

ویژگی های دمایی پردازنده های چند هسته ای چند نخه همزمان* به منظور مدیریت و کنترل دمای پویا

باقر سلامی^۱، حمید نوری^۲، وحید کیخوایی^۳

^۱ دانشکده مهندسی، دانشگاه فردوسی، مشهد، ایران
bagher.salami@stu-mail.um.ac.ir

^۲ دانشکده مهندسی، دانشگاه فردوسی، مشهد، ایران
hnoori@um.ac.ir

^۳ دانشکده مهندسی، دانشگاه فردوسی، مشهد، ایران
vahid.keykhaey@stu-mail.um.ac.ir

چکیده

افزایش دمای قطعات سیستم های کامپیوتری مساله ای است که روی پارامترهایی چون قابلیت اطمینان سیستم، کارایی، هزینه، طول عمر و... تاثیر گذار می باشد و استفاده از راه کارهای کارا برای مدیریت دما بسیار سودمند می باشد. تکنیک های مدیریت دمایی که تا کنون برای پردازنده های فاقد این ویژگی ارائه شده اند، در این معماری به کارایی حداکثر نمی رسند و نیاز به اعمال تغییراتی در آنها به منظور افزایش کارایی می باشد. همچنین الگوریتم هایی که به منظور مدیریت دمای پردازنده های چند نخه همزمان ارائه شده اند همگی به صورت شبیه سازی شده ارائه شده اند و به دلیل اینکه هر کدام از الگوریتم های شبیه سازی شده یک سری فرض ها را در نظر گرفته اند که ممکن است نتوان روی سیستم های واقعی اعمال نمود، روی سیستم های واقعی چند نخه همزمان قابل استفاده نیستند. در این مقاله ما نتایج آزمایش هایی که روی سیستم های واقعی چندنخه همزمان بدست آورده ایم را ارائه می نمایم که کمک می کند با در نظر گرفتن این نتایج تجربی و اعمال آنها روی الگوریتم های شبیه سازی شده موجود، بتوان از آنها روی سیستم های واقعی چند نخه همزمان استفاده نمود.

کلمات کلیدی

مدیریت دمای پویا، پردازنده های چند هسته ای، چند نخه همزمان، کارایی، دما

* Simultaneous Multithreading (SMT)

۱- مقدمه

امروزه نیاز به محاسبات با حجم بالا روز به روز بیشتر و بیشتر می‌شود و برای فراهم نمودن سیستم‌های با توان محاسباتی بالا اندازه ترازیستور به طور پیوسته کوچک‌تر می‌شود تا تعداد بیشتری ترازیستور در یک تراشه مجتمع شود و مدارهای محاسباتی پیچیده در فضای کوچک فراهم شود که این امر باعث افزایش پیوسته چگالی توان مصرفی (توان مصرفی تقسیم بر مساحت) شده است و افزایش چگالی توان مصرفی موجب افزایش دما می‌شود. همچنین افزایش توان مصرفی باعث افزایش جریان و توان ناشی می‌شود و افزایش توان ناشی باعث افزایش دما می‌شود. از سوی دیگر افزایش دما باعث افزایش توان ناشی و توان مصرفی می‌شود. بنابراین افزایش توان ناشی و دما یک رابطه دوطرفه داشته و باعث می‌شود در مدت کوتاهی دما بسیار بالا رفته و باعث خرابی قطعه شود.

الگوریتم‌های موجود آرایه شده برای پردازنده‌های غیر چند نخه همزمان، روی پردازنده‌های چند نخه همزمان باعث افت کارایی و عدم مدیریت بهینه دما می‌شوند که این به این دلیل است که با فعال کردن قابلیت چند نخه همزمان روی پردازنده‌ای به عنوان مثال ۴ هسته‌ای پردازنده تبدیل به پردازنده‌ی ۸ هسته‌ای می‌شود که از نظر فیزیکی ۴ هسته داریم و از نظر منطقی ۸ هسته که باعث می‌شود هر هسته فیزیکی بتواند دو وظیفه را همزمان اجرا کند. در پردازنده‌های چند نخه همزمان هر هسته فیزیکی با دوبرابر کردن بخش کمی از پردازنده مانند رجیسترها امکان اجرای دو وظیفه همزمان فراهم می‌شود و بخش عمده‌ای از واحد‌های پردازنده به صورت اشتراکی مورد استفاده دو وظیفه همزمان هر هسته قرار می‌گیرد. بنابراین با فعال نمودن قابلیت چند نخه همزمان تعداد هسته‌های ما از نظر منطقی دو برابر می‌شوند و از نظر فیزیکی دوبرابر نمی‌شوند که این نکته در الگوریتم‌های موجود آرایه شده برای پردازنده‌های غیر چند نخه همزمان در نظر گرفته نشده و معمولاً باعث افت کارایی و عدم مدیریت بهینه دما می‌شود. در [۲۱] یکی از الگوریتم‌های مدیریت دمایی غیر چند نخه همزمان روی پردازنده‌های چند نخه همزمان مورد بررسی قرار گرفته و اثبات شده که روی سیستم‌های چند نخه همزمان به کارایی حداکثر نمی‌رسد و نیاز به تغییر دارد.

الگوریتم‌های ارائه شده در زمینه مدیریت دمایی پردازنده‌های چند نخه همزمان [۲۰، ۲۳ و ۲۴] تا کنون همگی به صورت شبیه‌سازی شده ارائه شده‌اند و فقط الگوریتم‌هایی مانند الگوریتم ارائه شده در [۲۱] روی سیستم واقعی آرایه و تست شده که با هدف مدیریت دما نمی‌باشند و معیار کارایی را مد نظر قرار داده است.

در این مقاله کارایی و رفتار و عملکرد حرارتی پردازنده‌های چند نخه همزمان بررسی می‌شود. این نتایج به محققین مدیریت دمایی پردازنده‌های چند نخه همزمان کمک می‌کند تا با در نظر گرفتن این ویژگی

ها در الگوریتم‌های پیشنهادی خود حداکثر بهینه‌سازی را در کارایی و دما داشته باشند.

باقیمانده مقاله به این ترتیب سازماندهی شده که روش‌های موجود مدیریت دما در قسمت ۲ ارائه شده است. در قسمت ۳ جزئیات پیاده‌سازی بیان می‌شود. در قسمت ۴ نتایج آزمایش‌ها بیان می‌شود و نتیجه‌گیری در قسمت ۵ انجام شده است.

۲- پیشینه

به منظور مدیریت دما، تکنیک‌های متعددی ارائه شده است [۱، ۴، ۵، ۶، ۹، ۱۰، ۱۳، ۱۴، ۱۵، ۱۷، ۲۰، ۲۴] که شامل دو دسته تکنیک‌های ایستا و پویا می‌شوند. هرکدام از دسته‌های ایستا و پویا از مکانیزم‌های سخت‌افزاری، نرم‌افزاری و یا ترکیبی از سخت‌افزار و نرم‌افزار استفاده می‌نمایند.

تکنیک‌های ایستای مدیریت دما شامل دو دسته سخت‌افزاری و نرم‌افزاری می‌شوند. یک نمونه تکنیک ایستای سخت‌افزاری طراحی نقشه پردازنده به صورت آگاه از دما می‌باشد. دمایی بیشینه یک بلاک در یک تراشه نه تنها به چگالی توان مصرفی (توان مصرفی در واحد سطح) خود آن بلاک بستگی دارد بلکه به چگالی توان مصرفی بلاک‌های مجاور آن نیز بستگی دارد. در نتیجه چگونگی قرار گرفتن بلاک‌ها در نقشه یک پردازنده تأثیر زیادی در دمایی بیشینه قطعات دارد [۲۲]. تکنیک ایستای دیگر مدیریت دما، نرم‌افزاری می‌باشد که در زمان کامپایل انجام می‌شود. یکی از الگوریتم‌های مدیریت دما توسط کامپایلر TempNOP نام دارد که زمانی که با توجه به کد برنامه یک فشار گرمایی شدید پیش‌گویی شود در برنامه تعدادی دستور NOP قرار داده می‌شود [۳].

تکنیک‌ها و الگوریتم‌هایی که برای مدیریت پویای دما ارائه شده‌اند شامل دو دسته می‌شوند یک دسته فقط برای پردازنده‌های فاقد قابلیت چند نخه همزمان ارائه شده‌اند [۱ و ۴ و ۶ و ۱۳ و ۱۴ و ۱۷ و ۱۸] و دسته دیگر برای مدیریت دمایی پردازنده‌های چند نخه همزمان ارائه شده‌اند [۵ و ۲۰ و ۲۳ و ۲۴].

الگوریتم ارائه شده در [۱] ترکیب تکنیک پویای مدیریت دمایی سخت‌افزاری، تنظیم پویای ولتاژ-فرکانس و تکنیک نرم‌افزاری مهاجرت وظیفه می‌باشد. در این مقاله به منظور کاهش تعداد مهاجرت وظایف در سیستم در انتخاب هسته مقصد دمایی هسته‌های مجاور نیز در نظر گرفته شده است که باعث می‌شود کارایی افزایش یابد. الگوریتم ارائه شده در [۴] به منظور مدیریت دما نیز از ترکیب تنظیم پویای ولتاژ-فرکانس و مهاجرت وظیفه استفاده نموده با این تفاوت که از تنظیم پویای ولتاژ-فرکانس محلی به جای سراسری استفاده نموده است. بعضی دیگر از الگوریتم‌ها مانند الگوریتم ارائه شده در [۶] مهاجرت وظیفه را با تکنیک توقف-اجرا ترکیب نموده‌اند. استفاده از تکنیک توقف-اجرا نسبت به تکنیک تنظیم پویای ولتاژ-فرکانس دما را سریعتر کاهش می‌دهد ولی کارایی را بیشتر کاهش می‌دهد.

منبع داغ شده باشد، خیلی زیاد نخواهد شد. بنابراین اگر وظایف به گونه ای به هسته ها انتصاب پیدا کنند که منابع مختلفی از هسته ها را مورد استفاده قرار دهند ودمای آن منابع را بالا ببرند، کار انجام شده در واحد زمان افزایش پیدا می نماید. الگوریتم ارائه شده در [۲۱] با هدف افزایش کارایی پردازنده های چند نخه همزمان به این طریق عمل می نماید که با استفاده از اطلاعات شمارنده کارایی ترکیب مناسبی از برنامه ها را روی هسته ها اجرا می نماید تا کارایی افزایش یابد ولی به افزایش یا کاهش دما توجهی نشده است.

۳- جزئیات پیاده سازی

در این قسمت جزئیات سخت افزار و نرم افزار مورد استفاده بیان می شود.

۳-۱- سخت افزار مورد تست

برای انجام تمام آزمایش ها از یک سیستم 64_x86 دارای خاصیت چند نخه همزمان شرکت اینتل استفاده نمودیم. پردازنده مورد استفاده Core i7-2600 با فرکانس ۳.۴ گیگاهرتز می باشد. حافظه اصلی سیستم ۸ گیگابایت و سیستم عامل مورد استفاده لینوکس با نسخه هسته ۳.۲.۰ می باشد.

برای تنظیمات فرکانس از زیر سیستم cpufreq موجود در هسته لینوکس استفاده و آزمایش ها در فرکانس ثابت ۳.۴ و دمای محیط یکسان انجام شده است. سرعت خنک کننده پردازنده به منظور انجام تمام آزمایش ها تحت یک عملیات خنک سازی یکسان، ثابت نگه داشته شده است. فعال و غیر فعال نمودن قابلیت چند نخه همزمان پردازنده از دو طریق سیستم عامل و بایاس انجام شده است.

۳-۲- بار کاری نرم افزاری

برنامه های محک^۵ مورد استفاده در آزمایش ها gcc, mcf, hammer محک SPEC CPU2006 می باشد. که این سه برنامه با آزمایش برنامه های مختلف انتخاب شده اند، تا برنامه هایی از ترکیبات سرد و داغ داشته باشیم. برنامه hammer به عنوان برنامه ی داغ و دو برنامه دیگر جزء دسته سرد هستند. هر برنامه با ۵ مرتبه تکرار انجام شده اند تا یک برنامه روی یک هسته به دمای وضعیت پایدار^۶ برسد. منظور از دمای حالت پایدار، حداکثر دمایی است که چنانچه برنامه بطور نا متناهی در حال اجرا باشد، دمای هسته ای که آن برنامه را اجرا می کند، از آن حد گذر نکند. [۱۵].

۳-۳- اندازه گیری دما

برای دریافت اطلاعات دمایی پردازنده از حسگرهای تعبیه شده در پردازنده استفاده شده که این کار توسط ابزار lm-sensor [۲۵] انجام شده است. محدودیتی که در دریافت اطلاعات دمایی توسط حسگر داریم، این است که حسگرهای دمایی، حداقل مدت ۱ ثانیه نیاز دارند

دسته دیگر الگوریتم ها با استفاده از زمانبندی وظایف به مدیریت دما می پردازند. در [۱۴] تلاش شده با استفاده از زمانبندی وظایف تعداد تخطی ها از حد آستانه کاهش یابد. این الگوریتم بر اساس این اصل می باشد که اگر دو وظیفه داشته باشیم که یکی داغ و دیگری سرد باشد، اجرای وظیفه داغ قبل از اجرای وظیفه سرد منجر به دمای نهایی کمتری نسبت به اجرای ترتیب معکوس دارد. این الگوریتم در هر گام داغترین وظیفه ای که اجرای آن منجر به تخطی از حد آستانه دما نمی شود را برای اجرا انتخاب می نماید.

در [۲] و [۷] و [۸] و [۱۱] و [۱۲] و [۱۶] و [۱۸] و [۱۹] ویژگی های پردازنده های چند نخه همزمان مورد بررسی قرار گرفته شده است. دسته دیگر الگوریتم ها برای پردازنده های چند نخه همزمان ارائه شده اند. در [۵] ثابت شده است که الگوریتم ارائه شده با نام تاخیر نخ که برای برنامه های که به صورت موازی با مدل برنامه نویسی موازی سدی^۴ روی سیستم های غیر چند نخه همزمان اجرا می شوند روی سیستم های دارای قابلیت چند نخه همزمان به کارایی حداکثر نمی رسند، بنابراین الگوریتم تاخیر نخ را طوری تغییر داده تا برای سیستم های دارای قابلیت چند نخه همزمان به کارایی حداکثر برسند. تنها مشکل این روش این است که الگوریتم ارائه شده در این مقاله یک سری فرضیاتی از جمله گرفتن نقاط ملاقات و شمارش این نقاط به صورت سخت افزاری یا نرم افزاری، در نظر گرفته است که در عمل و سیستم های واقعی این فرضیات برقرار نمی باشد.

در [۲۰] نیز یک الگوریتم مدیریت دما برای پردازنده های چند نخه همزمان ارائه و در یک محیط شبیه سازی شده مورد بررسی و آزمایش قرار گرفته شده است. در این الگوریتم بر اساس دستیابی وظایف به رجیستر فایل صحیح و ممیز شناور تعیین می شود که کدام رجیستر فایل دارای دسترسی بالا بوده و دارای شرایط بحرانی می باشد. سپس برای آن واحد بحرانی به ازای تمام وظایف آن دسته تصمیم گیری می شود که هر وظیفه چه نقشی در بالابردن دسترسی به آن واحد بحرانی داشته اند و وظیفه ها بر اساس تاثیر در بالا بردن دمای واحد بحرانی تقسیم بندی شده و به ترتیب مشخص می شوند. بنابراین برای دوره بعدی زمانبندی وظایفی که کمترین تاثیر را بر واحد بحرانی دارند، برای اجرا انتخاب می شوند.

الگوریتم ارائه شده در [۲۴] که به صورت شبیه سازی شده مورد آزمایش قرار گرفته شده است، برای سیستم های چند نخه همزمان برای افزایش استفاده منابع توسط پردازنده ها به این طریق که وظایف در حال اجرای هر هسته از منابعی که مکمل یکدیگر می باشند استفاده نمایند. این الگوریتم دو قسمت عمده دارد. اولین بخش انتصاب وظایف و قسمت دوم مهاجرت وظیفه می باشد. بخش انتصاب وظیفه بر این اساس می باشد که اگر حتی یک قسمت از واحد های مختلف پردازنده مانند رجیستر فایل، حافظه نهان و... به دمای بحرانی خود برسند، تمام بخش های یک هسته باید متوقف شوند و اگر منابع بیشتر از یکی داغ شوند زمان سرد شدن نسبت به حالتی که فقط یک

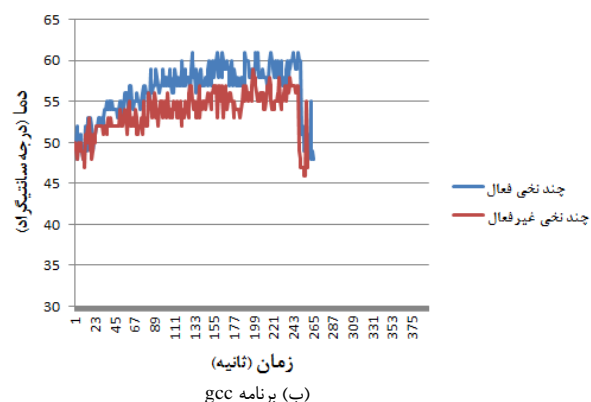
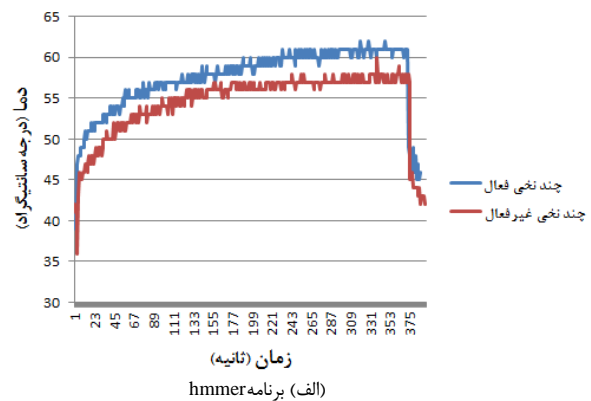
تا در برابر تغییرات دمایی واکنش نشان دهند، بنابراین دوره های نمونه گیری دمایی ما ۱ ثانیه می باشد.

۴- نتایج

در این قسمت یک سری تجربیاتی را که با انجام آزمایش ها روی پردازنده های دارای قابلیت چند نخه همزمان بدست آورده ایم، ارائه می نماییم که کمک می کند تا محققین با در نظر گرفتن این نکات، الگوریتم های موثرتری برای مدیریت دمایی این پردازنده ها ارائه کنند.

۴-۱- تغییر دما با فعال کردن چند نخه از بایاس

اولین ویژگی دمایی این پردازنده ها این است که برای بار کاری پایین یعنی زمانی که تعداد برنامه ها از تعداد هسته ها کمتر باشد، در دو حالت (۱) قابلیت چند نخه همزمان پردازنده فعال باشد و (۲) قابلیت چند نخه از طریق بایاس غیر فعال باشد دما متفاوت است. بدین منظور ما یک مرتبه قابلیت چند نخه را فعال کرده و یک برنامه را اجرا کرده و دمایی پردازنده را ثبت نمودیم. سپس قابلیت چند نخه را از طریق بایاس غیر فعال و همان برنامه را اجرا می کنیم و مجدداً دمایی پردازنده را ثبت می کنیم. این کار روی برنامه های مختلف انجام شده که نمودار دمایی دو برنامه hmc و gcc در دو حالت فعال یا غیر فعال بودن چند نخه همزمان در شکل (۱) نشان داده شده است.

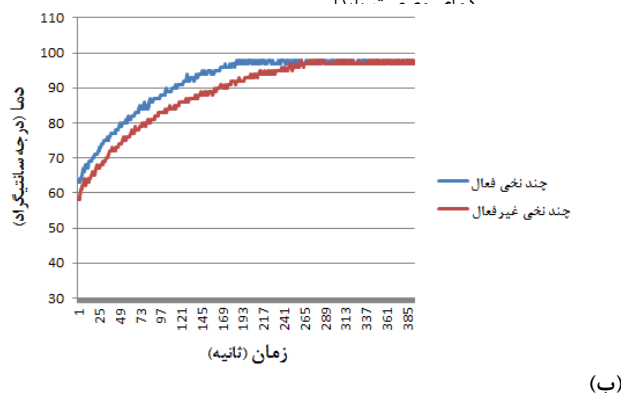
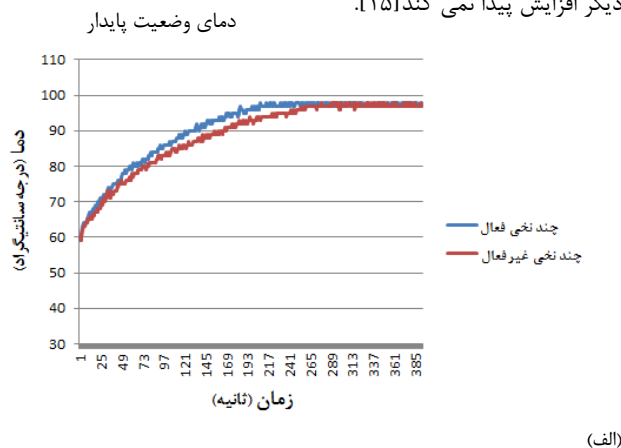


شکل (۱): نمودار دمایی پردازنده با اجرای برنامه های hmc و gcc در حالت فعال و غیر فعال بودن چند نخه همزمان

همان طور که در شکل (۱) مشاهده می شود با فعال نمودن قابلیت چند نخه همزمان از طریق بایاس شاهد افزایش ۳ تا ۵ درجه ای دمایی هسته ها هستیم و زمان اجرای برنامه ها حدود ۱٪ کاهش می یابد به عبارتی افزایش کارایی ۱٪ خواهیم داشت. این افزایش دما می تواند به این دلیل باشد که با فعال شدن چند نخه همزمان، یک سری سخت افزار و مدارهایی که اجرای چند نخه همزمان را فراهم می کنند فعال شده که باعث افزایش توان و در نتیجه باعث افزایش دما شده است. در بار کاری پایین با بررسی برنامه های مختلف با فعال شدن چند نخه همزمان زمان اجرا حدود ۱٪ کاهش می یابد. بنابراین در شرایطی که بار کاری موجود از تعداد هسته های فیزیکی کمتر می باشد اگر قابلیت چند نخه همزمان غیر فعال شود دما حدود ۳ تا ۵ درجه کاهش می یابد. و فقط زمان اجرا حدود ۱٪ افزایش می یابد.

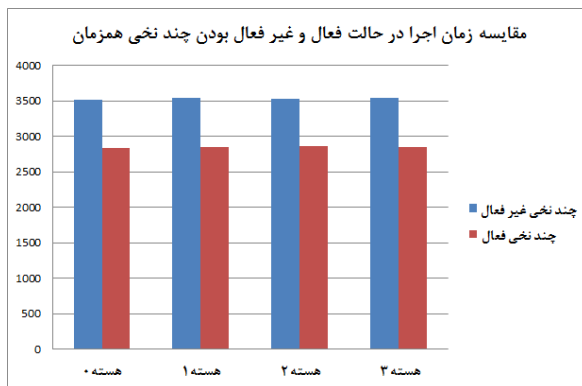
۴-۲- زمان رسیدن به دمایی وضعیت پایدار

ویژگی دمایی دیگر پردازنده های چند نخه همزمان در زمان رسیدن به دمایی وضعیت پایدار می باشد. دمایی وضعیت پایدار دمایی است که یک برنامه اگر به طور نامتناهی اجرا شود دمایی آن از این مقدار دیگر افزایش پیدا نمی کند [۱۵].



شکل (۲): الف) اجرای دو برنامه hmc و gcc روی هسته ۱ و روی هر یک از سه هسته دیگر یک نسخه از برنامه hmc ب) اجرای دو برنامه hmc و gcc روی هسته ۳ و روی هر یک از سه هسته دیگر یک نسخه از برنامه hmc

شدن چند نخه نسبت به غیر فعال بودن چند نخه حدود ۱۹٪ می باشد.



شکل (۳): زمان اجرای چهار نسخه برنامه hmmer و یک نسخه gcc در حالت فعال و غیر فعال بودن چند نخه همزمان

در این بخش این آزمایش را انجام دادیم که چهار نسخه برنامه hmmer و یک نسخه gcc را در حالت فعال و غیر فعال بودن چند نخه همزمان اجرا نمودیم و زمان اجرا را در هر حالت ثبت کردیم. ابتدا دو برنامه hmmer و gcc روی هسته صفر و روی بقیه هسته ها هر کدام یک نسخه از برنامه hmmer را اجرا نمودیم و سپس در دفعات بعد زوج برنامه را روی هسته های ۱ و ۲ و ۳ اجرا نمودیم که نتایج در شکل (۳) نشان داده شده است. زمان اجرا در حالت بار کاری کامل یعنی زمانی که تعداد وظایف از تعداد هسته ها بیشتر باشد، با فعال شدن چند نخه همزمان حدود ۱۹٪ کاهش پیدا می نماید.

بنابراین باید الگوریتم ها بررسی شوند که اگر وضعیت بار کاری کامل می باشد افزایش چشمگیر کارایی خواهیم داشت و اگر بار کاری پایین می باشد بهتر است قابلیت چند نخه همزمان پردازنده از طریق بایاس غیر فعال شود زیرا در بهبود کارایی تاثیری نخواهد داشت بلکه دما را ۳ تا ۵ درجه افزایش می دهد.

۵- نتیجه

در این مقاله ما نتایج آزمایش هایی که روی سیستم های واقعی چندنخی همزمان انجام دادیم را ارائه نمودیم. اولین ویژگی که بدست آوردیم این است که با فعال نمودن قابلیت چند نخه همزمان از طریق بایاس شاهد افزایش ۳ تا ۵ درجه ای دمای هسته ها هستیم. ویژگی دوم این است که با فعال شدن چند نخه همزمان پردازنده زود تر به دمای وضعیت پایدار می رسد. ویژگی سوم این است که مقدار دمای وضعیت پایدار با فعال یا غیر فعال شدن چند نخه همزمان تغییر نمی کند. آخرین ویژگی پردازنده های چند نخه همزمان که بدست آوردیم این است که در حالت بار کاری کامل یعنی وقتی تعداد وظایف از تعداد هسته ها بیشتر باشد با فعال شدن چند نخه همزمان زمان اجرا حدود

آزمایشی که ما در این قسمت انجام دادیم به این صورت است که روی یک هسته دو برنامه hmmer و gcc و روی هر کدام از ۳ هسته دیگر یک نسخه از برنامه hmmer را اجرا نمودیم.

در شکل (۲) زمان رسیدن پردازنده به وضعیت پایدار را در بار کاری کامل یعنی زمانی که تعداد برنامه ها از تعداد هسته ها بیشتر باشد روی هسته های مختلف، نشان می دهد. مشاهده می شود با فعال شدن چند نخه همزمان در یک بار کاری ثابت پردازنده زود تر به دمای وضعیت پایدار می رسد و اگر چند نخه همزمان غیر فعال شود دیرتر به دمای وضعیت پایدار می رسیم. همچنین همانطور که در شکل (۲) مشاهده می شود، در هسته های مختلف نیز زمان رسیدن به دمای وضعیت پایدار متفاوت می باشد.

دمای وضعیت پایدار یکی از پارامترهای دمایی است که در بسیاری از الگوریتم های مدیریت دما مورد استفاده قرار گرفته است. یکی از الگوریتم هایی که بر مبنای دمای وضعیت پایدار می باشد الگوریتم PDTM [۱۵] می باشد. الگوریتم PDTM که زمان رسیدن به وضعیت پایدار آن به صورت آفلاین محاسبه و مورد استفاده قرار گرفته است به این نکته توجه نشده است که فعال شدن یا غیر فعال شدن قابلیت چند نخه همزمان، مقدار زمان رسیدن به وضعیت پایدار را تغییر می دهد. بنابراین اگر به این نکته که زمان رسیدن به دمای وضعیت پایدار در حالت فعال و غیر فعال بودن چند نخه همزمان، متفاوت می باشد توجه نشود این الگوریتم در حالت غیر فعال بودن چند نخه بسیار کارا می باشد ولی با فعال شدن چند نخه همزمان کارایی اش را از دست می دهد. بنابراین این نکته باید در الگوریتم های مدیریت دمای پردازنده های چند نخه مورد توجه قرار گیرد.

۴-۳- مقدار دمای وضعیت پایدار

ویژگی دیگری که با انجام آزمایش ها بدست آوردیم مقدار دمای وضعیت پایدار در بار کاری کامل است که با فعال یا غیر فعال شدن چند نخه همزمان تغییر نمی کند. در شکل (۲) مشاهده می شود که زمان رسیدن به دمای وضعیت پایدار متفاوت است ولی مقدار دمای وضعیت پایدار در هر دو حالت مساوی می باشد. بنابراین با فعال و غیر فعال شدن چند نخه همزمان در بار کاری کامل مقدار دمای وضعیت پایدار تغییر نمی کند. بنابراین این ویژگی در الگوریتم هایی که از این پارامتر استفاده می کنند باید لحاظ شود.

۴-۴- بهبود کارایی

آزمایش دیگری که ما در این قسمت انجام دادیم بررسی مقدار بهبود کارایی با فعال و غیر فعال شدن چند نخه در حالت بار کاری کامل می باشد. با بررسی ما در قسمت ۴-۱ در حالت بار کاری پایین، با فعال شدن چند نخه همزمان کارایی حدود ۱٪ بهبود می یابد ولی اگر تعداد برنامه ها و وظایف از تعداد هسته ها بیشتر باشد، بهبود کارایی با فعال

[15] Yeo, Inchoon, Chih Chun Liu, and Eun Jung Kim. "Predictive dynamic thermal management for multicore systems." Proceedings of the 45th annual Design Automation Conference. ACM, 2008.

[16] Wu, Wei, et al. "Efficient power modeling and software thermal sensing for runtime temperature monitoring." ACM Transactions on Design Automation of Electronic Systems (TODAES) 12.3 (2007): 25.

[17] Choi, Jeonghwan, et al. "Thermal-aware task scheduling at the system software level." Proceedings of the 2007 international symposium on Low power electronics and design. ACM, 2007.

[18] Michaud, Pierre, et al. "A study of thread migration in temperature-constrained multicores." ACM Transactions on Architecture and Code Optimization (TACO) 4.2 (2007): 9.

[19] Donald, James, and Margaret Martonosi. "Techniques for multicore thermal management: Classification and new exploration." ACM SIGARCH Computer Architecture News 34.2 (2006): 78-88.

[20] Donald, James, and Margaret Martonosi. "Leveraging simultaneous multithreading for adaptive thermal control." Proc. of the Second Workshop on Temperature-Aware Computer Systems. 2005.

[21] Bulpin, James R., and Ian A. Pratt. "Hyper-threading aware process scheduling heuristics." Proc. of the USENIX Annual Technical Conf. 2005.

[22] Han, Yongkui, Israel Koren, and Csaba Andras Moritz. "Temperature aware floorplanning." Workshop on Temperature Aware Computer Systems. 2005.

[23] Donald, James, and Margaret Martonosi. "Temperature-aware design issues for SMT and CMP architectures." Workshop on complexity-effective design. 2004.

[24] Gomaa, Mohamed, Michael D. Powell, and T. N. Vijaykumar. "Heat-and-run: leveraging SMT and CMP to manage power density through the operating system." ACM SIGARCH Computer Architecture News. Vol. 32. No. 5. ACM, 2004.

[25] Lm sensors linux hardware monitoring: <http://www.lm-sensors.org>.

۱۹٪ کاهش می یابد ولی اگر بار کاری پایین باشد زمان اجرا تغییر چشمگیری پیدا نمی کند و حدود ۱٪ کاهش می یابد. این نتایج به محققین مدیریت دمای پردازنده های چند نخه همزمان کمک می کند تا با در نظر گرفتن این ویژگی ها در الگوریتم های پیشنهادی خود حداکثر بهینه سازی را در کارایی و دما داشته باشند.

مراجع

[1] Liu, Guanglei, Ming Fan, and Gang Quan. "Neighbor-aware dynamic thermal management for multi-core platform." Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012. IEEE, 2012.

[2] S. Zhuravlev, J. Carlos Saez, S. Blagodurov, A. Fedorova, M. Prieto, "Survey of Energy-Cognizant Scheduling Techniques." Parallel and Distributed Systems, IEEE Transactions on 05.01 (2012): 1-1.

[3] Kong, Joonho, Sung Woo Chung, and Kevin Skadron. "Recent thermal management techniques for microprocessors." ACM Computing Surveys (CSUR) 44.3 (2012): 13.

[4] Hanumaiah, Vinay, Sarma Vrudhula, and Karam S. Chatha. "Performance optimal online dvfs and task migration techniques for thermally constrained multi-core processors." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 30.11 (2011): 1677-1690.

[5] Cai, Qiong, et al. "Thread shuffling: Combining DVFS and thread migration to reduce energy consumptions for multi-core systems." Low Power Electronics and Design (ISLPED) 2011 International Symposium on. IEEE, 2011.

[6] Yun, Buyoung, Kang G. Shin, and Shige Wang. "Thermal-Aware Scheduling of Critical Applications Using Job Migration and Power-Gating on Multi-core Chips." Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on. IEEE, 2011.

[7] Saini, Subhash, et al. "The impact of hyper-threading on processor resource utilization in production applications." High Performance Computing (HiPC), 2011 18th International Conference on. IEEE, 2011.

[8] Schöne, Robert, Daniel Hackenberg, and Daniel Molka. "Simultaneous multithreading on x86_64 systems: an energy efficiency evaluation." Proceedings of the 4th Workshop on Power-Aware Computing and Systems. ACM, 2011.

[9] Ge, Yang, Parth Malani, and Qinru Qiu. "Distributed task migration for thermal management in many-core systems." Design Automation Conference (DAC), 2010 47th ACM/IEEE. IEEE, 2010.

[10] Wu, Guowei, et al. "Task Allocation and Migration Algorithm for Temperature-constrained Real-time Multi-Core Systems." Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing. IEEE Computer Society, 2010.

[11] Almeida, Gabriel Marchesan, et al. "Evaluating the impact of task migration in multi-processor systems-on-chip." Proceedings of the 23rd symposium on Integrated circuits and system design. ACM, 2010.

[12] Youssef, Ahmed, et al. "On the Power Management of Simultaneous Multithreading Processors." Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 18.8 (2010): 1243-1248.

[13] Rangan, Krishna K., Gu-Yeon Wei, and David Brooks. "Thread motion: fine-grained power management for multi-core systems." ACM SIGARCH Computer Architecture News. Vol. 37. No. 3. ACM, 2009.

[14] Yang, Jun, et al. "Dynamic thermal management through task scheduling." Performance Analysis of Systems and software, 2008. ISPASS 2008. IEEE International Symposium on. IEEE, 2008.

زیر نویس ها

¹Leakage Power

²Temperature Aware Floorplanning

³Stop-go

⁴Barrier

⁵Benchmark

⁶Steady State