پایان‌نامه کارشناسی ارشد

کاهش رقابت بر سر حافظه نهان در پردازنده‌های چند هسته‌ای با طراحی یک الگوریتم زمانبندی آگاه از اولویت

دانشجو

سید کاظم شکفته

استاد راهنما

جناب آقای دکتر حسین دلداری

استاد مشاور

جناب آقای دکتر محمود نقیب زاده

زمستان ۱۳۸۹

تعهدنامه

اینجانب **سید کاظم شکفته** دانشجوی دوره کارشناسی ارشد رشته **مهندسی کامپیوتر – گرایش نرم‌افزار** دانشکده **مهندسی** دانشگاه فردوسی مشهد نویسنده پایان‌نامه **کاهش رقابت بر سر حافظه نهان در پردازنده‌های چند هسته‌ای با طراحی یک الگوریتم زمانبندی آگاه از اولویت** تحت راهنمایی **جناب آقای دکتر حسین دلداری** متعهد می‌شوم:

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود و یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه فردوسی مشهد می‌باشد و مقالات مستخرج با نام "دانشگاه فردوسی مشهد" و یا "Ferdowsi University of Mashhad" به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تاثیرگذار بوده‌اند در مقالات مستخرج از رساله رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است، اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه فردوسی مشهد می‌باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

دانشگاه فردوسی مشهد
دانشکده مهندسی - گروه مهندسی کامپیوتر

آزمایشگاه تخصصی سیستم‌های توزیع شده و پردازش موازی

پایان‌نامه کارشناسی ارشد

کاهش رقابت بر سر حافظه نهان در پردازنده‌های چند هسته‌ای با طراحی یک الگوریتم زمانبندی آگاه از اولویت

دانشجو

سید کاظم شکفته

استاد راهنما

جناب آقای دکتر حسین دلداری

استاد مشاور

جناب آقای دکتر محمود نقیب زاده

زمستان ۱۳۸۹

تقدیم

به دنیای عظیم علم و دانش

به دنیای پیچیده‌ی سیستم‌های کامپیوتری

به تمام علاقمندان به مقوله‌های پردازش‌های فوق سریع

به همه‌ی دانشمندان و دانش‌پژوهان ایرانی

تقدیم به پدر و مادر مهربانم که همواره مشوق من در تمامی مراحل زندگی بوده‌اند و تقدیم به همسر عزیزم سرکار خانم مهندس مریم برداران خلخالی که در انجام این پایان نامه بسیار مرا یاری رساندند.

سپاسگزاری

از طرف دنیای مهندسی

سپاسگزارم خدای خالق طبیعت را

سپاسگزارم طبیعت شگفتی آفرین را

سپاسگزارم بنیانگذاران پردازنده‌های سریع را

با سپاس فراوان از اساتید محترم گروه مهندسی کامپیوتر دانشگاه فردوسی مشهد و همچنین جناب آقای دکتر حسین دلداری که به عنوان استاد راهنمای این پایان نامه همواره با سعه‌ی صدر اینجانب را در پیشبرد بهتر این پایان نامه راهنمایی نمودند و نیز جناب آقای پروفسور محمود نقیب‌زاده که به عنوان استاد مشاور این پایان نامه تأثیر بسزایی در انجام هر چه بهتر آن داشته‌اند.

با آرزوی موفقیت روزافزون برای این بزرگواران

...

چکیده

امروزه پردازنده‌های چند هسته‌ای که دارای واحدهای پردازشی متعددی بر روی یک تراشه‌ی سخت‌افزاری می‌باشند، به طور گسترده‌ای به عنوان راهی برای رسیدن به افزایش کارایی و موازی سازی درون پردازنده مورد استفاده قرار می‌گیرند. استفاده‌ی مناسب از منابع پردازشی در این پردازنده‌های می‌تواند در افزایش کارایی برنامه‌ها بسیار موثر باشد و در نقطه‌ی مقابل، عدم استفاده‌ی بهینه و شایسته از آنها نه تنها باعث عدم افزایش کارایی بلکه در مواردی باعث افت شدید کارایی در برنامه‌ها خواهد شد.

در این پردازنده‌ها برخی منابع به صورت اشتراکی بین هسته‌های مختلف مورد استفاده قرار می‌گیرند. مسئله‌ی رقابت بر سر منابع اشتراکی در پردازنده‌های چند هسته‌ای به عنوان یک چالش بزرگ اخیرا مورد توجه بسیاری از محققین در این زمینه قرار گرفته است. زمانبند سیستم‌عامل می‌تواند به عنوان یکی از ابزارهای بسیار مهم و کارا به منظور رفع این مشکل مورد استفاده قرار گیرد. در بسیاری از کارهای انجام شده در این زمینه، زمانبند صرفا به یک معیار در تصمیم‌گیری توجه می‌نماید: معیار میزان رقابت بر سر حافظه‌ی نهان. حال اینکه در یک محیط واقعی عوامل تأثیرگذاری دیگری نیز در تصمیم‌گیری زمانبند دخیل خواهند بود از جمله اولویت نخ‌ها. چرا که اولویت یکی از ابزار اعمال سیاست‌های مختلف محیطی بر روی پردازش‌ها می‌باشد.

در این پایان‌نامه با معرفی یک روش زمانبندی نخ‌ها بر روی یک پردازنده‌ی چند هسته‌ای میزان کاهش کارایی ناشی از رقابت نخ‌ها بر سر حافظه‌ی نهان بسیار کاهش یافته است. در این مکانیزم زمانبندی علاوه بر معیار رقابت، به اولویت نخ‌ها و نیز این مطلب که اجرای همزمان نخ‌های همکار تأثیر بسیار مثبتی در افزایش کارایی آنها خواهد داشت توجه شده است. نتایج شبیه‌سازی روش پیشنهادی نشان دهنده‌ی میانگین افزایش ۴.۹۳ برابر در کارایی سیستم بوده است.

فهرست مطالب

فهرست اشکال

فهرست جداول

فصل اول

مقدمه

۱–۱– معرفی پردازنده‌های چند هسته‌ای

در مدل معماری ساده‌ی فن نیومن[۱] یک برنامه، دنباله‌ای از دستورالعمل‌ها است که در حافظه‌ی کامپیوتر ذخیره شده و یکی پس از دیگری به شکل خطی اجرا می‌شوند.

با گذر زمان، پیشرفت‌های زیادی در تکنولوژی کامپیوترهای بزرگ[۲] که معماری فن نیومن را استفاده می‌نمودند، رخ داد. در دهه‌ی ۶۰ میلادی شاهد ظهور سیستم‌عامل‌های اشتراک زمانی[۳] بودیم. با اجرای این سیستم‌عامل‌ها بر روی کامپیوترهای بزرگ، برای اولین بار مفاهیم اجرای همروند[۴] برنامه‌ها معرفی شد. کاربران متعدد می‌توانستند به صورت همزمان به یک کامپیوتر دسترسی داشته باشند و کارهایشان[۵] را برای اجرا بر روی آن ارسال نمایند. از دید هر برنامه، آن برنامه تنها پردازشی[۶] است که در حال اجرا بر روی آن کامپیوتر می‌باشد. سیستم‌عامل مسئول انتساب پردازنده به هر یک از برنامه‌ها است. در این حالت، همروندی در سطح پردازش است و عمل تغییر وظایف به عهده‌ی برنامه‌نویس سیستمی باقی می‌ماند.

سیستم‌های کامپیوتری امروزی مانند کامپیوترهای شخصی[۷] دارای یک سیستم‌عامل تک کاربره هستند. در هر لحظه فقط یک برنامه قابل اجرا بر روی کامپیوتر است. برنامه از همان مدل معماری فن‌نیومن تبعیت می‌کند. در طی زمان، پیشرفت‌هایی که به صورت نمایی در کارایی سیستم‌های کامپیوتری رخ داده منجر به تولید بسترهای محاسباتی بسیار پیچیده‌ای شده است. تولیدکنندگان سیستم‌عامل‌ها از پیشرفت پردازنده‌ها و نیز سیستم‌های گرافیکی در طراحی و تولید محیط‌های کاری

---

[1] Von Neumann
[2] Mainframe
[3] Time-sharing
[4] Concurrent
[5] Job
[6] Process
[7] Personal Computer (PC)

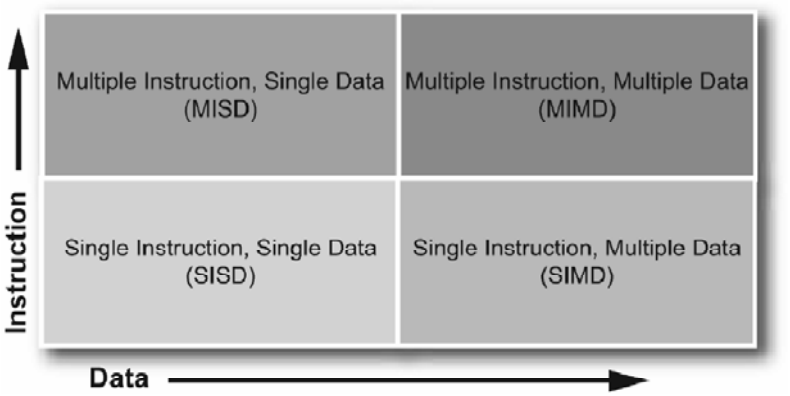
پیچیده‌تر استفاده نموده‌اند. واسط گرافیکی کاربر[1] به عنوان یک استاندارد شناخته می‌شود و این امکان را به کاربر می‌دهد که چندین برنامه را در محیط کاربری یکسانی اجرا کند.

این پیشرفت سریع هزینه‌هایی را در بر داشته است که یکی از آنها افزایش سطح انتظارات کاربر می‌باشد. کاربران انتظار دارند که بتوانند نامه‌های الکترونیک خود را ارسال نمایند در حالیکه در همان هنگام به یک موسیقی که بر روی اینترنت قرار دارد گوش می‌دهند. کاربران انتظار دارند بستر کاری آن به اندازه‌ی کافی سریع باشد و نیز اینکه برنامه‌ها به سرعت اجرا شوند و بسیاری از عملیات بدون هیچ وقفه‌ای انجام گردند. این موارد نمونه‌ای از چالش‌هایی است که امروزه طراحان نرم‌افزار با آن مواجه هستند.

۱-۱-۱- بسترهای محاسبات موازی

به منظور دستیابی به اجرای موازی برنامه‌ها، سخت‌افزارها بایستی بستری را فراهم نمایند که از اجرای همزمان چندین وظیفه پشتیبانی کند. به طور کلی، معماری‌های کامپیوتری می‌توانند از دو بُعد دسته‌بندی شوند. اولین بُعد تعداد جریان دستورالعمل‌ها[2] است که در یک معماری کامپیوتر خاص می‌توانند در یک لحظه در حال اجرا باشند. دومین بُعد تعداد جریان دادهایی[3] است که می‌توانند در یک لحظه از زمان مورد پردازش قرار گیرند. از اینرو یک طبقه‌بندی که به طبقه‌بندی Flynn [4] معروف است معرفی شد که در شکل ۱-۱ نشان داده شده است.

[1] Graphical User Interface (GUI)
[2] Instruction stream
[3] Data stream
[4] Flynn taxonomy



شکل ۱-۱ طبقه‌بندی Flynn

از دیدگاه این طبقه‌بندی، بسترهای کامپیوتری در ۴ دسته قرار می‌گیرند:

- یک ماشین با یک دستورالعمل و یک داده[۱] (SISD): کامپیوترهای سنتی ترتیبی هستند که هیچ گونه موازی سازی در سخت‌افزار ارائه نمی‌دهند. دستورالعمل‌ها به شکل ترتیبی اجرا می‌شوند. در هر چرخه‌ی زمانی[۲] (کلاک) فقط یک جریان داده توسط پردازنده پردازش می‌شود. نمونه‌هایی از این دسته کامپیوترها سیستم‌های دهه‌ی ۸۰ بودند.
- یک ماشین با چندین دستورالعمل و یک داده[۳] (MISD): قابلیت پردازش یک جریان داده از طریق جریان‌های داده‌ی متعدد به صورت همزمان را دارد. در بسیاری از موارد جریان‌های داده‌ی متعدد به منظور مفید بودن نیاز به جریان‌های داده‌ی متعدد دارند. از اینرو در حالت کلی، این کلاس از کامپیوترهای موازی بیشتر به عنوان یک مدل تئوری استفاده می‌شوند تا یک مدل عملی.
- یک ماشین با یک جریان دستورالعمل و داده‌های متعدد[۴] (SIMD): این ماشین‌ها کامپیوترهایی هستند که یک دستورالعمل قابلیت پردازش چندین جریان داده به صورت

[1] Single Instruction Single Data
[2] Clock cycle
[3] Multiple Instruction Single Data
[4] Single Instruction Multiple Data

همزمان را دارد. این کامپیوترها در کاربردهایی نظیر پردازش سیگنال‌های دیجیتالی عمومی، پردازش تصویر و نیز برنامه‌های چندرسانه‌ای صوتی و تصویری مفید می‌باشند. در ابتدا سوپرکامپیوترهایی که به عنوان پردازنده‌های آرایه‌ای[۱] یا پردازنده‌های برداری[۲] شناخته می‌شدند مانند Cray-1 این قابلیت‌ها را داشتند. تقریبا اکثر کامپیوترهای امروزی شکل خاصی از این قابلیت را پیاده‌سازی نموده‌اند.

- یک ماشین با چندین دستورالعمل و چندین داده[۳] (MIMD): ماشین‌هایی هستند که قابلیت اجرای چندین جریان دستورالعمل که بر روی جریان‌های داده‌ی متعدد و مجزا از هم عمل می‌کنند را دارند. این نوع سیستم‌ها پرکاربردترین نوع سیستم‌های موازی هستند. بسترهای جدید مانند سیستم‌های چند هسته‌ای که پرازنده‌های Intel® Core™ Duo از این نوع هستند در این گروه از ماشین‌ها قرار می‌گیرند.

با فرض داشتن کامپیوترهایی که SIMD یا MIMD هستند، طراحان نرم‌افزار می‌توانند از قابلیت موازی سازی این پردازنده‌ها در سطح داده و نیز در سطح وظیفه استفاده‌ی کارامدی داشته باشند.

۱-۱-۲- محاسبات موازی در پردازنده‌ها

در سال ۱۹۶۵، Gordon Moore مشاهده نمود که تعداد ترانزیستورهای[۴] موجود در نیمه‌رساناهای تولیدی تقریبا هر ۱۸ تا ۲۴ ماه به دو برابر می‌رسند که این ادعا با عنوان قانون مور[۵] شناخته می‌شود.

معماران کامپیوتر به منظور استفاده کارامد از منابع محاسباتی از تکنیک موازی سازی در سطح دستورالعمل[۶] استفاده نمودند تا کارایی پردازنده را افزایش دهند. موازی سازی در سطح دستورالعمل

[1] Array processor
[2] Vector processor
[3] Multiple Instruction Multiple Data
[4] Transistor
[5] Moore Law
[6] Instruction Level Parallelization (IPL)

که گاهی اوقات اجرای پویا[1] یا خارج از ترتیب[2] نیز نامیده می‌شود، این قابلیت را به پردازنده می‌دهد که ترتیب دستورالعمل‌ها را به شیوه‌ی بهینه‌ای جابه‌جا نمایند تا حفره‌های خط لوله[3] حذف شوند. هدف موازی سازی سطح دستوالعمل افزایش تعداد دستورالعمل‌هایی است که پردازنده در یک کلاک زمانی می‌تواند اجرا کند. به منظور کارا بودن این روش بایستی دستورالعمل‌هایی اجرای شوند که به تعداد زیاد و نیز مستقل از یکدیگر باشند. در حالت اجرای برنامه‌ها به صورت خارج از ترتیب، وابستگی‌های موجود بین دستورالعمل‌ها ممکن است تعداد دستورالعمل‌های فراهم برای اجرا را محدود نماید که نتیجه‌ی آن کاهش سطح اجرای موازی قابل اجرا خواهد بود. یک رویکرد متفاوت می‌تواند این باشد که ترتیب دستورالعمل‌ها را به نحوی تغییر داد که دستورالعمل‌های مستقل به طور همزمان اجرا شوند که نتیجه‌ی آن این خواهد بود که واحدهای اجرایی پردازنده به طور کامل در حال کار نگاه داشته شوند. در این حالت، دستورالعمل‌ها خارج از ترتیب اجرا خواهند شد. این نوع زمانبندی دستورالعمل‌های پویا توسط خود پردازنده انجام می‌شود و برای طراح نرم‌افزار کاملا شفاف است.

با رشد نرم‌افزارها، برنامه‌ها قابلیت اجرای چندین وظیفه به صورت همزمان را دارند. امروزه برنامه‌های اجرایی در سمت سرویس دهنده‌ها شامل تعدادی نخ[4] یا پردازش می‌باشند. به منظور پشتیبانی از این موازی سازی در سطح نخ، رویکردهای متفاوتی در هر دو راستای نرم‌افزاری و سخت‌افزاری ارائه شده است.

یک رویکرد به منظور بهره جُستن از این ذات همروند نرم‌افزارهای جدید استفاده از یک سیستم‌عامل چندوظیفه‌ای به صورت برش زمانی[5] است. این روش به طراحان اجازه می‌دهد که با برگ‌برگ[6] کردن اجراها و عملیات مختلف، تأخیرهای ناشی از ارتباطات ورودی-خروجی را مخفی نمایند. البته این مدل

[1] Dynamic execution
[2] Out-of-order
[3] Pipeline stall
[4] Thread
[5] Time-slice
[6] Interleave

اجازه‌ی اجرای موازی برنامه‌ها را نخواهد داد چرا که در یک لحظه فقط یک جریان دستورالعمل می‌تواند بر روی یک پردازنده اجرا شود.

رویکرد دیگر به منظور بهره‌گیری از موازی سازی سطح نخ افزایش تعداد پردازنده‌ها فیزیکی در کامپیوتر است. سیستم‌های چند پردازنده‌ای اجازه‌ی اجرای موازی واقعی را می‌دهند. نخ‌ها یا پردازش‌های متعدد می‌توانند به صورت همزمان بر روی پردازنده‌های مختلف اجرا شوند. تنها نکته‌ی موجود هزینه‌ی این سیستم‌ها است.

طراحان سخت‌افزارهای کامپیوتری به دنبال راهی هستند که بتوانند معماری پردازنده را با مدل موازی سازی در سطح نخ همگام نمایند. در بسیاری از موارد منابع یک پردازنده کمتر از توان‌شان[۱] مورد استفاده قرار می‌گیرند. یک پردازنده متشکل از تعدادی منابع مختلف است مانند وضعیت معماری[۲]، حافظه‌های عمومی پردازنده که به ثبات‌ها[۳] معروف هستند، ثبات‌های کنترل کننده‌ی وقفه[۴]، حافظه‌های نهان[۵]، کانال‌های ارتباطی[۶]، واحدهای اجرایی[۷] و تعدادی واحدهای دیگر. به منظور تعریف تعریف یک نخ، تنها وضعیت معماری مورد نیاز است. یک پردازنده‌ی منطقی می‌تواند با تکرار این فضای معماری ساخته شود. سپس منابع اجرایی می‌توانند بین پردازنده‌های منطقی مختلف به اشتراک گذاشته شوند. این تکنیک با عنوان چندنخی همزمان[۸] (SMT) شناخته می‌شود. پیاده‌سازی شرکت Intel از این تکنیک، با عنوان تکنولوژی اَبَرنخی[۹] (HT) شناخته می‌شود. این تکنولوژی باعث می‌شود یک پردازنده از دیدگاه برنامه به صورت چندین پردازنده دیده شود. این ویژگی به سیستم‌عامل و برنامه‌ها اجازه می‌دهد که چندین نخ را به نحوی بر روی پردازنده‌ها منطقی زمانبندی

[1] Underutilize
[2] Architecture state
[3] Register
[4] Interrupt
[5] Cache
[6] Bus
[7] Execution unit
[8] Simultaneous Multi-Threading (SMT)
[9] Hyper-Threading Technology (HT)

کنند که گویا بر روی سیستم‌ها چندپردازنده‌ای هستند. از دیدگاه یک ریزمعماری، دستورالعمل‌های یک پردازنده‌ی منطقی پایدار هستند و به صورت همزمان بر روی منابع اجرایی اشتراکی اجرا می‌شوند. به عبارتی نخ‌های متعدد می‌توانند زمانبندی شوند اما از آنجایی که منابع اجرایی مشترک هستند، تعیین چگونگی و زمان برگ‌برگ نمودن اجرای چندین نخ به عهده‌ی معماری پردازنده است. زمانی که یک نخ به وقفه‌ای برخورد می‌کند، اجازه‌ی اجرا به نخ دیگری داده می‌شود. این وقفه‌ها می‌توانند حاصل از یک عدم اصابت حافظه نهان[۱] یا پیش‌بینی اشتباه یک انشعاب[۲] باشد.

قدم منطقی بعد از پردازنده‌های چندنخی همزمان، پردازنده‌های چند هسته‌ای[۳] هستند. این پردازنده‌ها از قابلیت چندین پردازنده‌ای بر روی یک تراشه[۴] (CMP) استفاده می‌کنند. سازندگان پردازنده‌های چند هسته‌ای از تکنولوژی متفاوتی نسبت به پردازنده‌های قبلی به منظور جاسازی چندین هسته‌ی اجرایی[۵] بر روی یک پردازنده استفاده می‌کنند. هسته‌های اجرایی دارای مجموعه منابع اجرایی و ساختاری مختص خودشان هستند. بر حسب نوع طراحی ممکن است این هسته‌ها دارای حافظه‌های نهان بزرگی باشند که بین آنها با اشتراک گذاشته می‌شود. علاوه براین، این هسته‌ها ممکن است با تکنولوژی چند نخی همزمان (SMT) ترکیب شوند که نتیجه‌ی آن افزایش تعداد پردازنده‌های منطقی به تعداد دو برابر تعداد هسته‌های اجرایی خواهد بود. معماری‌های مختلف پردازنده‌ها در شکل ۱-۲ نشان داده شده است [AKH 06].

[1] Cache miss
[2] Branch misspredict‌ion
[3] Multi-core processor
[4] Chip Multiprocessing (CMP)
[5] Execution core