید و همچنین شامل تحلیل‌های شما روی پارامترهای خروجی شبیه‌سازی انجام شده، باشد.

۲. تمرین عملی دوم

یکی از مباحث تحقیقاتی در آزمایشگاه (DSN) Data Storage, Networks, and Processing، تحلیل رفتار کاربردها به منظور بهینه‌سازی معماری‌های مرتبط با حافظه‌های نهان^۴ است. در این تمرین قصد داریم با قسمت اول این شاخه تحقیقاتی آشنا شویم. همان‌طور که در مباحث درس نیز اشاره شد، تمامی سیاست‌های معماری حافظه نهان برای تمامی کاربردها مناسب نیستند. در صورتی که این سیاست‌ها برای کاربردها درست انتخاب نشود، نه تنها بهبودی در کارایی مشاهده نمی‌شود بلکه می‌تواند منجر به کاهش کارایی نیز گردد. بدین منظور باید این کاربردها شناسایی شوند و سیاست‌های مناسب برای حافظه‌های نهان انتخاب گردد. در این تمرین قصد داریم با یک مورد از این مؤلفه‌ها یعنی محلیت فضایی^۵ بیشتر آشنا شویم.

^۴Cache

^۵Spatial Locality

صورت تمرین

در این تمرین باید یک کد به زبان C++ یا Python بنویسید که محلّیت مکانی چند فایل ردگیری^۶ از فضای ابری Alibaba که در صفحه درس در CW قرار داده شده است را محاسبه کند.

فرمت فایل Alibaba

فرمت هر خط فایل‌های ردگیری Alibaba به صورت زیر است:

Time Stamp(ns), Response Time(ns), Offset(Byte), Request Size (Byte),
Request Type(Read/Write), Process ID, Major Disk Number, Minor Disk Number

محلّیت فضایی

در یک طبقه‌بندی پیشنهادی می‌توان درخواست‌های ورودی/خروجی را به چهار کلاس اصلی طبقه‌بندی کرد که عبارتند از: الف) متوالی^۷، ب) تصادفی^۸، ج) گام‌به‌گام^۹ و د) همپوشانی^{۱۰}. از آنجایی که زمان پاسخگویی یک درخواست به درخواست‌های قبلی بستگی دارد، هر درخواست بر اساس درخواست‌های قبلی ارزیابی و طبقه‌بندی می‌شود. شکل زیر یک دنباله‌ی دلخواه از درخواست‌ها را در طول زمان نشان می‌دهد. به منظور شناسایی نوع یک درخواست به محض ورود یک درخواست، درخواست با درخواست‌های قبلی مقایسه می‌شود. تعداد درخواست‌های گذشته برای مقایسه به معماری کلی سیستم عامل و دستگاه‌های ذخیره‌سازی بستگی دارد. در اینجا، صف تاریخچه درخواست‌ها را ۶۴ (۱۲۸) در نظر می‌گیریم که نزدیک به صف‌های استفاده شده در دیسک و زمانبندی I/O در سیستم عامل است.

در شکل زیر، درخواست ۶ به عنوان درخواست متوالی طبقه‌بندی می‌شود زیرا نقطه شروع آن دقیقاً بعد از نقطه پایان یکی از درخواست‌های صادر شده قبلی است.

در شکل زیر درخواست ۲ نسبت به درخواست ۳ از نوع گام‌به‌گام هست زیرا نقطه شروع آن فاصله کمی با نقطه پایان درخواست سوم دارد.

در شکل زیر درخواست ۴ یک درخواست همپوشانی است زیرا مقداری از داده درخواست ۴ با یکی از درخواست‌های قبلی اشتراک دارد.

اگر هیچ درخواستی در مجاورت یک درخواست وجود نداشته باشد، مشابه درخواست ۵ در شکل زیر، به صورت تصادفی در نظر گرفته می‌شود.

راهنمایی

۱. بهتر است اول درخواست‌های متوالی بعد درخواست‌های همپوشانی و سپس درخواست‌های گام به گام را تشخیص دهید. سپس در صورتی که درخواست در این ۳ گروه نبود، آن را تصادفی در نظر بگیرید.
۲. بهتر است طول صف را پارامتری در نظر بگیرید تا بتوانید مقدار آن به طور دلخواه تنظیم کنید.
۳. فاصله درخواست با درخواست قبلی (حالت گام‌به‌گام) را نیز به صورت پارامتری در نظر بگیرید و آن را روی ۶۴KB تنظیم کنید. یعنی اگر فاصله دو درخواست بیشتر از ۶۴KB بود در دسته گام‌به‌گام قرار نمی‌گیرد.
۴. فقط لازم است درخواست با درخواست‌های قبلی مقایسه شود. نیازی به مقایسه با درخواست‌های بعدی نیست.

گزارش

در این گزارش باید ۸ نمودار برای چهار بارکاری Alibaba رسم کنید، یکبار با طول صف ۶۴ و بار دیگر با طول صف ۱۲۸ این نمودارها را رسم کنید. هریک از این نمودارها باید شامل ۴ میله باشد که درصد ۴ کلاس مختلف را نشان دهد.

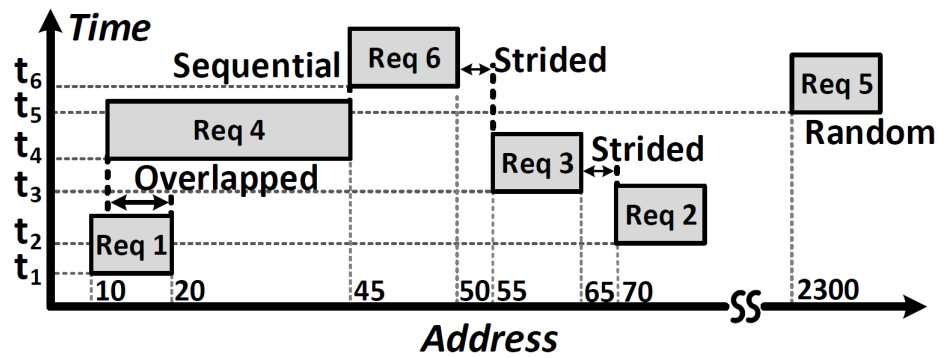
⁶Trace

⁷sequential

⁸random

⁹strided

¹⁰Overlapped



مطالعه بیشتر

در صورتی که علاقمند به مطالعه بیشتر در این شاخه پژوهشی هستید می‌توانید مقاله زیر را مطالعه نمایید:

Salkhordeh, Reza, Shahriar Ebrahimi, and Hossein Asadi. "ReCA: An efficient reconfigurable cache architecture for storage systems with online workload characterization." IEEE Transactions on Parallel and Distributed Systems 29, no. 7 (2018): 1605-1620.