

بهینه‌سازی انتساب برخط کارها در سیستم‌های پردازشی چندهسته‌ای برای مدیریت پویای گرمایش

فرناز نیک‌نیا^۱ وصال حکمی^۲

۱- دانش آموخته کارشناسی ارشد- دانشکده مهندسی کامپیوتر- دانشگاه علم و صنعت ایران - تهران- ایران

niknia_farnaz@alumni.iust.ac.ir

۲- استادیار- دانشکده مهندسی کامپیوتر- دانشگاه علم و صنعت ایران - تهران- ایران

vhakami@iust.ac.ir

چکیده: افزایش دما در پردازنده‌های چندهسته‌ای سبب افزایش توان مصرفی و کاهش طول عمر مفید پردازنده می‌شود. بهینه‌سازی انتساب کارها به هسته‌ها، راهکاری کارآمد برای مدیریت حرارتی این نوع پردازنده‌ها است. چالش اصلی در مسأله انتساب برخط، عدم قطعیت‌هایی است که محیط عملیاتی چندهسته‌ای با آنها روبرو می‌باشد که شامل: ورود تصادفی کارها به سیستم، جفت شدن تصادفی کارها با یکدیگر و تغییرات پروفایل دمایی تراشه در زمان می‌شود. در این مقاله، راهکاری برای انتساب برخط کارها به هسته‌ها ارائه شده است که در آن برای لحاظ سیستماتیک عوامل تصادفی، مسأله‌ی انتساب در قالب یک فرآیند تصمیم‌سازی مارکوف زمان-پیوسته مدل‌سازی می‌شود. برای محاسبه سیاست بهینه انتساب کارها، الگوریتمی مبتنی بر یادگیری تقویتی زمان-پیوسته پیشنهاد شده است که می‌تواند بی‌نیاز از دانش از پیش در خصوص مدل احتمالاتی سیستم، تنها از طریق دریافت بازخوردهای واقعی در زمان اجرا، انتساب برخط بهینه را تعیین نماید. همچنین، جهت تضمین کارکرد در ابعاد بالای مؤلفه-های وضعیت سیستم (تعداد زیاد حسگرهای دمایی و صف نامتناهی)، از تکنیک تقریب تابع برای رفع معضل تعدیت استفاده شده است. نتایج شبیه‌سازی نشان می‌دهند که راهکار پیشنهادی به اندازه‌ی ۶ درجه‌ی سانتی‌گراد میانگین دمای بیشینه‌ی سیستم و به میزان ۶۶ میلی‌ثانیه زمان سرویس کارها را کاهش می‌دهد.

واژه‌های کلیدی: پردازنده‌های چندهسته‌ای، انتساب برخط کارها، مدیریت گرمایی، یادگیری تقویتی

تاریخ ارسال مقاله : ۱۳۹۸/۱/۱۱

تاریخ پذیرش مشروط مقاله : ۱۳۹۸/۱۰/۱

تاریخ پذیرش مقاله : ۱۳۹۹/۳/۳۱

نام نویسنده‌ی مسئول: وصال حکمی

نشانی نویسنده‌ی مسئول: تهران، میدان رسالت، خیابان هنگام، خیابان دانشگاه، دانشگاه علم و صنعت ایران دانشکده مهندسی کامپیوتر

۱- مقدمه

کامپیوترهای نسل قدیم از پردازنده‌های تک‌هسته‌ای استفاده می‌کردند. این پردازنده‌ها به علت عدم توانایی در اجرای موازی برنامه‌ها، نسبت به پردازنده‌های نسل جدید کارایی پایین‌تری داشتند. به همین دلیل، جهت افزایش کارایی و بهره‌وری پردازنده‌ها، از تکنیک‌های موازی‌سازی در سطوح مختلف استفاده می‌شد. اما موازی‌سازی نیز با مشکلات مواجه بود. تولید حرارت زیاد و طراحی دشوار از جمله این مشکلات بودند. همین امر سبب ظهور عصر جدیدی در معماری کامپیوتر شد. عصری که در آن پردازنده‌های چندهسته‌ای نقش کلیدی دارند.

یک پردازنده‌ی چندهسته‌ای متشکل از چند هسته‌ی پردازشی و منابع مشترک است که مدیریت واحدها و همکاری‌های بین آنها، همگی توسط یک سیستم‌عامل انجام می‌شود [۱]. در این سیستم‌ها از سخت‌افزارهای موازی جهت افزایش کارایی سیستم استفاده شده است. پردازنده‌های چندهسته‌ای شامل تعداد بسیار زیادی ترانزیستور هستند که افزایش توان مصرفی دمای تراشه را در پی دارد. در واقع، پردازنده‌های چندهسته‌ای با مشکلاتی نظیر دمای بالا، هزینه‌ی برق زیاد، نیاز به خنک‌سازی و مصرف زیاد باتری در دستگاه‌های سیار، روبه‌رو هستند. همچنین، افزایش دما، سبب کاهش طول عمر مفید قطعات سیستم شده و تأثیر مخربی بر مواد سازنده‌ی آن می‌گذارد. بعلاوه، به دلیل وابستگی حرکت حامل‌ها به دما، با افزایش دما کارایی سیستم نیز کاهش می‌یابد. لذا جلوگیری از افزایش دمای پردازنده به لحاظ هزینه و انرژی مصرفی حائز اهمیت است [۲].

به‌طور کلی، سه رویکرد برای کاهش دما در پردازنده‌های چندهسته‌ای وجود دارد: «مدیریت پویای توان»^۱ (DPM)، «مقیاس - دهی پویای ولتاژ و فرکانس»^۲ (DVFS) و «انتساب کارها». در ادامه، مختصراً به مرور پیشینه در این سه حوزه (با تأکید ویژه بر رویکرد سوم) پرداخته شده است.

۱-۱- مروری بر کارهای پیشین

رویکرد DVFS توان پویای مصرفی پردازنده را به‌وسیله‌ی کاهش فرکانس کلاک و یا ولتاژ منبع تغذیه کاهش می‌دهد. این رویکرد به‌طور کلی به سه صورت محلی [۳]، سراسری [۴] و خوشه‌بندی [۵] استفاده می‌شود. در روش محلی، DVFS برای هر هسته به‌صورت مستقل صورت می‌گیرد ولی در DVFS سراسری امکان تنظیم ولتاژ و فرکانس تمامی هسته‌ها به‌طور یکنواخت و هم‌زمان فراهم است. در روش خوشه‌بندی نیز، هسته‌ها به چندین گروه تقسیم شده و مقادیر ولتاژ و فرکانس به صورت یکدست و هم‌زمان به هسته‌های موجود در گروه اعمال می‌شوند. DVFS محلی به دلیل انتساب ولتاژ و فرکانس مجزا به هر هسته انعطاف‌پذیرتر است. اما DVFS سراسری با کاهش ولتاژ و فرکانس تمامی هسته‌ها سبب افت کارایی سیستم می‌شود.

ساده‌ترین رویکرد مبتنی بر DPM، روش ایست و حرکت [۶] برای کنترل دما است که قابل استفاده در هر دو مقیاس محلی و سراسری می‌باشد. در حالت سراسری، اگر دمای یکی از هسته‌ها به مقدار آستانه برسد، کل تراشه خاموش می‌شود. در روش محلی نیز فقط هسته‌ای که دمایش بیشتر از مقدار آستانه باشد متوقف می‌شود و تا زمانی که خنک نشده باشد، خاموش می‌ماند. از ایرادات روش ایست و حرکت، می‌توان به مشکل کاهش کارایی سیستم، اشاره کرد.

به‌طور کلی، راهکارهای مبتنی بر رویکردهای DPM و DVFS علاوه بر کاهش کارایی سیستم، از وقوع بحران‌های حرارتی پیش‌گیری نمی‌کنند. در واقع، این رویکردها تنها زمانی فعال می‌شوند که دمای سیستم به حد آستانه‌ی از پیش تعریف شده برسد.

در مقابل، رویکرد انتساب، بر پایه نگاهت هدفمند کارها به هسته‌های پردازنده استوار است. راهکارهای زیرمجموعه این رویکرد، به زبردسته‌های برون‌خط (یا ایستا) [۷] و برخط (پویا) تقسیم می‌شوند. در روش مدیریت حرارت برون‌خط، تصمیمات مربوط به مدیریت حرارتی قبل از شروع اجرا اتخاذ می‌شوند. این در حالی است که در روش‌های پویا تصمیم‌های انتساب حین اجرا گرفته می‌شوند. خود راهکارهای برخط نیز برحسب بر خورداری از قابلیت پیش‌بینی به دو زبردسته تقسیم می‌شوند: با قابلیت پیش‌بینی [۸] و بدون قابلیت پیش‌بینی [۲]. الگوریتم‌های بدون قابلیت پیش‌بینی هیچ مدلی برای پیش‌بینی دما در آینده ندارند و عملکرد آن‌ها تنها بستگی به مقادیر دمای اندازه‌گیری شده توسط حسگرهای حرارتی سطح تراشه و تخمین وضعیت فعلی آن دارد. اما الگوریتم‌های دارای قابلیت پیش‌بینی برای تصمیم‌گیری از مدل‌های پیچیده‌ای جهت پیش‌بینی دمای آینده‌ی هسته استفاده می‌نمایند. ایراد الگوریتم‌های این دسته، پیچیدگی قابل ملاحظه در کد و زمان محاسبه می‌باشد.

علی‌رغم اینکه مسأله انتساب برخط در کارهای پیشین مورد توجه بوده، اما یک راهکار واقع‌گرایانه برای انتساب کارها به هسته‌ها باید عدم قطعیت‌هایی را که در یک سیستم پردازشی چندهسته‌ای به‌طور طبیعی وجود دارند، در منطق انتساب خود لحاظ نماید؛ مهم‌ترین این عدم قطعیت‌ها عبارتند از: (۱) به‌طور کلی کارها به صورت تصادفی وارد سیستم می‌شوند، ماهیت‌های متفاوتی دارند و بنابراین، هر یک از آن‌ها زمان اجرای متفاوتی داشته و تأثیر خاص خود را بر پروفایل دمایی تراشه دارند. (۲) کارهای در حال اجرا می‌توانند به صورتی اتفاقی با هم دوبره‌دو جفت شده و حجمی غیرقابل پیش‌بینی از داده را تبادل می‌نمایند^۳ (IPC). در نتیجه‌ی این ارتباط، تعدادی از مسیرهای-های سطح تراشه نیز برای انتقال داده‌ها درگیر می‌شوند. تعداد و مکان مسیرهای درگیر شده در ارتباط کارها، به محل هسته‌هایی بستگی دارد که کارهای جفت شده، در حال اجرا بر آنها می‌باشند. (۳) واحدهای مجاور نیز تأثیر متقابل دمایی دارند. با توجه به غیر قابل پیش‌بینی بودن ماهیت کارها و جفت‌شدن آنها، تأثیر متقابل دمایی واحدهای مجاور نیز قابل پیش‌بینی نمی‌باشد.

در واقع، وضعیت آینده‌ی سیستم چندهسته‌ای تنها با انتساب کارها به هسته‌ها تعیین نشده و به عوامل تصادفی نیز وابسته است. عمده کارهای پیشین (مانند [۹، ۱۰])، از رویکردهای نزدیک‌بینانه^۴ برای کاهش دمای سیستم با انتساب کارها استفاده نموده‌اند. در حالی که استفاده از رویکردهای بهینه‌سازی نزدیک‌بینانه برای این مسئله، به دلیل عدم در نظرگیری عدم قطعیت‌ها، در درازمدت منجر به عملکرد زیر-بهینه می‌شوند. در واقع، روش اصولی این است که به جای بهینه‌سازی لحظه‌ای، خط سیر تصادفی سیستم بهینه شود. به این معنا، هدف سیستم می‌تواند به صورت کاهش دمای بیشینه در درازمدت و یا افزایش متوسط کارایی سیستم در درازمدت تعریف شود.

معدودی از راهکارهای انتساب برخط با قابلیت پیش‌بینی که از الگوریتم‌های یادگیری تقویتی استفاده کرده‌اند [۱۱-۱۳]، قابلیت تطبیق با وضعیت سیستم و تصمیم‌گیری در شرایط تصادفی را دارند؛ اما کلیه این راهکارها نیز حداقل با دو مشکل مواجه هستند: (۱) در مدل‌سازی مسئله، تنها بخشی از عدم قطعیت سیستم لحاظ شده (مثلاً: یا تصادفی بودن ورود کارها یا تصادفی بودن IPC لحاظ نشده است). (۲) با توجه به ورود تصادفی کارها، نیاز به یک راهکار زمان-پیوسته برای انتساب کارها به هسته‌ها می‌باشد تا بهره‌وری سیستم کاهش نیابد. در حالی که راهکارهای مذکور، از رویکردهای انتساب زمان-گسسته استفاده نموده‌اند که کاهش بهره‌وری را در پی دارد.

۱-۲- نوآوری‌ها

در این مقاله، یک راهکار سیستماتیک برای مدیریت پویای گرمایش در یک سیستم پردازشی چندهسته‌ای ارائه شده است که به صورت ترکیبی از هر دو رویکرد انتساب برخط و DVFS بهره می‌گیرد؛ جنبه-های نوآورانه راهکار پیشنهادی عبارتند از:

با توجه به نقش محوری عدم قطعیت‌های محیط عملیاتی در تعیین پروفایل گرمایی تراشه، از اساس، سیستم پردازنده‌ی چندهسته-ای به عنوان یک سیستم تصادفی پویا مدل شده است که در آن، وضعیت‌های سیستم شامل دمای لحظه‌ای هر یک از هسته‌ها، تعداد کارهای داخل سیستم و وضعیت (مشغول یا بی‌کار) بودن هر یک از هسته‌ها می‌باشد. مدل‌سازی ریاضی مسئله با استفاده از چارچوب فرآیند تصمیم مارکوف^۵ (MDP) [۱۴] انجام می‌شود که در آن اعمال کنترلی نیز شامل انتساب یک کار به یکی از هسته‌های پردازنده و تعیین ولتاژ و فرکانس کاری آن است. وجه تمایز روش پیشنهادی، لحاظ کردن همه جنبه‌های تصادفی سیستم (شامل: فرآیند ورود کارها، وضعیت صف و IPC) در مدل‌سازی مسئله است.

بر خلاف راهکارهای پیشین، مدل‌سازی پیشنهادی در قالب فرمالیسم زمان-پیوسته MDP یعنی SMDP^۶ است که دستیابی به بهره‌وری بالا در رسیدگی به صف کارهای ورودی به سیستم را تضمین می‌نماید.

برای محاسبه سیاست بهینه انتساب برخط کارها و DVFS، یک روال مبتنی بر یادگیری تقویتی ویژه SMDP پیشنهاد شده است. یادگیری تقویتی در سال‌های اخیر در بسیاری از حوزه‌های کاربردی شبکه و سخت‌افزار برای بهبود کارایی سیستم‌ها بکار گرفته شده است [۱۵، ۱۶]. این روال امکان بهینه‌سازی متوسط کارایی سیستم را در درازمدت فراهم می‌کند ضمن اینکه محاسبه سیاست بهینه به صورت بی‌نیاز از دانش آماری فرآیندهای تصادفی سیستم و تنها از طریق تعامل مکرر و واقعی با محیط عملیاتی و کسب تجربه و بازخورد انجام می‌شود. همچنین، برای تأمین مقیاس‌پذیری روال پیشنهادی در سیستم‌های با ابعاد وضعیتی بزرگتر، یک معماری تقریب تابع مبتنی بر ویژگی‌های مسئله انتساب پیشنهاد شده است.

عملکرد روش پیشنهادی به لحاظ دمایی و زمان سرویس با استفاده از ابزارهای تخصصی شبیه‌سازی سیستم‌های پردازشی چندهسته‌ای پیاده‌سازی شده و با روش‌های موجود مشابه برای انتساب برخط کارها مقایسه شده است.

۱-۳- ساختار مقاله

در بخش دوم مدل سیستم و فرضیات پایه ارائه می‌شود. در بخش سوم، راهکار پیشنهادی برای مسئله انتساب برخط کارها معرفی شده و نحوه‌ی عملکرد آن تشریح شده است. در بخش چهارم نیز به بیان نحوه‌ی پیاده‌سازی راهکار پیشنهادی و تحلیل نتایج شبیه‌سازی پرداخته شده و در انتها، جمع‌بندی و نتیجه‌گیری ارائه شده است.

۲- مدل سیستم

در این بخش، به تشریح بخش‌های مختلف مدل سیستم، شامل جزئیات معماری و عدم قطعیت‌های محیط عملیاتی، پرداخته شده است.

۲-۱- معماری سیستم

یک پردازنده‌ی چندهسته‌ای، شبکه‌ای بر تراشه است که در آن، هسته-ها از طریق مسیریاب‌ها و اتصالات شبکه با یکدیگر در ارتباط می‌باشند. متداول‌ترین توپولوژی برای پردازنده‌های چندهسته‌ای، توپولوژی Mesh است که در آن به ازای هر هسته‌ی پردازشی یک مسیریاب وجود دارد که به هسته متصل است [۱۷]. پردازنده‌ی مفروض، شامل M هسته‌ی $M = \{m_1, m_2, m_3, \dots, m_m, \dots, m_M\}$ و مسیریاب شبکه $\mathcal{R} = \{r_1, r_2, r_3, \dots, r_m, \dots, r_M\}$ می‌باشد به طوری که r_m مسیریاب متصل به هسته‌ی m_m می‌باشد. همچنین، فرض شده است که تمامی هسته‌های پردازنده مجهز به تنظیم‌کننده‌های مجزای ولتاژ و فرکانس می‌باشند. واحد زمان‌بند سیستم عامل نیز وظیفه‌ی انتساب کارها را بر عهده دارد. این واحد خود متشکل از سه بخش است [۱۸]:

- **نظاره‌گر نخ‌ها:** وظیفه‌ی این بخش، خواندن مقادیر حسگرهای حرارتی سطح تراشه و ارسال آن به بخش‌های مدیریت حرارتی و

توزیع کننده می باشد. M حسگر حرارتی در سطح پردازنده قرار داده شده است که c_m حسگر قرار داده شده کنار مسیریاب r_m می باشد. بردار $c(t) = \{c_1, c_2, c_3, \dots, c_m, \dots, c_M\}$ نیز پروفایل دمایی قرائت شده از سطح تراشه در لحظه t است.

- **توزیع کننده:** این بخش، مجموعه‌ی $c(t)$ را از بخش نظاره‌گر نخبها دریافت کرده و با توجه به آن، یکی از کارهای آماده‌ی اجرا را به یکی از هسته‌های بی کار انتساب می دهد.
- **مدیریت حرارتی:** این بخش، مقادیر $c(t)$ را از بخش نظاره‌گر نخبها دریافت کرده و وظیفه‌اش انتخاب سطح ولتاژ و فرکانس برای هسته‌ای است که کار جدید به آن انتساب داده می شود. $\mathcal{L} = \{l_1, l_2, l_3, \dots, l_i, \dots, l_L\}$ مجموعه‌ی سطوح ولتاژ و فرکانس قابل پشتیبانی برای هسته‌ها می باشد.

فرض ۱: واحد زمان بند، بر یک هسته‌ی اختصاصی قرار دارد.

نکته‌ی قابل توجه این است که عملکرد واحد زمان بند، مبتنی بر رویداد است. در صورتی که کار جدیدی وارد سیستم شود و یا اجرای یکی از کارهای در حال اجرا به پایان رسد، واحد زمان بند فراخوانی می شود تا کار دیگری را به یکی از هسته‌های بی کار انتساب دهد.

تعریف ۱: حاشیه‌ی دمایی برابر است با میزان اختلاف دمایی بیشینه‌ی سیستم و دمایی آستانه‌ی مجاز T_{th} (رابطه‌ی (۱)).

$$T_m = \max(T_{th} - T_p, 0) \quad (1)$$

که در آن T_p بیانگر دمایی بیشینه سیستم است که در لحظه t با رابطه (۲) قابل محاسبه است:

$$T_p(t) = \max_m c_m(t) \quad (2)$$

۲-۲-۲- عدم قطعیت در سیستم

به واسطه غیرقطعیت بودن محیط عملیاتی، هر تصمیم اتخاذ شده در لحظه جاری، سهم به‌سزایی در تعیین وضعیت آتی سیستم دارد. بنابراین، توجه به معیارهای لحظه‌ای در تصمیم‌گیری‌ها کافی نبوده و در نظرگیری تبعات تصمیمات فعلی در روند آتی سیستم ضروری می باشد. از این رو، شناسایی عناصر غیرقطعیت و توصیف آنها برای بهینه سازی الزامی است. در ادامه، به بررسی عدم قطعیت‌های سیستم، شامل ورود تصادفی کارها و جفت شدن آنها با یکدیگر پرداخته شده است.

۲-۲-۱- ورود تصادفی کارها به سیستم

در این سیستم، I نوع کار مختلف وارد سیستم می شوند که هر کدام به تعداد N زیر-نوع دارند. مجموعه‌ی انواع کارها و زیر-نوع‌ها به ترتیب، $\mathcal{I} = \{1, 2, 3, \dots, i, \dots, I\}$ و $\mathcal{N} = \{1, 2, 3, \dots, n, \dots, N\}$ می باشند و برای نمایش کاری از نوع i و زیر-نوع n از نماد $\tau_{i,n}$ استفاده می شود. هر یک از کارها نیز با نرخ متفاوتی وارد سیستم می شوند. نماد $\lambda_{i,n}$ نشان دهنده‌ی نرخ متوسط ورود کار $\tau_{i,n}$ بوده و λ نرخ متوسط ورود تجمعی تمامی کارها را نشان می دهد.

فرض ۲: $\lambda_{i,n}$ به ازای $\forall i, n$ پارامتر پواسون بوده و نامعلوم است.

توضیح ۱: با توجه به اینکه تمامی نرخ‌ها پواسن می باشند، نرخ مجموع، یعنی λ نیز پواسن خواهد بود.

ماهیت کارها و نوع آنها عامل مهمی در تعیین مدت زمان اجرا و پروفایل دمایی تراشه می باشند. از سوی دیگر، فرکانس کاری هسته‌ای که کار بر آن اجرا می شود نیز نقش به‌سزایی در تعیین مدت زمان اجرا و دمایی پردازنده ایفا می کند. به این ترتیب که افزایش فرکانس کاری هسته‌ها سبب افزایش سرعت اجرا و در نتیجه بالا رفتن دما می شود. $k_{i,n,l}$ مدت زمان اجرای کار $\tau_{i,n}$ را در سطح کاری l (مقدار ولتاژ و فرکانس انتخاب شده از مجموعه‌ی سطوح ولتاژ و فرکانس قابل قبول (\mathcal{L})) و بدون در نظر گیری زمان صرف شده برای ارتباط کارها نشان می دهد.

فرض ۳: قبل از اجرای کارها، اطلاعاتی از نوع آنها، مدت زمان اجرای آنها بر تراشه و پروفایل دمایی حاصل از اجرا در دسترس نمی باشد.

فرض ۴: در هر سطح فرکانس و ولتاژ، بدون در نظر گرفتن زمان تبادل داده‌ی کارها، هر یک از کارها، مانند $\tau_{i,n}$ زمان اجرای قطعی و ثابتی در هر یک از هسته‌های پردازنده دارند.

فرض ۵: هر یک از کارها تنها بر یکی از هسته‌ها اجرا می شوند و امکان اجرای بخش‌هایی از یک کار بر هسته‌های مختلف وجود ندارد.

۲-۲-۲- عدم قطعیت در ارتباطات کارها

در حالت کلی، هنگامی که یک کار جهت اجرا به یکی از هسته‌های پردازنده انتساب داده می شود، می تواند به صورت تصادفی با یکی از کارهای موجود جفت شده و تبادل داده داشته باشد. تعداد و محل مسیریاب‌های درگیر در هر ارتباط، به مکان هسته‌هایی بستگی دارد که کارهای جفت شده، در حال اجرا بر آنها می باشند. از سوی دیگر، میزان توان مصرفی مسیریاب‌ها و گرمای تولیدشده توسط آنها، در مقایسه با سایر اجزای پردازنده، قابل توجه بوده [۱۱] و بر دمایی واحدهای مجاور، تأثیرگذار است. بنابراین، لحاظ کردن عدم قطعیت IPC ، در انتساب کارها با هدف کاهش دمایی پردازنده ضروری است.

فرض ۶: هر یک از کارهای در حال اجرا، در آن واحد تنها می توانند با یکی از کارهای دیگر جفت شوند.

فرض ۷: زمان تبادل داده و حجم داده‌ی منتقل شده به صورت تصادفی مشخص می شوند. فرض می شود زمان‌های لازم برای تبادل داده‌ها دارای توزیع نمایی ولی با پارامتر نامعلوم ξ می باشند.

۲-۲-۳- صف کارهای آماده‌ی اجرا

با توجه به ورود پواسن و سرویس‌گیری نمایی کارها، سیستم پردازنده‌ی چند هسته‌ای مفروض با یک سیستم صف $M/M/M$ معادل است که در آن یک صف انتظار با نظام سرویس $FCFS^V$ و طول نامحدود و M

سرویس‌دهنده یکسان حضور دارند [۱۹]. وضعیت صف کارهای آماده در لحظه t با نماد $q(t) \in \{1, 2, 3, \dots\}$ نشان داده می‌شود.

۳- فرمول بندی مسأله انتساب برخط

در این بخش، از چارچوب فرمول بندی MDP برای بیان رسمی مسأله انتساب برخط کارها استفاده شده است. چارچوب ریاضی MDP روال تصمیم‌سازی یک عامل در یک محیط غیرقطعی چندوضعیتی را مدل می‌کند که از طریق انتخاب اعمال کنترلی می‌تواند فرآیند تصادفی تغییرات وضعیت محیط را کنترل کند. مسأله انتساب کارها به هسته‌ها را می‌توان با استفاده از MDP زمان-گسسته مدل‌سازی کرد. در این صورت، زمان‌بند سیستم عامل در بازه‌های زمانی منظمی اقدام به انتساب یکی از کارهای آماده اجرا می‌نماید. با این حال، تحت این نوع فرمول بندی، ممکن است گاهی تعدادی کار آماده در صف، منتظر اجرا بوده و هسته یا هسته‌های بی‌کار در سیستم وجود داشته باشند اما واحد زمان‌بند در ابتدای هر بازه‌ی زمانی، تنها یکی از کارهای منتظر در صف را جهت اجرا به یکی از هسته‌های بی‌کار انتساب دهد. این امر، سبب بالا رفتن زمان اجرای کارها با وجود هسته‌های بی‌کار در سیستم شده و بهره‌وری سیستم را کاهش می‌دهد. در مقابل، در این مقاله ایده استفاده از MDP زمان-پیوسته یا همان SMDP برای مدل کردن سیستم مطرح شده است. نحوه‌ی کار SMDP به این صورت است که هنگام وقوع یک "رویداد" در سیستم، در صورت وجود هسته‌ی بی‌کار، واحد زمان‌بند یکی از کارهای منتظر در صف را به یکی از هسته‌های بی‌کار انتساب می‌دهد. یک رویداد در سیستم زمانی اتفاق می‌افتد که کار جدیدی وارد سیستم شود و یا اجرای یکی از کارهای موجود خاتمه یافته و هسته‌ی مشغول، بی‌کار شود.

۳-۱- فرمالیسم SMDP

در این بخش، به معرفی و تشریح اجزای SMDP شامل وضعیت‌های سیستم، اعمال (تصمیم‌ها)، پویایی تصادفی، پاداش لحظه‌ای و نیز، سیاست تصمیم‌سازی و هدف بهینه‌سازی پرداخته شده است. در ادامه راهکاری مبتنی بر یادگیری تقویتی برای محاسبه سیاست بهینه تصمیم‌سازی ارائه شده است.

۳-۱-۱- وضعیت‌های سیستم

هر وضعیت سیستم $s(t)$ در لحظه‌ی t برداری به شکل زیر می‌باشد:

$$s(t) = \{c(t), b(t), \kappa(t)\} \quad (3)$$

که در آن $b(t)$ به صورت رابطه‌ی (۴) تعریف شده و نشان‌دهنده‌ی

وضعیت هسته‌ها می‌باشد به این ترتیب که مقدار صفر نشان‌دهنده‌ی بی‌کار بودن هسته و مقدار یک، بیان‌گر مشغول بودن هسته است:

$$b(t) \in \mathcal{B} = \{0, 1\}^M \quad (4)$$

تعداد کارهای حاضر در سیستم $(\kappa(t))$ در لحظه‌ی t نیز به عنوان مؤلفه‌ی دیگری از بردار وضعیت در نظر گرفته شده است. به طوری که طبق رابطه‌ی (۵) مقادیر 0 تا بی‌نهایت را در برمی‌گیرد.

$$\kappa(t) \in K = \{0, 1, 2, \dots\} \quad (5)$$

۳-۱-۲- اعمال کنترلی

زمان‌بند سیستم عامل در هر وضعیت سیستم، از میان اعمال قابل انجام، یک عمل را انتخاب کرده و انجام می‌دهد. در این مسئله، هر عمل به شکل رابطه‌ی (۶) تعریف می‌شود که در آن m_m هسته‌ای است که برای اجرای کار سر صف انتخاب شده است. l_i نیز سطح ولتاژ و فرکانس انتخاب شده جهت اجرای کار مربوط به هسته‌ی m_m می‌باشد.

$$a(t) = (m_m, l_i) \quad (6)$$

همان‌طور که پیش‌تر گفته شد، عملکرد واحد زمان‌بند، مبتنی بر رویداد است. به این معنا که تنها در صورت وقوع رویدادی در سیستم، واحد زمان‌بند اقدام به انتساب کار موجود در سر صف به یکی از هسته‌های بی‌کار و تنظیم ولتاژ و فرکانس کاری برای آن می‌نماید.

۳-۱-۳- پویایی تصادفی

در این بخش به بررسی تغییرات مؤلفه‌های بردار وضعیت در طول زمان پرداخته شده است. تعداد کارهای حاضر در سیستم در طول زمان به شکل رابطه‌ی زیر تغییر می‌کند (برای $t' > t$):

$$\kappa(t') = \max\{\kappa(t) - O(t), 0\} + \text{poissrand}(\lambda(t' - t)) \quad (7)$$

که $K(t)$ و $K(t')$ به ترتیب تعداد کارهای موجود در سیستم در لحظه‌ی t و t' می‌باشند. $O(t)$ نشان‌دهنده‌ی تعداد کارهایی است که اجرای آن‌ها در فاصله‌ی زمانی t تا t' به پایان می‌رسد. بنابراین، این دسته از کارها از سیستم خارج شده و به همان تعداد، هسته بی‌کار می‌شوند. $(\lambda(t' - t))$ Poissrand تعداد کارهایی است که در بازه‌ی زمانی t تا t' طبق توزیع پواسون وارد سیستم شده‌اند.

مقادیر دمای هسته‌ها نیز مؤلفه‌ی تصادفی دیگر سیستم می‌باشد. طبق خاصیت مارکوفی، در یک MDP، وضعیت بعدی سیستم بستگی به وضعیت جاری و عمل انتخاب شده در آن دارد. علاوه بر این، عوامل تصادفی نیز در تعیین وضعیت بعدی سیستم نقش به‌سزایی ایفا می‌کنند. به این ترتیب، در صورتی که زمان فعلی برابر با t باشد، وضعیت بعدی سیستم یا همان پروفایل دمایی تراشه در لحظه‌ی t از رابطه‌ی $c(t') = f(c(t), a(t), \varrho)$ تبعیت خواهد کرد که در آن، ϱ پارامتر مربوط به عوامل تصادفی (جفت شدن تصادفی کارها، مکان و تعداد مسیربای‌های درگیر در ارتباط آن‌ها) می‌باشد.

۳-۱-۴- پاداش لحظه‌ای

همان‌طور که گفته شد، زمان‌بند سیستم عامل، در هر وضعیتی از سیستم، یکی از اعمال قابل انجام را انتخاب نموده و انجام می‌دهد. پس از انجام عمل، سیستم به وضعیت جدیدی گذار کرده و زمان‌بند به

$$R(s, \bar{a}) = T_s(a)r(s, a) \quad (12)$$

در رابطه‌ی بالا، $R(s, a)$ امید ریاضی پاداشی است که در وضعیت s با انجام عمل a دریافت شده است. $r(s, a)$ نیز پاداش استفاده از عمل a در وضعیت s می‌باشد. همچنین، $\bar{T}_s(a)$ امید ریاضی زمان انتقال از وضعیت s با انجام عمل a بوده و با رابطه‌ی زیر محاسبه می‌شود:

$$\bar{T}_s(a) = \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} \int_{C_0} T dF_{s, s'}(T, a) dc' \quad (13)$$

میانگین پاداش‌های دریافتی لحظه‌ای در بی‌نهایت، نشان‌دهنده‌ی کارایی دنباله تصمیمات اتخاذشده در راستای رسیدن به هدف سیستمی بلندمدت (یعنی: حداکثرسازی میانگین امید ریاضی پاداش-های لحظه‌ای دریافت شده در بی‌نهایت) است [۱۴]. تحقق این هدف زمانی ممکن است که عامل سیاست بهینه را مبنای تصمیم‌گیری‌های خود قرار دهد. بنابراین، هدف ما به‌دست آوردن یک سیاست تطبیقی بهینه (با نماد π^*) برای انتساب برخط کارها به هسته‌ها و تنظیم پویای ولتاژ و فرکانس کاری آن‌ها می‌باشد به‌طوری‌که استفاده از این سیاست، منجر به دریافت حداکثر پاداش میانگین در درازمدت شود:

$$\pi^* = \arg \max_{\pi} \bar{R}_{\pi} \triangleq \max_{\pi} \lim_{k \rightarrow \infty} \frac{1}{k} \times E_{\pi} \left\{ \sum_{k=0}^{k-1} \int_{t_k}^{t_{k+1}} r(s_k, \pi(s_k)) dt \mid s_0 = s \right\} \quad (14)$$

۳-۱-۷- الگوریتم یادگیری پیشنهادی

چنانچه پیش‌تر خاطر نشان شد، فرآیند ورود کارها به سیستم، با فرآیند پواسون مدل شده است ($\text{poissrand}(\lambda)$). همچنین، تابع $f(c(t))$ ، Q و $a(t)$ نیز پویایی تصادفی دما را مدل می‌کند. با توجه به اینکه در حالت کلی، λ برای هر سیستمی از پیش معلوم نیست و فرم تحلیلی تابع f نیز به‌طور عمومی نامشخص بوده و احتمال جفت شدن کارها با یکدیگر نیز از پیش مشخص نمی‌باشد، امکان محاسبه‌ی سیاست بهینه به‌صورت برون-خط و مبتنی بر یک مدل احتمالاتی خاص وجود ندارد. برای جبران کمبود این اطلاعات، عامل به قابلیت یادگیری و کسب تجربه مجهز شده است. در ادامه، به تشریح میانی نظری که الگوریتم پیشنهادی بر آن استوار است، پرداخته شده است.

مطابق رویه استاندارد یادگیری تقویتی (مثلاً: [۱۴])، برای هر یک از اعمال در هر یک از وضعیت‌ها با استفاده از معادله‌ی بهینگی Bellman، مقداری به عنوان تابع Q یا تابع "ارزش وضعیت-عمل" انتساب داده می‌شود. این مقدار از رابطه‌ی زیر قابل محاسبه است:

$$Q(s, a) = R(s, a) - \rho \bar{T}_s(a) + \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{K}} \left[\int_C f_{cc'}(a) \max_{a' \in A(s')} Q(s', a') dc' \right], \quad \forall (s, a) \quad (15)$$

که در آن، ρ متوسط پاداش دریافتی بهینه در یک گام، $P(b' | b, a)$ و $P(k' | k, a)$ مقادیر احتمالات گذار و $f_{cc'}(a)$ مقدار

عنوان «عامل» تصمیم‌ساز، مقداری پاداش دریافت می‌کند. در مسئله‌ی انتساب کارها بر هسته‌ها، تصمیم‌گیری‌های عامل باید به‌گونه‌ای باشد که متوسط "دمای بیشینه‌ی سیستم" در طول زمان کاهش یابد. به‌طور لحظه‌ای، دمای بیشینه‌ی سیستم، دمای بیشینه‌ی اجزای پردازنده در یک لحظه‌ی مشخص می‌باشد. سپس، دمای واحدی که در مقایسه با سایر واحدها بالاتر است، به عنوان دمای بیشینه‌ی سیستم در نظر گرفته می‌شود. بنابراین، تابع پاداش لحظه‌ای (در واحد زمان) در لحظه‌ی t به شکل رابطه‌ی زیر تعریف می‌شود:

$$r(t) = T_m = T_{th} - T_p \quad (8)$$

همان‌طور که پیش‌تر گفته شد، در این مسئله، عامل می‌خواهد متوسط دمای بیشینه‌ی سیستم را کاهش دهد. کاهش متوسط دمای بیشینه‌ی سیستم به معنای حداکثرسازی متوسط حاشیه‌ی دمایی است. بنابراین، پاداشی که عامل بعد از انتساب یک کار به یک هسته دریافت می‌کند، برابر با میزان حاشیه‌ی دمایی می‌باشد.

۳-۱-۵- سیاست تصمیم‌ساز وقتی

در SMDP، یک سیاست، نحوه‌ی رفتار عامل یادگیرنده (تصمیم-گیرنده) را مشخص می‌نماید. به عبارت بهتر، زمانی که سیستم در هر کدام از وضعیت‌ها قرار داشته باشد، بر اساس سیاستی که دنبال می‌کند، یکی از اعمال قابل انجام را انتخاب می‌نماید یک سیاست کنترلی به صورت نگاشت احتمالی ثابت $\pi: \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{A})$ تعریف می‌شود که در آن \mathcal{S} مجموعه‌ی وضعیت‌ها، \mathcal{A} مجموعه‌ی اعمال کنترلی و $\Delta(\mathcal{A})$ نشان‌دهنده‌ی تمامی توزیع‌های احتمالی در فضای اعمال بوده و از رابطه‌ی زیر بیان می‌شود:

$$\Delta(\mathcal{A}) = \left\{ P : P \in (R^+)^{|\mathcal{A}|}, \sum_{a \in \mathcal{A}} P(a) = 1 \right\} \quad (9)$$

۳-۱-۶- هدف سیستمی (پاداش میانگین درازمدت)

تابع پاداش درازمدت برای مسئله‌ی زمان-پیوسته به‌صورت [۱۴]:

$$\bar{R} = \lim_{T \rightarrow \infty} \frac{1}{T} E_{\pi} \left\{ \int_0^T r(s(t), a(t)) dt \right\} \quad (10)$$

می‌باشد که در آن، $r(s(t), a(t))$ مقدار پاداش دریافتی عامل در لحظه‌ی t است که طی آن سیستم در وضعیت $s(t)$ بوده و تصمیم $a(t) \sim \pi(s(t))$ را انجام داده باشد. نماد \sim به معنی نمونه‌گیری عمل $a(t)$ از تابع سیاست اتفاقی و مانای π (یعنی: توزیع احتمال روی مجموعه اعمال امکان‌پذیر در وضعیت $s(t)$) است. رابطه‌ی (۱۰) معادل رابطه‌ی زیر می‌باشد [۱۴]:

$$\bar{R}_{\pi} = \lim_{N \rightarrow \infty} \frac{1}{N} E_{\pi} \left\{ \int_0^{t_N} r(s(t), a(t)) dt \right\} \quad (11)$$

که در آن، t_N زمان کامل شدن N امین گذار می‌باشد. متوسط پاداش دریافتی لحظه‌ای را نیز می‌توان از طریق رابطه‌ی (۱۲) محاسبه نمود:

وضعیت اولیه را به فضای ویژگی تصویر کرده و سپس ارزش وضعیت‌ها را در فضای ویژگی جدید توسط رابطه (۱۸) به صورت خطی تخمین می‌زند [۲۱]:

$$\hat{Q}(s, a) = \theta^T \times \phi(s, a) \quad (18)$$

در رابطه‌ی (۱۸)، $\phi(s, a)$ تابع ویژگی وضعیت s و عمل a و بردار وزن برای ویژگی‌ها است. در این مسئله، از تکنیک $RBFA$ برای تعریف توابع ویژگی‌ها استفاده شده است [۲۱] که به صورت زیر تعریف می‌شود:

$$\phi_i(s) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\|w_i - c\|^2 / 2\sigma^2} \quad (19)$$

در رابطه‌ی (۱۹)، σ یک عدد ثابت بوده و عرض ویژگی‌ها را مشخص می‌کند. w نیز یک بردار M -مؤلفه‌ای می‌باشد که مراکز هریک از ویژگی‌ها را تعیین می‌کند و هر یک از مؤلفه‌های آن یک مقدار از محدوده‌ی عادی دمای هسته‌ها را شامل می‌شود. w - c نیز میزان اختلاف دمای خوانده شده از حسگرها و مقادیر محدوده‌ی عادی دما را به دست می‌آورد.

پس از هر بار تصمیم‌گیری و انتخاب یکی از اعمال، بردار پارامتر θ به نحوی به‌روزرسانی می‌شود که مقدار خطای E که به صورت مربع اختلاف میان مشاهده جدید از تابع Q و مقدار تقریب‌زده شده آن یعنی \hat{Q} تعریف می‌شود، حداقل‌سازی شود [۲۱]:

$$E_k \triangleq [Q_{k+1}(s_k, a_k) - \hat{Q}(s_k, a_k)]^2 = [Q_{k+1}(s_k, a_k) - \theta^T \times \phi(s_k, a_k)]^2 \quad (20)$$

که در آن، $Q_{k+1}(s_k, a_k)$ برابر است با:

$$Q_{k+1}(s_k, a_k) = \bar{R}(s_k, a_k) - \hat{Q}(s^*, a^*)t(s_k, a_k, s_{k+1}) + \max_{b \in A(s_{k+1})} \hat{Q}(s_{k+1}, b) \quad (21)$$

برای حداقل‌سازی میزان اختلاف مقادیر یادشده، از تکنیک گرادینان نزولی (با اندازه گام β_k) برای به‌روزرسانی θ استفاده می‌شود:

$$\frac{\partial E}{\partial \theta_k} = -2\phi(s_k, a_k) [Q_{k+1}(s_k, a_k) - \theta^T \times \phi(s_k, a_k)] \quad (22)$$

$$\theta_{k+1} = \theta_k - \beta_k \frac{\partial E}{\partial \theta_k} \quad (23)$$

نهایتاً، با جایگذاری $\alpha_k = 2\beta_k$ داریم:

$$\theta_{k+1} = \theta_k + \alpha_k (Q_{k+1}(s_k, a_k) - \hat{Q}(s_k, a_k)) \phi(s_k, a_k) \quad (24)$$

۴- ارزیابی راهکار پیشنهادی

در این بخش، با استفاده از شبیه‌سازی، راهکار پیشنهادی مورد ارزیابی و مقایسه قرار گرفته است.

چگالی احتمال گذار می‌باشند. در صورتیکه بتوان تابع Q را محاسبه کرد، سیاست بهینه $\pi^* = \arg \max_a Q(s, a)$ به صورت قابل محاسبه خواهد بود [۲۰]. اِثاً، برای محاسبه‌ی $Q(s, a)$ بر اساس رابطه-ی (۱۵)، مدل احتمالاتی سیستم یعنی توابع احتمال گذار و چگالی احتمال مورد نیاز می‌باشند. همچنین، لازم است مقدار بهینه‌ی متوسط پاداش هر گام (p^*) محاسبه شود. در غیاب دانش آماری، از الگوریتم «یادگیری Q » می‌توان استفاده نمود که به جای اتکا به مقادیر احتمالات گذار از نمونه‌های محقق شده پاداش لحظه‌ای و نیز وضعیت مشاهده شده بعدی سیستم برای تخمین تابع $Q(s, a)$ استفاده می‌کند. همچنین، برای حل مشکل محاسبه‌ی p^* ، در [۱۴] ثابت شده است که به جای مقدار p^* می‌توان از مقدار ارزش یکی از اعمال در یکی از وضعیت‌ها استفاده نمود یعنی، فرضاً: $Q_k(s^*, a^*)$. در واقع، می‌توان از الگوریتم «یادگیری Q نسبی زمان-پیوسته» زیر استفاده کرد [۲۰]:

$$Q_{k+1}(s, a) = (1 - \alpha_k) Q_k(s, a) + \alpha_k [\bar{R}(s, a) - Q_k(s^*, a^*)t(s, a, s') + \max_{b \in A(s')} Q_k(s', b)] \quad (16)$$

که در آن، $t(s, a, s')$ زمان سپری شده طی انتقال از وضعیت s به s' با انجام عمل a ، $\bar{R}(s, a)$ نمونه‌ی پاداش انباشته شده در طی زمان قرار سیستم در وضعیت s قبل از گذار به وضعیت s' و α_k نرخ یادگیری بوده و از رابطه‌ی زیر محاسبه می‌شود که در آن، A و B مقادیر ثابت می‌باشند:

$$\alpha_k = \frac{A}{B + k} \quad (17)$$

به ترتیبی که گفته شد، برای هریک از اعمال در هریک از وضعیت‌ها، مقدار ارزش (Q) در یک جدول نگهداری و در هر گام به-روزرسانی می‌شود. نکته‌ی قابل توجه این است که مقادیر دمای خوانده شده از حسگرهای حرارتی پیوسته است. بنابراین، اندیس تابع Q از یک فضای پیوسته مقدار می‌گیرد و عملاً بی‌نهایت مقدار خواهد داشت. بنابراین، استفاده از جدول برای نگهداری ارزش اعمال برای هر وضعیت عملی نخواهد بود. برای رفع این مشکل، در بخش بعدی، از تکنیک تقریب تابع Q استفاده شده است.

۳-۱-۸- معماری تابع تقریب

با توجه به پیوسته بودن فضای وضعیت دمایی تراشه، فُرم استاندارد تابع Q که متشکل از جدولی با اندیس‌های وضعیت-عمل است، عملاً نامتناهی می‌شود. برای رفع این چالش، از تکنیک تقریب تابع استفاده شده است که از شباهت‌سنجی وضعیت‌ها برای تقریب ارزش آن‌ها استفاده می‌کند [۲۱]. بر این اساس، عامل یادگیرنده دیگر نیازی به تجربه‌اندوزی مستقیم در تمام وضعیت‌های محیط را برای تخمین ارزش آن‌ها ندارد که این موضوع می‌تواند سرعت همگرایی را تا حد زیادی افزایش دهد. در این تکنیک، "تابع تقریب" جایگزین جدول ذخیره‌سازی ارزش Q می‌شود. تابع تقریب ابتدا توسط «تابع ویژگی»^۸

۴-۱- ابزارها و گردش کار شبیه‌سازی

در شبیه‌سازی انجام شده، به تعداد هسته‌های پردازنده حسگرهای حرارتی در سطح تراشه و در محل مسیریاب‌ها قرار داده شده‌اند. بنابراین، پس از خواندن حسگرهای حرارتی، به تعداد هسته‌ها مقادیر دما خواهیم داشت که با افزایش تعداد هسته‌ها، پارامترهای دمایی نیز به همان نسبت بیشتر می‌شوند. برای حل این مشکل، از روش درون-یابی دوخطی^{۱۲} استفاده شده است. به این ترتیب، مستقل از تعداد هسته‌های پردازنده، تنها از نُه مقدار دما برای ساخت ماتریس ویژگی استفاده می‌شود. همچنین، برای سادگی، تمامی مقادیر دما، در بازه‌ی (۰ و ۱) مقیاس‌دهی می‌شوند. مقادیر متغیرهای شبیه‌سازی در جدول (۲) قرار داده شده است.

۴-۳- روش‌های مورد مقایسه

نتایج حاصل از شبیه‌سازی راهکار پیشنهادی، مشخصاً با دو روش پیشین [۱۰] و [۱۱] مقایسه شده است. راهکار ارائه‌شده در [۱۰] با عنوان TBO مشخص شده و عملکرد آن به این صورت است که با توجه به میزان فاصله‌ی هسته‌ها از مرکز پردازنده و دمای آن‌ها، به هر-

شبیه‌سازی راهکار پیشنهادی در دو بخش برخط و برون-خط صورت می‌گیرد. بخش برون‌خط، بخشی است که تمامی مراحل آن تنها یک‌بار در ابتدای شبیه‌سازی انجام می‌شود و هدف آن جمع‌آوری داده‌های مورد نیاز برای انجام بخش برخط می‌باشد. در بخش برون-خط، ابتدا یک پردازنده‌ی چند هسته‌ای همگون^{۱۰} با استفاده از ابزار Sniper [۲۲] شبیه‌سازی شده است. هر یک از برنامه‌های محک Splash2 به عنوان یک کار مستقل، بر یک هسته‌ی پردازنده (با سطوح ولتاژ و فرکانس مختلف) اجرا شده و زمان اجرای آن محاسبه شده است. طرح هسته نیز با استفاده از شبیه‌ساز HotFloorPlan (بخشی از شبیه‌ساز حرارتی HotSpot [۲۳]) برای یک هسته آماده شده است. با استفاده از ابزار McPAT [۲۴] و نتایج حاصل از شبیه‌ساز Sniper، مقدار توان پویا برای اجرای هر یک از کارها در هر سطح کاری محاسبه شده است. سپس، با استفاده از شبیه‌ساز DSENT [۲۵] مقادیر توان مسیریاب‌ها به ازای نرخ‌های تزریق مختلف^{۱۱} محاسبه شده‌اند.

بخش برخط نیز به‌ازای انتساب هر یک از کارها، یک بار تکرار می‌شود. به این صورت که ابتدا واحد زمان‌بند سیستم‌عامل کار واقع در سر صف را به یکی از هسته‌ها انتساب داده و سطح ولتاژ و فرکانس کاری آن را مشخص می‌نماید. کار مذکور به‌طور تصادفی با یکی از کارهای در حال اجرا که با کار دیگری جفت نشده است، جفت می‌شود. جفت شدن کارها نیز به این صورت است که هر یک از دو کار جفت شده، ضمن اجرا، با نرخ تزریق تصادفی به دیگری داده ارسال می‌کند. مدت‌زمان ارتباط کارهای جفت شده نیز از یک توزیع نمایی با پارامتر ξ به دست می‌آید. همچنین، زمان اجرای کارها نیز بسته به فرکانس کاری هسته، از پیش توسط شبیه‌ساز Sniper به دست آمده است. در مرحله‌ی بعدی، مقادیر توان ایستا، طرح هسته و دمای هسته‌ها به عنوان ورودی به شبیه‌ساز HotSpot [۲۳] داده شده و پروفایل دمایی تراشه به عنوان خروجی دریافت می‌شود. این ابزار، قابلیت در نظرگیری تأثیر دمایی واحدها بر یکدیگر را دارا بوده و پروفایل حرارتی پردازنده را با در نظرگیری مکان و دمای اجزا تولید می‌کند. سپس، بسته به مقادیر دمای هسته‌ها، مقدار حاشیه‌ی دمایی (پاداش) محاسبه شده و به زمان‌بند سیستم‌عامل گزارش می‌شود.

۴-۲- پارامترها و تنظیمات شبیه‌سازی

هر یک از هسته‌های پردازنده از نوع Intel Xeon Gainestown (Nehalem-EP) می‌باشند. مسیریاب‌ها و هسته‌ها نیز با استفاده از تکنولوژی 45nm و تنظیمات جدول (۱) ساخته شده‌اند. سطوح ولتاژ و فرکانس هسته‌ها [۱۱] نیز در جدول (۱) قرار دارند. نرخ تزریق مسیریاب‌های درگیر در ارتباط کارها، به صورت تصادفی از بازه‌ی (۰ و ۱) انتخاب می‌شود. الگوریتم مسیریابی استفاده شده برای تبادل داده‌ها نیز الگوریتم xy می‌باشد [۲۶].

جدول (۱): تنظیمات شبیه‌سازی

تنظیمات هسته‌ها		تنظیمات مسیریاب‌ها		سطوح کاری	
L1-I	۱۶ کیلوبایت	۵	تعداد پرت‌ها	ولتاژ (V)	فرکانس (GHz)
L1-D	۱۶ کیلوبایت	۲ GHz	فرکانس	۰.۹	۲.۷
L2	۲۵۶ کیلوبایت	۸	تعداد کانال‌های مجازی هر پرت	۱.۰	۳.۰
ITLB	۱۶ جزء	۱۴۴ b	اندازه‌ی هر flit	۱.۱	۳.۳
DTLB	۱۶ جزء	۲۴ flit	طول بافر	۱.۲	۳.۶

جدول (۲): مقادیر متغیرها

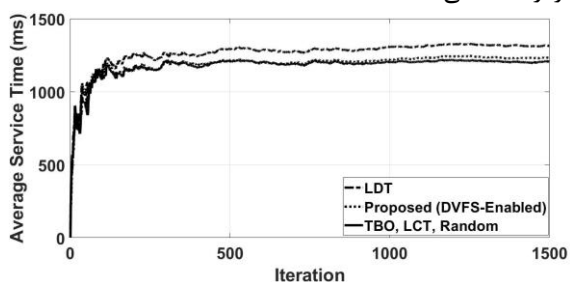
نماد	توضیح	مقدار
M	تعداد هسته‌های پردازشی و مسیریاب‌ها	۷×۷ و ۶×۶، ۵×۵، ۴×۴، ۳×۳
λ	نرخ تجمعی ورود کارها	۸.۴۱ کاربرثانیه
ξ	نرخ نمایی زمان تبادل داده‌ی کارها	میانگین زمان اجرای دو کار جفت شده
T_{th}	آستانه‌ی مجاز دما	۳۵۸ کلوین
α	نرخ یادگیری	$50/(1000+k)$
σ	عرض ویژگی‌ها	۰.۰۹
ω	مراکز ویژگی‌ها	۰.۳۳ و ۰.۶۶
Z	تعداد توابع ویژگی	۵۱۲

یک از هسته‌ها مقداری را به عنوان هزینه تخصیص می‌دهد. سپس، انتساب کارها به هسته‌ها را در راستای حداقل‌سازی آن هزینه انجام می‌دهد. مشابه ما، راهکار [۱۱] نیز انتساب کارها را بر اساس یک MDP صورت می‌دهد؛ اما، در حالیکه تمامی روابط استفاده شده در [۱۱] برای MDP زمان-گسسته می‌باشند، عملاً در شبیه‌سازی، به صورت رویداد-محور عمل شده است. بنابراین، به لحاظ فرمول‌بندی، فرمالیسم صحیحی به کار گرفته نشده است. در اینجا، دو نسخه از این-

استفاده می‌کند و در نتیجه گاه‌ها زمان سرویس برخی کارها بالاتر می‌رود. این در حالیست که سایر روش‌ها همگی بر اساس یک سطح ولتاژ-فرکانس ثابت عمل می‌نمایند. اما از سوی دیگر، با توجه به این‌که روش DVFS-Enabled عملکرد زمان-پیوسته دارد، این افزایش زمان تا حدی جبران شده است.

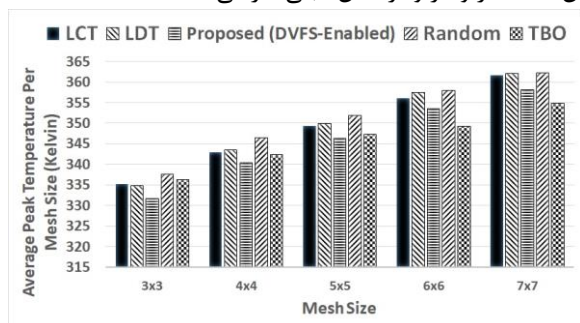
لازم به ذکر است که روش‌های TBO، LCT و Random به این دلیل که به صورت زمان-پیوسته پیاده‌سازی شده‌اند و از سطوح متوسط و ثابت ولتاژ و فرکانس استفاده می‌کنند، متوسط زمان اجرای کمتری نسبت به سایر روش‌ها از خود نشان داده‌اند.

مطابق شکل (۳) با افزایش تعداد هسته‌ها، الگوریتم DVFS-Enabled با افت عملکرد نسبت به الگوریتم TBO مواجه است. علت عملکرد بهتر TBO با افزایش اندازه‌ی Mesh این است که TBO از انتساب کارها به هسته‌های نزدیک به مرکز تراشه اجتناب کرده و ترجیح در انتخاب هسته‌های دورتر از مرکز را دارد. به دلیل احاطه شدن کمتر این هسته‌ها با هسته‌های دیگر، هسته‌های نزدیک لبه‌ها تأثیرپذیری دمایی کمتری از سایر هسته‌ها پذیرفته و گرمای خود را سریع‌تر از دست می‌دهند.



شکل (۲): میانگین زمان سرویس کارها

مطابق شکل (۴)، با افزایش نرخ ورود کارها دمایی بیشینه‌ی پردازنده افزایش می‌یابد. علت این امر، افزایش تعداد کارهای در حال اجرا در سیستم است که سبب بالا رفتن دمایی پردازنده می‌شود. همچنین، علت عملکرد بهتر روش DVFS-Enabled نسبت به روش LCT، استفاده از سطوح ولتاژ و فرکانس مختلف است. به این ترتیب که این روش، برای اجرای برخی کارها، با کاهش ولتاژ و فرکانس کاری هسته، مانع از افزایش دمایی پردازنده می‌شود. این در حالی است که روش LCT در ولتاژ و فرکانس ثابتی کار می‌کند.



شکل (۳): میانگین دمایی بیشینه بر حسب اندازه‌ی Mesh

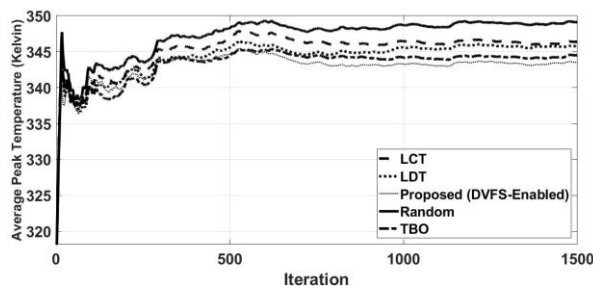
الگوریتم بر اساس دو فرمول‌بندی کاملاً زمان-گسسته (LDT) و فرمول‌بندی کاملاً زمان-پیوسته (LCT) پیاده‌سازی شده است. LDT با استفاده از روبرط موجود در [۱۱] شبیه‌سازی شده‌اند. برای پیاده‌سازی LCT نیز از [۲۰] استفاده شده است. راهکار پیشنهادی ما نیز با عنوان (DVFS-Enabled) مشخص می‌شود. در روش Random نیز انتساب کارها به صورت کاملاً تصادفی می‌باشد.

۴-۴- معیارهای ارزیابی

- متوسط دمایی بیشینه‌ی پردازنده: هدف نهایی راهکار ارائه‌شده کاهش مقدار متوسط دمایی بیشینه‌ی پردازنده در درازمدت است. بنابراین، کمتر بودن این معیار در هر یک از روش‌ها نشان از عملکرد بهتر در راستای رسیدن به هدف نهایی می‌باشد.
- متوسط زمان سرویس کارها: زمان سرویس به فاصله‌ی زمانی ورود یک کار به سیستم و پایان اجرای آن اطلاق می‌شود. راهکاری که زمان سرویس کمتری داشته باشد، کارایی بالاتری دارد.
- مقدار توان پویا: مقدار توان پویا به سطح ولتاژ و فرکانس کاری هسته‌های پردازنده بستگی دارد. به دلیل محدودیت در بودجه‌ی توان، کاهش توان پویا مطلوب است [۲۷].

۴-۵- آزمایش‌ها و تحلیل نتایج شبیه‌سازی

مطابق شکل (۱) روش DVFS-Enabled در مقایسه با روش‌های LDT و LCT در حدود ۳ کلوین و در مقایسه با روش‌های TBO و Random به ترتیب، حدوداً ۱،۵ و ۶ کلوین دمایی بیشینه‌ی پردازنده را کاهش داده است.



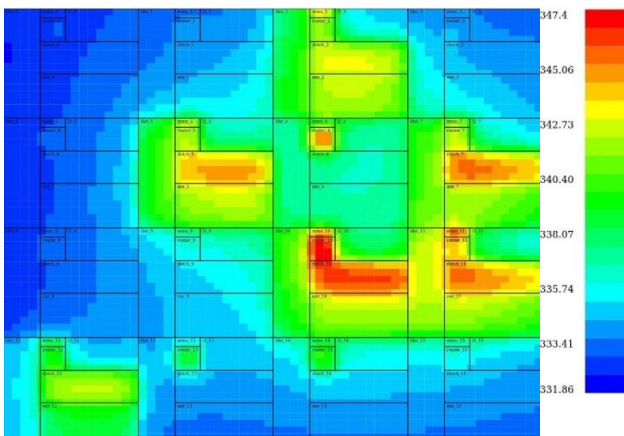
شکل (۱): میانگین دمایی بیشینه‌ی پردازنده

طبق شکل (۲) روش DVFS-Enabled حدوداً به اندازه‌ی ۶۶ میلی‌ثانیه زمان سرویس کارها را کاهش داده است. علت عملکرد بهتر این روش نسبت به LDT این است که روش LDT ماهیت زمان-گسسته دارد. به این معنا که تنها در ابتدای بازه‌های زمانی خاص اقدام به انتساب یکی از کارها نموده و تا ابتدای بازه‌ی بعدی، هیچ انتسابی انجام نمی‌دهد. به این ترتیب، زمان انتظار کارها در سیستم افزایش پیدا می‌کند. علت بیشتر بودن جزئی متوسط زمان سرویس کارها در روش DVFS-Enabled در مقایسه با روش‌های LCT، TBO و Random این است که روش DVFS-Enabled، برخلاف سایر روش‌ها، برای اجرای برخی از کارها، از سطوح ولتاژ و فرکانس پایین‌تری

- کار در حال اجرا بر هسته‌ی ۵ با کار در حال اجرا بر هسته‌ی ۷ جفت شده و مسیریاب‌های مؤثر در برقراری ارتباط، مسیریاب‌های ۵، ۶ و ۷ می‌باشند.
- همان‌طور که در تصویر مشاهده می‌شود، با وجود عدم انتساب کار به هسته‌ی ۶، تجمع کارها اطراف آن سبب تأثیرپذیری گرمایی و بالا رفتن دمای هسته‌ی ۶ شده است.

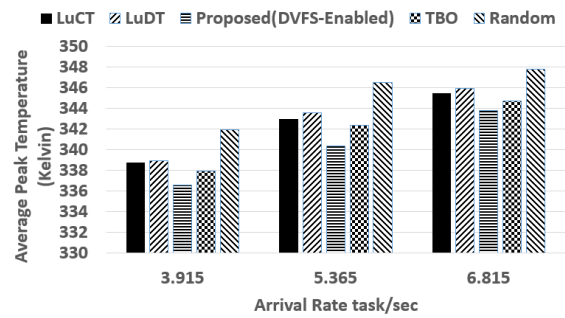
همان‌طور که در شکل (۵) مشهود است، دمای بیشینه‌ی پردازنده برابر است با ۳۵۴٫۳۱ درجه کلوین. این درحالی است که با استفاده از الگوریتم Proposed (DVFS-Enabled) دمای بیشینه‌ی تراشه برای انتساب کارهای مشابه به ۳۴۷٫۴ درجه کلوین کاهش یافته است (شکل (۶)). در واقع، هدف ما از ارائه این الگوریتم، ایجاد یکنواختی و تعادل دمایی در سطح تراشه نیست؛ بلکه، کاهش دمای بیشینه‌ی تراشه است.

برخی از راهکارهای ارائه شده در حوزه‌ی مدیریت حرارتی پردازنده، سعی در کاهش دمای بیشینه با ایجاد تعادل دمایی در سطح تراشه دارند. به عنوان مثال، در مقاله‌ی [۱۱] نشان داده شده است که راهکار TBO با استفاده از توزیع یکنواخت دما در سطح تراشه، عملکردی بهتری (به لحاظ کاهش دمای بیشینه و افزایش سرعت اجرای کارها) در مقایسه با سایر روش‌ها داشته است. این درحالی است که علی‌رغم عدم تضمین یکنواختی دما در روش ما، عملکرد روش پیشنهادی DVFS-Enabled به لحاظ سرعت انجام محاسبات و میانگین زمان تکمیل سرویس بسیار نزدیک به TBO می‌باشد.



شکل (۶): انتساب کارها پس از یادگیری با الگوریتم Proposed (DVFS-Enabled)

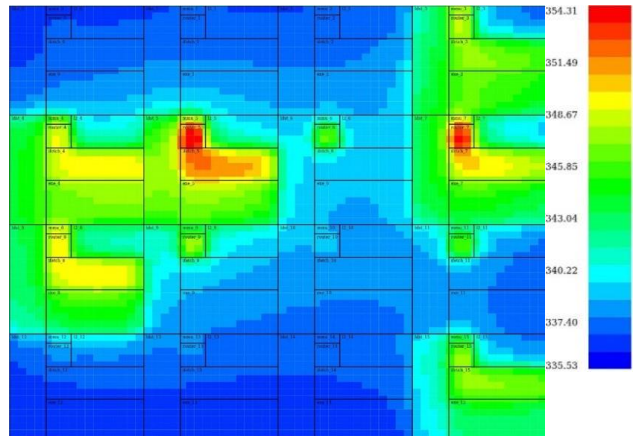
در آخرین قسمت از بخش شبیه‌سازی، به بررسی تأثیر افزایش دما بر سرعت اجرای کارها پرداخته شده است. با توجه به این‌که شبیه‌سازی‌های پردازنده‌های چند هسته‌ای موجود، قادر به انجام شبیه‌سازی اجرای برنامه‌ها در دماهای مختلف نمی‌باشند، لازم است تأثیر دما بر تأخیر مدار محاسبه شده و نهایتاً میزان تأخیر به دست آمده در



شکل (۴): میانگین دمای بیشینه بر حسب نرخ ورود

شکل (۵)، پروفایل گرمایی به انتساب کارها قبل از یادگیری را نشان می‌دهد که در آن ترتیب جفت شدن کارها به صورت زیر می‌باشد:

- کار در حال اجرا بر هسته‌ی ۳ با کار در حال اجرا بر هسته‌ی ۱۵ (شماره‌گذاری از ابتدای سمت چپ هر سطر بوده و از صفر شروع شده است) جفت شده و مسیریاب‌های مؤثر در برقراری ارتباط، مسیریاب‌های ۳، ۷، ۱۱ و ۱۵ می‌باشند.
- کار در حال اجرا بر هسته‌ی ۵ با کار در حال اجرا بر هسته‌ی ۸ جفت شده و مسیریاب‌های مؤثر در برقراری ارتباط، مسیریاب‌های ۵، ۹ و ۸ می‌باشند.
- کار در حال اجرا بر هسته‌ی ۴ با کار در حال اجرا بر هسته‌ی ۷ جفت شده و مسیریاب‌های مؤثر در برقراری ارتباط، مسیریاب‌های ۴، ۵، ۶ و ۷ می‌باشند.



شکل (۵): انتساب کارها قبل از یادگیری

شکل (۶) انتساب کارها را پس از یادگیری با الگوریتم Proposed (DVFS-Enabled) نشان می‌دهد که در آن:

- کار در حال اجرا بر هسته‌ی ۱۲ با کار در حال اجرا بر هسته‌ی ۲ جفت شده و مسیریاب‌های مؤثر در برقراری ارتباط، مسیریاب‌های ۲، ۱۲، ۱۳، ۱۴، ۱۰، ۶ و ۲ می‌باشند.
- کار در حال اجرا بر هسته‌ی ۱۰ با کار در حال اجرا بر هسته‌ی ۱۱ جفت شده و مسیریاب‌های مؤثر در برقراری ارتباط، مسیریاب‌های ۱۰ و ۱۱ می‌باشند.

فرکانس کاری پردازنده اعمال شود. سپس، شبیه‌سازی مجددا در فرکانس محاسبه شده، صورت گیرد.

نکته‌ی قابل توجه این است که طبق [۲۸]، در تکنولوژی‌های زیر ۹۰ نانومتر، کارایی مدار تحت تاثیر عواملی چون وابستگی دمایی معکوس^{۱۳} قرار می‌گیرد. به این معنا که زمانی که مدار با ولتاژ پایین کار می‌کند، تاخیر انتشار با افزایش دما کاهش می‌یابد [۲۹]. در مقاله-ی [۳۰] نیز به این نکته اشاره شده است که در تکنولوژی ۴۵ نانومتری CMOS، با افزایش دما سرعت مدار (در ولتاژ نامی منبع) بهبود می‌یابد.

باتوجه به این که دما در میزان تاخیر ترانزیستورهای سازنده‌ی مدار، تاثیرگذار است، لازم است تاثیر افزایش دما بر میزان تاخیر ترانزیستورها محاسبه شده و تاثیر آن در فرکانس کاری مدار اعمال گردد. برای این منظور، از [۳۰] برای استخراج مقادیر تاخیر دروازه^{۱۴} های مختلف در دماهای ۲۵ و ۱۲۵ درجه‌ی سانتی‌گراد، برای تکنولوژی ۴۵ نانومتر استفاده شده است که در آن، به عنوان مثال، تاخیر دروازه‌ی NOR ۴ ورودی در دماهای ۲۵ و ۱۲۵ درجه‌ی سانتی‌گراد، به ترتیب برابر با $3,55E-10$ و $3,42E-10$ ثانیه می‌باشد. به عنوان مثال، در صورت اجرای برنامه‌ی FFT از مجموعه برنامه-های محک SPLAS2 در دمای ۲۵ درجه‌ی سانتی‌گراد، با ولتاژ ۰٫۸ ولت و فرکانس ۲٫۴ گیگاهرتز، زمان اجرا برابر خواهد بود با ۲۳٫۶۴ میکروثانیه. از آنجایی که فرکانس برابر است با تاخیر/۱، میزان تاخیر 10^{-10} ثانیه خواهد بود.

همانطور که پیش‌تر گفته شد، در دمای ۱۲۵ درجه‌ی سانتی‌گراد، میزان تاخیر هر دروازه‌ی NOR ۴ ورودی برابر است با $3,42E-10$ که به معنای کاهش 10^{-10} ، $1,3E-10$ ثانیه‌ای در مقایسه با میزان تاخیر در دمای ۲۵ درجه‌ی سانتی‌گراد می‌باشد. حال، در صورتی که پردازنده از ۱۰ لایه تشکیل شده و در هر لایه دروازه‌ی NOR ۴ ورودی داشته باشد، کل تاخیر مدار به اندازه‌ی $10^{-10} \times 10 = 1,3E-10$ کاهش یافته و برابر با 10^{-10} ، $2,8E-10$ ، $1,3E-10$ و $4,1E-10$ ثانیه خواهد بود. بنابراین، برای مشاهده‌ی تاثیر افزایش دما بر سرعت اجرای برنامه‌ی FFT، این بار شبیه‌سازی در فرکانس $3,57E-10 = 1/2,8E-10$ انجام شده و زمان اجرای برنامه‌ی FFT برابر با ۱۷٫۷۱ میکروثانیه بوده است. همانطور که از نتایج پیداست، در ازای افزایش دما به اندازه‌ی ۱۰۰ درجه‌ی سانتی‌گراد، زمان اجرای برنامه‌ی FFT به اندازه‌ی ۵٫۹۳ میکروثانیه کاهش یافته است.

۵- نتیجه‌گیری

در این مقاله، مسئله‌ی انتساب برخط کارها به هسته‌های یک سیستم پردازشی چندهسته‌ای با چارچوب ریاضی MDP مدل شده است. نوآوری روش پیشنهادی در هم‌افزایی آن با تکنیک DVFS و همچنین، لحاظ کردن سیستماتیک عوامل تصادفی محیط عملیاتی (شامل ورود تصادفی کارها، IPC و تغییرات دمایی تراشه) در فرآیند

تصمیم‌گیری است. برای محاسبه سیاست بهینه تصمیم‌گیری، یک روال مبتنی بر یادگیری تقویتی پیشنهاد شده است که قادر است کارایی متوسط سیستم در درازمدت را بیشینه‌سازی نماید. علاوه بر بهینگی نظری، روش پیشنهادی از طریق شبیه‌سازی با کارهای مشابه مقایسه شده است. نتایج حاکی از آن است که روش ارائه شده به لحاظ کاهش حداکثر دمای پردازنده نسبت به سایر روش‌ها عملکرد بهتری دارد. همچنین، در رابطه با کاهش توان مصرفی پویا، روش ارائه شده به دلیل کاهش سطوح ولتاژ و فرکانس کاری هسته‌ها، عملکرد بهتری داشته است. از نظر زمان سرویس کارها نیز این روش بخاطر ماهیت زمان-پیوسته عملکرد بهتری نسبت به راهکارهای زمان-گسسته دارد که تاثیر این مورد هنگامی مشهودتر است که نرخ ورود کارها افزایش می‌یابد.

مراجع

- [1] Hennessy, John L., Patterson, David A., Computer architecture: a quantitative approach, Elsevier, 2012.
- [2] Coskun, Ayse K., Tajana, Simunic R., Whisnant K., "Temperature aware task scheduling in MPSoCs", Proceedings on Design, automation and test in Europe: EDA Consortium, pp. 1659-1664, 2007.
- [3] Pruhs, K., Stee, R., Uthaisombut, P., "Speed scaling of tasks with precedence constraints", Theory of Computing Systems Vol. 43, No. 1, pp. 67-80, 2008.
- [4] Herbert, S., Marculescu, D., "Analysis of dynamic voltage/frequency scaling in chip-multiprocessors", Proceedings of the international symposium on Low power electronics and design (ISLPED), pp. 38-43, 2007.
- [5] Kolpe, T., Zhai, A., Sapatnekar, Sachin, S., "Enabling improved power management in multicore processors through clustered DVFS", Design, Automation & Test in Europe, pp. 1-6, 2011.
- [6] Donald, J., Martonosi, M., "Techniques for multicore thermal management: Classification and new exploration", ACM SIGARCH Computer Architecture News, Vol. 34, No. 2, pp. 78-88, 2006.
- [7] Zhang, S., Chatha, Karam, S., "Approximation algorithm for the temperature-aware scheduling problem", Proceedings of the IEEE/ACM international conference on Computer-aided design, pp. 281-288, 2007.
- [8] Yeo, I., Chih, Chun, L. Eun, Jung, K. "Predictive dynamic thermal management for multicore systems", Proceedings of the 45th ACM annual Design Automation Conference, pp. 734-739, 2008.
- [9] Stavrou, K., Trancoso, P., "Thermal-aware scheduling for future chip multiprocessors", EURASIP Journal on Embedded Systems, pp.1-15, 2007.
- [10] Wang, J., Lu, J., Guo, S., Chen, Z., Li, Y., "A Thermal Balance Oriented Task Mapping for CMPs", Proceedings of the 8th International Conference on Information Communication and Management, pp. 12-16, 2018.
- [11] Lu, Shiting. J., Tessier, R., Bursleson, W., "Reinforcement Learning For Thermal-Aware Many-Core Task Allocation", ACM Proceedings of the 25th edition on Great Lakes Symposium on VLSI, pp. 379-384, 2015.
- [12] Das, A., Rishad, S., Geoff, M., Bashir, M., Akash, K., Bharadwaj, V., "Reinforcement learning-based inter-and intra-application thermal optimization for lifetime improvement of multicore systems", ACM Proceedings

[30] Kumar, R., Kursun, V., "Reversed temperature-dependent propagation delay characteristics in nanometer CMOS circuits", Vol. 53, No. 10, pp. 1078-1082, 2006.

زیر نویس ها

- ¹ Dynamic Power Management
- ² Dynamic Voltage and Frequency Scaling
- ³ Inter Process Communication
- ⁴ Near-Sighted
- ⁵ Markov Decision Process
- ⁶ Semi-Markov Decision Process
- ⁷ First Come First Served
- ⁸ Feature Function
- ⁹ Radial Basis Function
- ¹⁰ Homogenous
- ¹¹ Injection Rate
- ¹² Bilinear Interpolation
- ¹³ Inverted Temperature Dependence (ITD)
- ¹⁴ Gate

of the 51st Annual Design Automation Conference, pp. 1-6. 2014.

[13] Das, A., Bashir, M., Merrett, G., "Adaptive and hierarchical runtime manager for energy-aware thermal management of embedded systems", Vol. 15, No. 2, p. 1-25, 2016.

[14] Bertsekas, Dimitri P., Dynamic programming and optimal control, Athena scientific, 1995.

[15] Karimi, A., Taherpour, A., "Spectrum Sensing in the Presence of Smart Primary User Emulation Attacker and Its Impact on Performance and Throughput of Cognitive Radio Networks", Journal of Iranian Association of Electrical and Electronics Engineers, Vol. 16, No. 4, pp. 101-111, 2019.

[16] Younesi, A., Shayeghi, H., Akbari, A., Hashemi, E., "Design of PSS3B stabilizer using KH Algorithm and Q-Learning for damping Low-frequency Oscillations in SMIB", Journal of Iranian Association of Electrical and Electronics Engineers, Vol. 14, No. 3, pp. 69-77, 2017.

[17] Tatas, K., Siozios, K., Soudris, D., Jantsch, A., Designing 2D and 3D network-on-chip architectures, Springer, 2014.

[18] Chou, C-H, Mehmet, B., Laxmi, B., "Thermal prediction and scheduling of network applications on multicore processors", Proceedings of the ninth ACM/IEEE symposium on Architectures for networking and communications systems, pp. 115-116, 2013.

[19] Brill, Percy H., Level Crossing Methods in Stochastic Models, Springer, 2017.

[20] Gosavi A., "Relative value iteration for average reward semi-Markov control via simulation", IEEE Winter Simulation Conference, pp. 623-630, 2013.

[21] Sutton, Richard S., Barto, Andrew G., Introduction to reinforcement learning, MIT press, 1998

[22] Carlson, E., Heirman, W., Eeckhout L., "Sniper: exploring the level of abstraction for scalable and accurate parallel multi-core simulation", ACM Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, 2011.

[23] (2019-01-23). HotSpot thermal simulator, Available: <http://lava.cs.virginia.edu/HotSpot>

[24] Li, S., Jung, A., Richard, S., Jay, B., Dean, T., Norman, J., "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures", Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, pp. 469-480, 2009.

[25] Sun, C., "DSENT-a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling", Sixth IEEE/ACM International Symposium on Networks on Chip (NoCS), pp. 201-210, 2012.

[26] Rantala, V., Lehtonen, T., Plosila, J., Network on chip routing algorithms, Citeseer, 2006.

[27] Mehri, H., Alizadeh, B., "Analytical Performance Model for FPGA-based Reconfigurable Computing", Journal of Iranian Association of Electrical and Electronics Engineers, Vol. 13, No. 4, pp. 1-13, 2017.

[28] Wu, Sean H., Tetelbaum, A., Wang, Li-C, "How does inverse temperature dependence affect timing sign-off", Emerging Technologies and Circuits, pp. 179-189, 2010.

[29] Dasdan, A., Hom, I., "Handling inverted temperature dependence in static timing analysis", Vol. 11, No. 2, pp. 306-324, 2006.