



دانشگاه آزاد اسلامی

واحد علوم و تحقیقات (تهران)

Science and Research Branch, Islamic Azad University

فرم پیشنهاد تحقیق

رساله دکتری

عنوان تحقیق به فارسی:

نام: _____
نام خانوادگی دانشجو: _____
رشته تحصیلی: _____
سال ورود به مقطع جاری: _____
دانشکده: _____
گروه تخصصی: _____
گرایش: _____
نیمسال ورودی: _____

نام و نام خانوادگی استاد (اساتید) راهنما: _____
نام و نام خانوادگی استاد (اساتید) مشاور: _____

-1 _____
-2 _____

تأیید شورای تحقیق:

این قسمت توسط حوزه معاونت پژوهشی واحد تکمیل میگردد

تاریخ تصویب در شورای گروه: _____ تاریخ دریافت حوزه پژوهشی واحد: _____

تاریخ تصویب در شورای پژوهشی دانشکده: _____ تاریخ تصویب در حوزه پژوهشی واحد: _____

تأیید مدیر کل پژوهشی

تأیید کارشناس پژوهشی

تأیید معاون پژوهشی

تأیید دفتر امور پژوهشی

به نام خدا

توجه: لطفاً این فرم با مساعدت و هدایت استاد راهنما تکمیل شود.

1- اطلاعات مربوط به دانشجو:

نام: نام خانوادگی: شماره دانشجویی:
مقطع: رشته تحصیلی: گروه تخصصی:
گرایش: نام دانشکده: سال ورود به مقطع جاری:
نیمسال ورودی:
آدرس پستی در تهران:
تلفن ثابت محل سکونت: تلفن همراه: پست الکترونیک:
آدرس پستی در شهرستان:
تلفن ثابت محل سکونت: تلفن محل کار: دورنگار:

2- اطلاعات مربوط به استاد راهنما:

تذکرات:

- دانشجویان دوره کارشناسی می‌توانند یک استاد راهنما و حداکثر دو استاد مشاور و دانشجویان دوره دکتری حداکثر تا دو استاد راهنما و دو استاد مشاور می‌توانند انتخاب نمایند.
- در صورتی که اساتید راهنما و مشاور **مدعو** می‌باشند، لازم است سوابق تحصیلی، آموزشی و پژوهشی کامل ایشان (رزومه کامل) شامل فهرست پایان‌نامه‌های کارشناسی ارشد و رساله‌های دکتری دفاع شده و یا در حال انجام که اساتید مدعو، راهنمایی و یا مشاوره آنرا بر عهده داشته‌اند، **به همراه مدارک** مربوطه و همچنین آخرین حکم کارگزینی (حکم هیأت علمی) ضمیمه گردد.
- اساتید راهنما و مشاور موظف هستند قبل از پذیرش پروپوزال، به سقف ظرفیت پذیرش خود توجه نموده و در صورت تکمیل بودن ظرفیت پذیرش، از ارسال آن به دانشکده و حوزه پژوهشی و یا در نوبت قرارداد و ایجاد وقفه در کار دانشجویان جداً پرهیز نمایند.

اطلاعات مربوط به استاد راهنمای اول:

نام و نام خانوادگی: آخرین مدرک تحصیلی _____: دانشگاه
حوزوی
عضو هیأت علمی دانشگاه
تخصص اصلی: رتبه دانشگاهی (مرتبه علمی): تلفن همراه:
تلفن منزل یا محل کار: نام و نام خانوادگی به زبان انگلیسی:
نحوه همکاری با واحد علوم و تحقیقات:
 تمام وقت نیمه وقت مدعو

اطلاعات مربوط به استاد راهنمای دوم:

نام و نام خانوادگی: آخرین مدرک تحصیلی _____: دانشگاه
حوزوی
عضو هیأت علمی دانشگاه
تخصص اصلی: رتبه دانشگاهی (مرتبه علمی): تلفن همراه:
تلفن منزل یا محل کار: نام و نام خانوادگی به زبان انگلیسی:
نحوه همکاری با واحد علوم و تحقیقات:
 تمام وقت نیمه وقت مدعو

اطلاعات مربوط به استاد مشاور اول:

نام و نام خانوادگی: آخرین مدرک تحصیلی _____: دانشگاه
حوزوی
عضو هیأت علمی دانشگاه
تخصص اصلی: رتبه دانشگاهی (مرتبه علمی): تلفن همراه:
تلفن منزل یا محل کار: نام و نام خانوادگی به زبان انگلیسی:
نحوه همکاری با واحد علوم و تحقیقات:
 تمام وقت نیمه وقت مدعو

اطلاعات مربوط به استاد مشاور دوم:

نام و نام خانوادگی: آخرین مدرک تحصیلی _____: دانشگاه
حوزوی
عضو هیأت علمی دانشگاه
تخصص اصلی: رتبه دانشگاهی (مرتبه علمی): تلفن همراه:
تلفن منزل یا محل کار: نام و نام خانوادگی به زبان انگلیسی:
نحوه همکاری با واحد علوم و تحقیقات:
 تمام وقت نیمه وقت مدعو

الف- عنوان تحقیق

1- عنوان به زبان فارسی:

پیش بینی کارآیی الگوریتم‌های برنامه‌نویسی پویا روی پردازنده‌ی گرافیکی بر مبنای مدل BSP

2- عنوان به زبان انگلیسی / (آلمانی، فرانسه، عربی):

تذکره: صرفاً دانشجویان رشته‌های زبان آلمانی، فرانسه و عربی مجازند عنوان رساله خود را به زبان مربوطه در این بخش درج نمایند و برای بقیه دانشجویان، عنوان بایستی به زبان انگلیسی ذکر شود.

Performance Prediction of Dynamic Programming Algorithms on GPUs Based on BSP Model

ب - تعداد واحد رساله:

ج- بیان مسأله اساسی تحقیق به طور کلی (شامل تشریح مسأله و معرفی آن، بیان جنبه‌های مجهول و مبهم، بیان متغیرهای مربوطه و منظور از تحقیق):

واحد پردازنده‌ی گرافیکی¹ دارای هسته‌های پردازشی زیادی است که هر هسته قابلیت‌های یک واحد محاسبات و منطق در پردازنده‌ی اصلی را دارد. همچنین مدل اجرای این پردازنده به صورت موازی داده² است [1] و به همین دلیل دارای قابلیت‌های موازی‌سازی فوق‌العاده‌ای در محاسبات آرایه‌ای، پردازش تصویر و محاسبات علمی است. مهم‌ترین زبان برنامه‌نویسی روی این پردازنده‌ها کودا³ [2] است که محصول شرکت انویدیا است. به دلیل جزئیات کمی که درباره‌ی این پردازنده‌ها منتشر شده است، تنظیم⁴ این برنامه‌ها یک چالش جدی برای برنامه‌نویسان است و بسیاری از طراحی‌ها و پیکربندی‌های این برنامه‌ها به صورت آزمون و خطا انجام می‌شود [3]. به همین دلیل نیازمند مدل‌های کارآیی هستیم که زمان اجرای این برنامه‌ها را پیش‌بینی کنند.

در این تحقیق هدف پیش‌بینی زمان اجرای برنامه‌های کودا است. ما منحصراً روی برنامه‌هایی متمرکز می‌شویم که بر مبنای الگوریتم‌های برنامه‌نویسی پویا⁵ ایجاد شده‌اند و دامنه‌ی ورودی مدل کارآیی، این برنامه‌ها هستند. برای این منظور، یک مدل تحلیلی⁶ از عملکرد برنامه‌های کودا ارائه می‌کنیم. مدل‌های تحلیلی با استفاده از

¹ Graphic Processing Unit (GPU)

² Data Parallel

³ Compute Unified Device Architecture (CUDA)

⁴ Tuning

⁵ Dynamic Programming

⁶ Analytical Model

پارامترها و روابط ریاضی سعی می‌کنند ویژگی‌ها و عملکرد برنامه‌های ورودی را مدل‌سازی کنند. ویژگی بارز این مدل‌ها این است که قابلیت حمل فوق‌العاده‌ای دارند و می‌توان آن‌ها را روی هر دستگاهی بکار گرفت [4]. هر چند که ایجاد یک مدل تحلیلی بسیار سخت و زمان‌بر است و نیاز به دانش زیادی از عملکرد دستگاه دارد. نقطه‌ی مقابل مدل‌های تحلیلی، مدل‌های مبتنی بر یادگیری ماشین هستند که قابلیت حمل پایینی دارند و در عوض به آسانی تولید می‌شوند. در شکل 1 هدف این مدل تحلیلی را به صورت یک جعبه سیاه نشان داده‌ایم. ورودی مدل منحصرًا برنامه‌های کودایی هستند که مبتنی بر الگوریتم‌های برنامه‌نویسی پویا طراحی شده‌اند. خروجی مدل نیز زمان اجرای پیش‌بینی شده برای هر برنامه است.



شکل 1 نمای کلی مدل کارآیی پیشنهادی

مدل کارآیی پیشنهادی در این تحقیق براساس مدل BSP⁷ [5] ایجاد می‌شود. این مدل ابتدا در سال 1990 برای سیستم‌های توزیع شده ایجاد شده است. سپس در سال 2011 برای سیستم‌های چند هسته‌ای توسعه یافته است. مدلی که Multi-BSP [6] نام گرفت. همچنین در سال 2015 از این مدل برای پیش‌بینی کارآیی برنامه‌های کودا استفاده شده است [7] که البته مدل ضعیفی بوده است. این مدل کارآیی یک برنامه‌ی توزیع شده را توسط سه پارامتر برای تاخیر محاسبات، تاخیر ارتباطات و تاخیر همگام‌سازی⁸ مدل‌سازی می‌کند. این سه مرحله در هر تکرار از برنامه‌های مبتنی بر الگوریتم‌های برنامه‌نویسی پویا دیده می‌شوند. به همین دلیل ما براساس این مدل تاخیرهای⁹ محاسباتی¹⁰، ارتباطاتی¹¹ و همگام‌سازی این برنامه‌ها را روی پردازنده‌ی گرافیکی محاسبه می‌کنیم. ابزارهای ما برای ایجاد مدل، ریزمحک‌زنی¹² و تحلیل کدهای کودا و PTX¹³ [8] است. جزئیات بیشتر در بخش روش تحقیق بیان خواهند شد.

⁷ Bulk Synchronouse Parallel

⁸ Synchronization

⁹ Latency

¹⁰ Computation

¹¹ Commenciation

¹² Micro-benchmarking

¹³ Parallel Thread Execution

د - اهمیت و ضرورت انجام تحقیق (شامل اختلاف نظرها و خلاءهای تحقیقاتی موجود، میزان نیاز به موضوع، فواید احتمالی نظری و عملی آن و همچنین مواد، روش و یا فرآیند تحقیقی احتمالاً جدیدی که در این تحقیق مورد استفاده قرار می‌گیرد:

ایجاد یک مدل تحلیلی از پردازنده‌ی گرافیکی، به درک بهتر ما نسبت به این دستگاه کمک شایانی می‌کند. متأسفانه اطلاعات کمی از این دستگاه توسط ایجادکنندگان آن منتشر شده است. بسیاری از عملکردهای این دستگاه در سطح راه انداز^{۱۴} انجام می‌شود و برنامه‌نویس هیچ کنترلی روی آن ندارد. همین موضوع باعث شده که تنظیم برنامه‌ها روی این دستگاه، به صورت آزمون و خطا انجام می‌شود. داشتن یک مدل کارآیی تحلیلی باعث شفاف‌تر شدن گلوگاه‌های کارآیی و شناخت بهتر عملکرد دستگاه می‌شود. با توجه به اینکه در مدل تحلیلی ویژگی‌ها و اجزای دستگاه به صورت پارامتر و روابط ریاضی ارائه می‌شوند، بنابراین اجزای تاثیرگذار روی کارآیی تا حدودی شناخته می‌شوند و چراغ راهی برای برنامه‌نویس برای تنظیم بهتر برنامه‌ها می‌شوند.

پیش‌بینی زمان اجرای برنامه‌ها کاربردهای زیادی دارند که برای مرتفع کردن این نیازمندی‌ها لازم است که یک مدل کارآیی ایجاد شود. در ادامه به چند نمونه از این کاربردها اشاره می‌کنیم:

- همیشه در اجرای برنامه‌های موازی سربارهایی وجود دارد که باعث افت تسریع^{۱۵} می‌شود. مثلاً تداخل‌هایی که در سطوح حافظه رخ می‌دهد و باعث می‌شوند که نخ‌های موازی با تاخیر ارتباطاتی مواجه شوند. همچنین در برنامه‌های موازی مراحل تجزیه و ترکیب وجود دارند که باعث افت کارآیی و کاهش تسریع می‌شوند. حتی در برخی موارد این سربارها به قدری زیاد هستند که تسریع کمتر از یک می‌شود. به عبارت دیگر اجرای موازی برنامه‌ها در مقابل نسخه‌ی ترتیبی مقرون به صرفه نیست. بنابراین نیازمند یک مدل کارآیی هستیم تا بتوانیم زمان اجرای برنامه‌ها را تخمین بزنیم و به این ترتیب بدون اجرای برنامه‌ها نسخه‌ی بهینه‌تری از برنامه (موازی یا ترتیبی) را فراخوانی کنیم.

- در شبکه‌های محاسبات ابری نیازمند این هستیم که اجرای برنامه‌ها را تا یک مهلت زمانی^{۱۶} تضمین کنیم. بنابراین در این موارد نیاز است که بدون اینکه برنامه را اجرا کنیم، زمان اجرای آن را تخمین بزنیم و سپس به لحاظ مهلت زمانی، عملی بودن اجرای یک برنامه را تضمین کنیم.

¹⁴ Driver

¹⁵ Speedup

¹⁶ Deadline

- برخی از سیاست‌های زمانبندی پردازنده در سیستم عامل مانند SRT¹⁸ و SJF¹⁷ و نیازمند این هستند که از بین پردازش‌های آماده اجرا، آن که زمان اجرای کمتری دارد را شناسایی کنند. این یک چالش در سیستم عامل است، چراکه سیستم عامل قادر به انجام آن نیست. ایجاد یک مدل کارآیی برای برنامه‌ها به سیستم عامل در تخمین زمان اجرای پردازش‌ها کمک می‌کند.

ه- مرور ادبیات و سوابق مربوطه (بیان مختصر پیشینه تحقیقات انجام شده در داخل و خارج کشور پیرامون موضوع تحقیق و نتایج آنها و مرور ادبیات و چارچوب نظری تحقیق):

این بخش شامل چهار زیربخش است. ابتدا معماری پردازنده‌های شرکت انویدیا تشریح می‌شوند و سپس به اختصار مدل برنامه‌نویسی کودا معرفی می‌شود. با توجه به اینکه مدل پیشنهادی در این تحقیق بر مبنای مدل محاسباتی BSP بنا خواهد شد، در ادامه این مدل بیان می‌شود و در انتها نیز پیشینه‌ی تحقیق بررسی خواهد شد.

▪ معماری پردازنده‌ی گرافیکی:

معماری پردازنده‌های گرافیکی شرکت انویدیا در شکل 2 نشان داده شده است. این معماری شامل یک حافظه‌ی سراسری¹⁹، یک حافظه‌ی نهان L2²⁰ و تعدادی چندپردازنده‌های جریان‌ی (SM²¹) است. حافظه‌ی سراسری تنها حافظه‌ای است [9] که با پردازنده‌ی اصلی که معمولاً آن را میزبان²² می‌نامیم (در مقابل پردازنده‌ی گرافیکی را دستگاه²³ می‌نامیم)، در ارتباط است و داده‌ها بین این حافظه و حافظه‌ی اصلی سمت میزبان (حافظه‌ی RAM) می‌توانند جابه‌جا شوند. این جابه‌جایی مختص پردازنده‌های گرافیکی گسسته²⁴ است که فضای حافظه‌ی میزبان و دستگاه کاملاً جدا از هم هستند [10]. در این پردازنده‌ها قبل از اجرای برنامه روی پردازنده‌ی گرافیکی، داده‌ها باید از حافظه‌ی میزبان به حافظه‌ی اصلی دستگاه منتقل شوند و در انتهای اجرای برنامه نیز نتایج در جهت معکوس از حافظه‌ی دستگاه به حافظه‌ی میزبان کپی شوند. در مقابل، پردازنده‌های گرافیکی یکپارچه²⁵ هستند که در آن‌ها فضای حافظه‌ی میزبان و دستگاه مشترک هستند و نیازی به در و بدل کردن داده بین طرفین وجود

¹⁷ Shortest Job First

¹⁸ Shortest Remining Time

¹⁹ Global Memory

²⁰ L2 Cache

²¹ Streaming Multi-processor

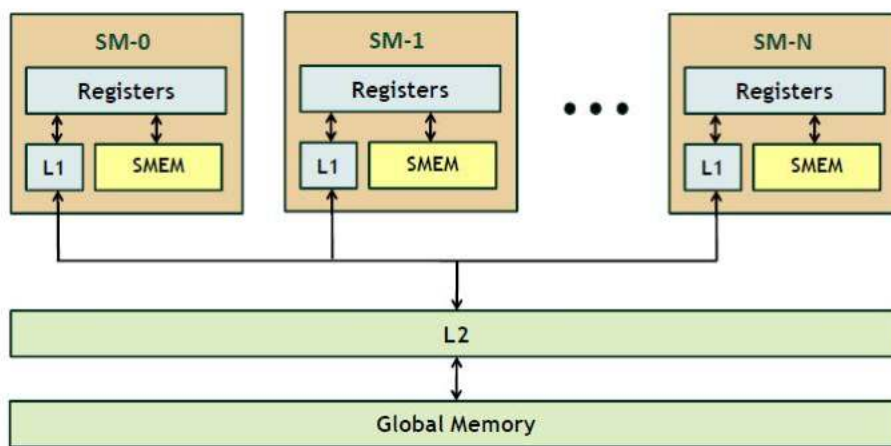
²² Host

²³ Device

²⁴ Descrete

²⁵ Integrated

ندارد [10]. پردازنده‌های گرافیکی نصب شده بر روی کامپیوترهای خانگی همگی از نوع گسسته هستند. در این تحقیق ما روی برنامه‌ای که در پردازنده‌ی گرافیکی اجرا می‌کنیم متمرکز می‌شویم و رد و بدل کردن داده‌ها در مدل پیشنهادی این تحقیق نقشی ندارد.



شکل 2 نمایش اجزای درونی یک پردازنده‌ی گرافیکی

SM ها کاملا مشابه هستند و هسته‌های پردازشی^{۲۶} در آن‌ها دسته‌بندی شده‌اند. این هسته‌ها بصورت SIMD^{۲۷} بکار گرفته شده‌اند و نخ‌ها را به صورت موازی داده اجرا می‌کنند. علاوه‌براین، در یک SM واحد زمانبند وارپ^{۲۸} برای عملیات زمانبندی وجود دارد که زمانبندی سطح دو در پردازنده‌ی گرافیکی محسوب می‌شود. واحد محاسبات دقت مضاعف (DPU^{۲۹})، واحد تابعی خاص (SFU^{۳۰})، واحد محاسبات یادگیری عمیق (TPU^{۳۱}) و واحدهای LD/ST^{۳۲} برای بارگذاری و ذخیره داده‌ها وجود دارند [11]. در داخل SM ها نیز یک حافظه‌ی اشتراکی^{۳۳} برای همکاری و رد و بدل کردن داده‌های نخ‌هایی که روی یک SM مشترک اجرا می‌شوند وجود دارد. این حافظه بسیار سریع است و از اهمیت بسیار زیادی برخوردار است. حافظه‌های داخل SM را حافظه‌های بر تراشه^{۳۴} و حافظه‌ی خارج از آن را حافظه‌ها خارج تراشه^{۳۵} می‌نامیم.

²⁶ Cores
²⁷ Single Instruction Multiple Data
²⁸ Warp Scheduler
²⁹ Double Precision Unit
³⁰ Special Function Unit
³¹ Tensor Processing Unit
³² Load/Store
³³ Shared Memory
³⁴ On-chip
³⁵ Off-chip

▪ مدل برنامه‌نویسی کودا:

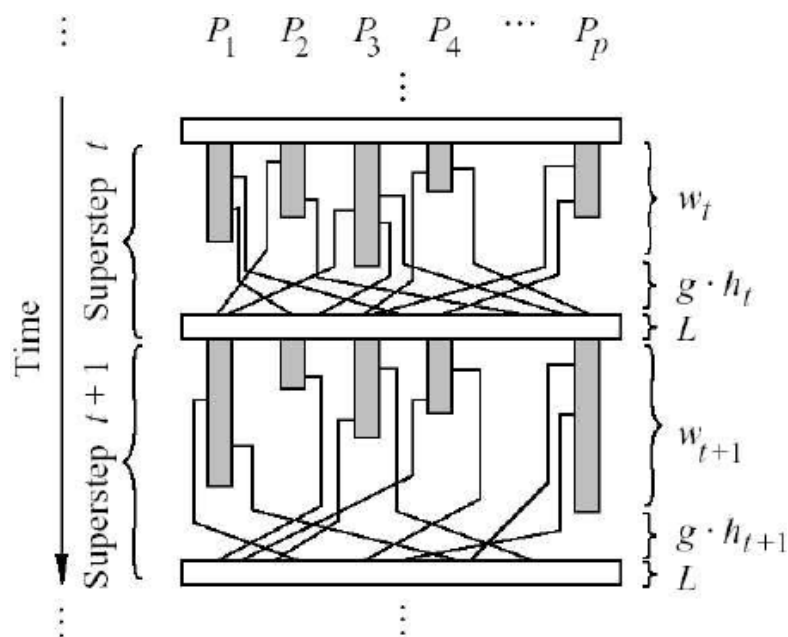
یک برنامه‌ی کودا شامل دو بخش میزبان و دستگاه است. برنامه‌ی بخش میزبان کرنل نام دارد و تابعی است که روی پردازنده‌ی گرافیکی اجرا می‌شود. برنامه‌ی کودا با اجرای برنامه‌ی میزبان (که روی پردازنده‌ی مرکزی اجرا می‌شود) شروع می‌شود و در ادامه کرنل فراخوانی می‌شود. داده‌هایی که باید کرنل روی آن‌ها اجرا شوند و آرگومان‌های کرنل محسوب می‌شوند، نیز قبل از فراخوانی کرنل در سمت حافظه‌ی سراسری دستگاه کپی می‌شوند. سپس کرنل فراخوانی و اجرا می‌شود و در انتها نتایج به سمت میزبان برگردانده می‌شوند. این چهار مرحله همیشه در برنامه‌های کودا که روی پردازنده‌های گرافیکی گسسته اجرا می‌شوند، برقرار است. در شکل 3 کد کودا برای مسئله جمع برداری نشان داده شده است.

```
__global__ void addKernel(int *c, const int *a, const int *b){
    int i = threadIdx.x;
    c[i] = a[i] + b[i];
}
void main(){
    const int arraySize = 5;
    const int a[arraySize] = { 1, 2, 3, 4, 5 };
    const int b[arraySize] = { 10, 20, 30, 40, 50 };
    int c[arraySize] = { 0 };
    int *dev_a = 0, *dev_b = 0, *dev_c = 0;
    cudaMalloc((void**)&dev_c, arraySize * sizeof(int));
    cudaMalloc((void**)&dev_a, arraySize * sizeof(int));
    cudaMalloc((void**)&dev_b, arraySize * sizeof(int));
    cudaMemcpy(dev_a, a, arraySize * sizeof(int), cudaMemcpyHostToDevice);
    cudaMemcpy(dev_b, b, arraySize * sizeof(int), cudaMemcpyHostToDevice);
    addKernel << 1, arraySize >> >(dev_c, dev_a, dev_b);
    cudaDeviceSynchronize();
    cudaMemcpy(c, dev_c, arraySize * sizeof(int), cudaMemcpyDeviceToHost);
    printf("{1,2,3,4,5} + {10,20,30,40,50} = {%d,%d,%d,%d,%d}\n", c[0], c[1], c[2], c[3], c[4]);
    cudaError_t cudaStatus = cudaGetLastError();
    if (cudaStatus != cudaSuccess)printf("cudaDeviceReset failed!");
}
```

شکل 3 کد کودا برای مسئله جمع برداری.

در کد جمع برداری ابتدا متغیرهای a ، b و c روی حافظه‌ی میزبان و متغیرهای dev_a ، dev_b و dev_c روی حافظه‌ی دستگاه تعریف می‌شوند. سپس آرگومان‌های کرنل (متغیرهای a و b) توسط تابع `cudaMemcpy` به سمت حافظه‌ی دستگاه کپی می‌شوند. در ادامه کرنل فراخوانی می‌شود و سپس مجدداً با استفاده از دستور `cudaMemcpy` نتایج که در آرایه dev_c ذخیره شده است، برگردانده می‌شود. در انتهای این برنامه دستور `cudaGetLastError` فراخوانی می‌شود که اگر در سمت دستگاه خطایی رخ داده است، برنامه‌ی سمت میزبان مطلع شود.

تابع `addKernel` کرنلی است که عملیات جمع برداری را انجام می‌دهد. همانطور که در شکل 3 نشان داده شده است، این تابع با توصیف‌گر³⁶ `__global__` در ابتدای آن مشخص شده است. با توجه بیشتر به این کد مشخص می‌شود که این تابع کاملاً ترتیبی است و هیچ اثری از اجرای موازی دیده نمی‌شود. در واقع در هنگام فراخوانی این تابع کرنل مشخص می‌شود که از این کد ترتیبی چند نخ ایجاد شود. این ویژگی توسط پیکربندی کرنل تعیین می‌شود و نقش مهمی در موازی‌سازی دارد. پیکربندی در کد شکل 3 به صورت `<<<1,arraySize>>>` تعیین شده است. این نشان می‌دهد که این کرنل شامل یک بلوک است که در هر بلوک به تعداد `arraySize` نخ وجود دارد. نخ‌ها به دسته‌های مساوی به نام بلوک تقسیم‌بندی شده‌اند. تعداد کل بلوک‌ها را اندازه‌ی گرید و تعداد نخ‌ها در هر بلوک را اندازه‌ی بلوک می‌نامیم. تعداد نخ‌های کل کرنل از ضرب اندازه‌ی گرید در اندازه‌ی بلوک به دست می‌آید. نکته‌ی بسیار مهم این است که نخ‌های هر بلوک الزاماً روی یک SM اجرا می‌شوند. البته روی یک SM چند بلوک نیز می‌توانند اجرا شوند. نخ‌های یک بلوک به دسته‌های 32 تایی تقسیم می‌شوند که وارپ³⁷ نامیده می‌شود. در عمل 32 نخ موجود در یک وارپ همزمان با هم اجرا می‌شوند.



شکل 4 Superstep های مدل BSP.

▪ مدل محاسباتی BSP:

BSP یک مدل محاسباتی برای مدل‌سازی محاسبات در سیستم‌های توزیع شده است. این مدل یک انتزاعی از یک کامپیوتر در یک سیستم توزیع شده ارائه می‌کند. این مدل پلی بین دیدگاه برنامه‌نویس و پیاده‌سازی

³⁶ Qualifier

³⁷ Warp

فیزیکی ماشین موازی ارائه می‌کند. مزیت مدل BSP این است که ساده و غیرمبهم است و ویژگی‌های فیزیکی سیستم توزیع شده را با استفاده از پارامترهای عددی بیان می‌کند. انتزاعی که این مدل ارائه می‌کند باعث می‌شود که برنامه‌نویس با جزئیات معماری سیستم درگیر نشود.

مدل BSP به صورت دنباله‌ای از Superstep ها در نظر گرفته شده است. شکل 4 این دنباله را نشان داده است. در هر Superstep ابتدا هر ماشین محاسبات محلی خود را انجام می‌دهد. سپس ارتباطات بین ماشین‌ها از طریق شبکه‌ی ارتباطی انجام می‌شود. سپس عملیات همگام‌سازی انجام می‌شود. فرض بر این است که یک تابع اولیه^{۳۸} برای عملیات همگام‌سازی در سرتاسر ماشین‌های موازی وجود دارد. بنابراین هزینه‌ی هر Superstep شامل هزینه‌ی محاسبات، ارتباطات و همگام‌سازی است و از طریق روابط زیر به دست می‌آید. کل هزینه‌ی برنامه‌ی موازی مجموع هزینه‌ی Superstep ها است. پارامترهای مدل در جدول 1 معرفی شده‌اند.

$$\text{Cost of a superstep} = \max_{i=1}^p (w_i) + \max_{i=1}^p (h_i \cdot g) + l$$

$$W + H \times g + S \times l = \sum_{s=1}^S w_s + g \times \sum_{s=1}^S h_s + S \times l$$

جدول 1 پارامترهای مدل BSP

پارامتر	توضیحات
p	تعداد پردازنده‌های موازی
s	تعداد دستورات محاسباتی که یک پردازنده در واحد زمان اجرا می‌کند.
l	تعداد قدم‌های عملیات همگام‌سازی
g	تعداد قدم‌های انتقال یک کلمه داده بین دو کامپیوتر موازی
w _i	هزینه‌ی محاسبات در ماشین موازی i ام
h _i	تعداد پیام‌های ارسال یا دریافت شده توسط فرایند i ام

▪ پیشینه‌ی تحقیق:

در این بخش پژوهش‌هایی که در حوزه‌ی مدل پیشنهادی انجام شده‌اند، مورد بررسی قرار می‌گیرند. این پژوهش‌ها را به دو دسته تقسیم می‌کنیم.

دسته‌ی اول پژوهش‌هایی هستند که یک مسئله‌ی دنیای واقعی که به روش برنامه‌نویسی پویا حل شده‌اند را در بستر کودا موازی‌سازی کرده‌اند. این دسته از پژوهش‌ها و برنامه‌های کودا که در آن‌ها طراحی شده‌اند، می‌توانند

³⁸ Primitive

به عنوان یک محک برای ارزیابی مدل پیشنهادی مورد استفاده قرار گیرند. از آنجا که محوریت الگوریتم‌های برنامه‌نویسی پویا براساس آرایه‌ها بنا شده است و از طرفی آرایه‌ها به موازی‌سازی داده که اساس موازی‌سازی در پردازنده‌ی گرافیکی است، کمک شایانی می‌کنند، به همین دلیل راه‌حل‌های زیادی روی پردازنده‌ی گرافیکی در این دسته از مسائل ارائه شده‌اند. یاماشیتا و ایتو [12] برای مسئله مثلث‌سازی چندضلعی بهینه راه‌حلی براساس پردازش موازی روی پردازنده‌ی گرافیکی ارائه کرده‌اند. آن‌ها تمام تلاش خود را برای یکپارچه^{۳۹} شدن دسترسی‌های حافظه‌ی سراسری انجام داده‌اند. آن‌ها دو پیاده‌سازی برای این مسئله روی پردازنده‌ی گرافیکی ارائه کرده‌اند که به ترتیب به تسریع 106.05 و 26.78 نسبت به برنامه‌های چندهسته‌ای با 8 نخ شده‌اند. محمد و همکاران در [13,14,15] پیاده‌سازی‌های برای مسئله تطبیق صوت با استفاده از پردازنده‌ی گرافیکی انجام داده‌اند. راه حل آنها براساس الگوریتم‌های برنامه‌نویسی پویاست و هدف آنها تسریع الگوریتم بوده است. در برخی از این پژوهش‌ها سعی شده است که با بکارگیری برخی از تکنیک‌های کارآیی این الگوریتم‌ها بهبود یابد. مارسین و پاول [16] از حافظه‌ی یکپارچه‌ی کودا و همچنین از تکنیک پیش‌واکشی داده‌ها را برای بهبود عملکرد استفاده کرده‌اند. همچنین نیشیدا و همکاران [17] نیز با تغییر مکان داده‌ها در حافظه عملکرد الگوریتم برنامه‌نویسی پویا روی پردازنده‌ی گرافیکی را بهبود دادند. از این نمونه مسائل بسیار زیاد هستند و می‌توانیم به [18-22] اشاره کنیم. باتوجه به اینکه این مسائل موضوع بحث این تحقیق نیستند، بنابراین در اینجا از پرداختن به جزئیات بیشتر پرهیز می‌کنیم.

دسته‌ی دوم پژوهش‌هایی هستند که روش برنامه‌نویسی پویا را فقط برای یک مسئله‌ی خاص مدل‌سازی کرده‌اند. این دسته از پژوهش‌ها اهمیت بیشتری در این تحقیق دارند. اما این دسته از پژوهش‌ها نیز مدل‌سازی تک منظوره انجام داده‌اند و مدلی ایجاد نشده است که تمامی الگوریتم‌های برنامه‌نویسی پویا را شامل شود. از این دسته از پژوهش‌ها می‌توان به [23-27] اشاره کرد. موضوع این تحقیق هیچکدام از این دو دسته نیستند. ما می‌خواهیم مدلی ایجاد کنیم که از طریق آن بتوانیم زمان اجرای هر الگوریتم برنامه‌نویسی پویا را پیش‌بینی کنیم.

در برخی از تحقیقات مدل BSP روی پردازنده‌ی گرافیکی به کار گرفته شده است. در [28] برمبنای مدل BSP یک مدل تحلیلی برای تخمین زمان اجرای کرنل کودا ارائه شده است. این مدل منحصراً برای الگوریتم‌های برنامه‌نویسی پویا نیست و تمام برنامه‌های کودا را شامل می‌شود. اما نکته‌ی بسیار مهم این است که در این پژوهش مدل کارآیی براساس پارامترهایی ایجاد شده‌اند که در اثر پروفایلینگ برنامه قابل محاسبه است و

³⁹ CoaScsed

نویسندگان به تحلیل ایستای کدها وفادار نبوده‌اند. این یک نقطه ضعف بسیار بزرگ در این پژوهش است. چون اساس پیش‌بینی زمان اجرا را تحت الشعاع قرار می‌دهد. اگر برای محاسبات مدل نیازمند اجرای برنامه باشیم، طبیعتاً پیش‌بینی کارآیی کارکرد خود را از دست می‌دهد. نویسندگان همچنین علاوه بر ارائه یک مدل تحلیلی، چند روش یادگیری ماشین را نیز برای پیش‌بینی زمان اجرای برنامه‌های کودا استفاده کرده‌اند. نتایج آن‌ها نشان می‌دهد که روش‌های یادگیری ماشین عملکرد بهتری نسبت به مدل تحلیلی ارائه کرده‌اند.

و - جنبه جدید بودن و نوآوری در تحقیق:

مهم‌ترین نوآوری این تحقیق ایجاد یک مدل کارآیی برای برنامه‌های مبتنی بر الگوریتم‌های برنامه‌نویسی پویا روی پردازنده‌ی گرافیکی بر مبنای مدل BSP است. طبق بررسی‌هایی که در پیشینه‌ی مدل‌های کارآیی انجام شده است، تاکنون مدلی با این ویژگی‌ها ایجاد نشده است.

ز - اهداف مشخص تحقیق (شامل اهداف آرمانی، کلی، اهداف ویژه و کاربردی):

اهداف تحقیق عبارتند از:

- ایجاد یک مدل کارآیی برای برنامه‌های کودا مبتنی بر تکنیک برنامه‌نویسی پویا و بکارگیری مدل محاسباتی BSP برای توصیف عملکرد دستگاه.
- ارائه روابط ریاضی برای تخمین کارآیی یک برنامه‌ی کودا مبتنی بر تکنیک برنامه‌نویسی پویا
- بیان اجزای تاثیرگذار در پردازنده‌ی گرافیکی به صورت پارامترهای قابل درک و قابل اندازه‌گیری از طریق عملیات ریزمحک‌زنی.
- آزمون مدل به لحاظ دقت پیش‌بینی با استفاده از محک‌های دنیای واقعی

ح - در صورت داشتن هدف کاربردی، نام بهره‌وران (سازمان‌ها، صنایع و یا گروه ذینفعان) ذکر شود (به عبارت دیگر محل اجرای مطالعه موردی):

ط - سؤالات تحقیق:

سؤالات تحقیق عبارتند از:

- آیا می‌توان مدل محاسباتی BSP را برای توصیف کارآیی برنامه‌های کودا و مدل‌سازی عملکرد آن‌ها بکار گرفت؟

- آیا می‌توان از این مدل کارآیی برای تخمین زمان اجرای برنامه‌های مبتنی بر الگوریتم‌های برنامه-نویسی پویا استفاده کرد؟

ی- فرضیه‌های تحقیق:

فرضیه‌های تحقیق عبارتند از:

- با توجه به تطبیقی که پارامترهای کارآیی مدل BSP با مراحل یک الگوریتم برنامه‌نویسی پویا دارند، فرضیه‌ی ما این است که می‌توان این مدل را برای ایجاد مدل کارآیی روی پردازنده‌ی گرافیکی بکار گرفت.

- با توجه به تطبیقی که بیان شد، فرضیه‌ی ما این است که می‌توانیم زمان اجرای این برنامه‌ها را با دقت قابل قبول پیش‌بینی کرد.

ک- تعریف واژه‌ها و اصطلاحات فنی و تخصصی (به صورت مفهومی و عملیاتی):

کرنل: تابعی است که روی پردازنده‌ی گرافیکی اجرا می‌شود.

پیکربندی کرنل: تعیین ابعاد گرید و ابعاد بلوک را پیکربندی کرنل گوئیم.

حافظه‌ی سراسری: تنه‌ای حافظه‌ی ای است که از سمت میزبان می‌توان داده رد و بدل شود.

حافظه‌ی اشتراکی: حافظه‌ای در درون SM ها است که تاخیر دسترسی آن در حد حافظه‌ی نهان است و برنامه-نویس می‌تواند روی آن متغیر تعریف کند. ویژگی مهم آن این است که نخ‌های یک بلوک می‌توانند داده‌هایی را روی این حافظه به اشتراک گذارند و برای همکاری از آن استفاده کنند.

مدل محاسباتی: یک مدل محاسباتی سعی می‌کند ابعاد و جنبه‌هایی از یک پدیده (سخت‌افزاری یا نرم‌افزاری) را به صورت پارامترهای عددی بیان کند و به ساده‌سازی آن پدیده کمک می‌کند. مدل محاسباتی ویژگی‌هایی را بیان می‌کند که در محدوده‌ی مسئله است و روی هدف و عملکرد مدل متمرکز می‌شود. به این ترتیب مواردی که

مرتبط با دامنه‌ی مسئله نیستند را از پدیده حذف می‌کند. با استفاده از مدل‌های محاسباتی عملکرد پدیده‌ها به صورت ریاضی بین می‌شوند و امکانات خوبی برای اعتبارسنجی^{۴۰} ایجاد می‌کنند.

مدل برنامه‌نویسی کودا: مهم‌ترین مدل برنامه‌نویسی برای پردازنده‌های گرافیکی است که از سال 2007 توسط شرکت انویدیا ارائه شده است. این مدل برنامه‌نویسی فقط روی پردازنده‌های شرکت انویدیا قابل بکارگیری است و از این جهت قابلیت حمل^{۴۱} پایین‌تری نسبت زبان‌هایی مثل OpenCL دارد. اما در عوض، برنامه‌های نوشته شده با استفاده از کودا زمان اجرای بهتری دارند.

مدل کارآیی: مدلی است که یک جنبه از کارآیی یک دستگاه را نشان می‌دهد. کارآیی می‌تواند زمان اجرای برنامه‌ها، میزان توان و انرژی مصرفی، دمای پردازنده و ... باشد. در این تحقیق منظور ما از کارآیی، زمان اجرای برنامه‌ی ورودی است.

ریزمحک‌زنی: طراحی و اجرای یک برنامه‌ی کوچک برای استخراج یک یا چند ویژگی یا پارامتر از یک دستگاه را ریزمحک‌زنی گوییم.

تنظیم برنامه‌ی کودا: بکارگیری هر تکنیکی که باعث بهبود کارآیی یک برنامه شود را تنظیم نامیم. در برخی مستندات به جای این کلمه از کلمه‌ی بهینه‌سازی استفاده می‌شود که خیلی مناسب نیست.

کدهای PTX: یک روایت شبه اسمبلی از برنامه‌ی کودا است. در واقع برنامه‌ی کودا از طریق کامپایلر NVCC ابتدا به کدهای PTX و سپس توسط کامپایلر JIT^{۴۲} به کدهای باینری تبدیل می‌شود.

همگام‌سازی: در پردازش موازی گاهی اوقات لازم است که نخ‌ها در یک نقطه‌ی به‌خصوص همگام شوند. به عبارت دیگر نقطه‌ای به نام سد^{۴۳} تعریف می‌شود که همه‌ی نخ‌هایی که زودتر به این نقطه رسیدند، صبر می‌کنند تا سایر نخ‌ها هم به این سد برسند و سپس برنامه ادامه پیدا کند.

الگوریتم برنامه‌نویسی پویا: یکی از روش‌های طراحی الگوریتم است که محاسبات در آن برمبنای آرایه‌ها بنا شده است. به این ترتیب که محاسبات از مقادیر جزئی شروع شده و نتایج هر مرحله در آرایه ذخیره می‌شود. سپس در مراحل بعدی از این نتایج استفاده می‌شود. این رویکرد یک روش پایین به بالاست.

⁴⁰ Verification

⁴¹ Portability

⁴² Just-in-time

⁴³ Barrier

5- روش تحقیق:

الف- شرح کامل روش تحقیق بر حسب هدف، نوع داده ها و نحوه اجراء (شامل مواد، تجهیزات و استانداردهای مورد استفاده در قالب مراحل اجرایی تحقیق به تفکیک):

تذکر: درخصوص تفکیک مراحل اجرایی تحقیق و توضیح آن، از به کار بردن عناوین کلی نظیر، «گردآوری اطلاعات اولیه»، «تهیه نمونه‌های آزمون»، «انجام آزمایش‌ها» و غیره خودداری شده و لازم است در هر مورد توضیحات کامل در رابطه با منابع و مراکز تهیه داده‌ها و ملزومات، نوع فعالیت، مواد، روش‌ها، استانداردها، تجهیزات و مشخصات هر یک ارائه گردد.

همانطور که در بخش‌های قبلی بیان شد، مدل پیشنهادی در این تحقیق براساس مدل پایه‌ی BSP بنا خواهد شد. مدل BSP در بخش پیشینه‌ی تحقیق بیان شده است. در این بخش ابتدا تطبیق یک برنامه که براساس الگوریتم برنامه‌نویسی پویا ایجاد شده است، با مراحل و جزئیات مدل BSP بیان می‌شود و سپس نحوه‌ی ایجاد مدل تشریح خواهد شد.

در بسیاری از الگوریتم‌های برنامه‌نویسی پویا محاسبات به صورت تکراری انجام می‌شود. در هر تکرار یک عنصر از آرایه محاسبه می‌شود و سپس این مقدار محاسبه شده در آرایه ذخیره می‌شود تا در محاسبات تکرار بعدی استفاده شوند. معمولاً داده‌های محاسبه شده در هر تکرار در تکرار مرحله بعدی استفاده می‌شوند. بنابراین نخ-های موازی باید صبر کنند تا محاسبات مرحله‌ی قبلی خاتمه یافته و نتایج در آرایه ذخیره شود. بنابراین در انتهای هر تکرار همگام‌سازی بین نخ‌های یک بلوک باید انجام شود. مثال بسیار خوب برای مسائل این چنینی الگوریتم ضرب زنجیره‌ای ماتریس‌هاست که ترتیب محاسبات آن به صورت قطری است و در شکل 5 نشان داده شده است. در این الگوریتم در محاسبه‌ی یک عنصر از قطر k ام، به همه عناصر در قطرهای $1, 2, \dots, k-1$ بستگی دارد. بنابراین نخ‌ی که این عنصر را محاسبه می‌کند، باید از خاتمه محاسبه در قطر قبلی اطمینان یابد. بنابراین به وضوح برای محاسبات نیازمند همگام‌سازی هستیم. بنابراین هر تکرار از این الگوریتم (که در آن یک قطر به صورت موازی توسط نخ‌های مختلف محاسبه می‌شود) ابتدا محاسبات، سپس ذخیره نتایج در آرایه (ارتباطات) و در آنها همگام‌سازی دارد.

	قطر 1	قطر 2	قطر 3	قطر 4	قطر 5	قطر 7
1	0	?	?	?	?	?
2	?	0	?	?	?	?
3	?	?	0	?	?	?
4	?	?	?	0	?	?
5	?	?	?	?	0	?
6	?	?	?	?	?	0

شکل 4 نمایش محاسبات قطری در الگوریتم‌های برنامه‌نویسی پویا.

ما هر تکرار از این الگوریتم را با یک Superstep از مدل BSP معادل قرار می‌دهیم. حداکثر تعداد نخ‌های درگیر در محاسبات (که قطر اصلی را محاسبه می‌کنند) که در اینجا برابر اندازه‌ی قطر 1 است همان ابعاد ماتریس (مثلاً پارامتر n) است می‌باشد. ما این پارامتر را با p (تعداد پردازنده‌ها در BSP) معادل قرار می‌دهیم. هزینه‌ی یک ارتباطات در مدل BSP را با هزینه‌ی تاخیر نوشتن در حافظه‌ی سراسری در پردازنده‌ی گرافیکی معادل قرار می‌دهیم. هزینه‌ی محاسبات و همگام‌سازی نیز واضح هستند. همانطور که در مدل BSP هزینه‌ی کل از طریق مجموع هزینه‌های Superstep ها به دست می‌آید، در این مدل نیز هزینه‌ی کل از مجموع هزینه‌های همه‌ی تکرارها به دست می‌آید.

ب- متغیرهای مورد بررسی در قالب یک مدل مفهومی و شرح چگونگی بررسی و اندازه‌گیری متغیرها:

برخی از پارامترهای مدل پیشنهادی به شرح جدول 2 هستند:

پارامتر	توصیف پارامتر
g_{L1}	زمان دسترسی به حافظه‌ی نهان سطح یک
g_{L2}	زمان دسترسی به حافظه‌ی نهان سطح دو
g_{GM}	زمان دسترسی به حافظه‌ی سراسری
g_{SHMEM}	زمان دسترسی به حافظه‌ی اشتراکی
#SM	تعداد SM ها در پردازنده‌ی گرافیکی
#core	تعداد هسته‌های پردازشی در هر SM
L	کل هزینه‌ی همگام‌سازی
COMP	کل هزینه‌ی محاسبات
COMM	کل هزینه‌ی ارتباطات

ج - شرح کامل روش (میدانی، کتابخانه‌ای) و ابزار (مشاهده و آزمون، پرسشنامه، مصاحبه، فیش‌برداری و غیره) گردآوری داده‌ها:

یک برنامه‌ی کودا یا PTX دنباله‌ای از دستورالعمل‌هاست که این دستورالعمل‌ها به سه دسته‌ی دستورالعمل‌های محاسباتی، ارتباطی و همگام‌سازی دسته‌بندی می‌شوند. تاخیر هریک از این دستورالعمل‌ها با استفاده از ریزمحک‌زنی محاسبه می‌شوند. این عملیات ریزمحک‌زنی در سطح کدهای PTX انجام می‌شوند و دقت بسیار خوبی دارند. بنابراین با استفاده از تحلیل همزمان کدهای کودا و PTX تعداد Superstep های موجود در الگوریتم برنامه‌نویسی پویا و همچنین اندازه‌ی محاسبات، ارتباطات و همگام‌سازی در هر Superstep را محاسبه می‌کنیم. بنابراین هزینه‌ی کل برنامه محاسبه می‌شود.

ریزمحک‌زنی ابزار مهمی در ایجاد مدل‌های تحلیلی است. در مدل پیشنهادی این تحقیق پارامترهای زیادی وجود دارند که باید از ریزمحک‌زنی برای محاسبه‌ی آن‌ها استفاده کنیم. از مصادیق مهم‌ترین پارامترها محاسبه‌ی تاخیر سلسله‌مراتب سطوح مختلف حافظه است. برای این منظور ریزمحک‌هایی طراحی خواهد شد. به عنوان مثال برای محاسبه‌ی زمان تاخیر حافظه‌های نهان سطح یک و سطح دو دسترسی‌هایی به اندیس‌های متوالی از یک آرایه انجام می‌دهیم و این برنامه را یکبار در حالت خاموشی حافظه‌ی نهان سطح یک و یکبار با روشن کردن آن تاخیر دسترسی را محاسبه می‌کنیم. در زمان روشنی این حافظه تاخیر g_{L1} و در زمان خاموشی آن g_{L2} محاسبه می‌شوند.

د - جامعه آماری، روش نمونه‌گیری و حجم نمونه (در صورت وجود و امکان):

برای آزمون میزان دقت مدل پیشنهادی در تخمین زمان اجرای برنامه‌های ورودی از محک‌های دنیای واقعی مثل NAS، Rodinia و مجموعه مثال‌های کودا استفاده می‌کنیم. از بین این محک‌ها مواردی که از الگوریتم‌های برنامه‌نویسی پویا استفاده کرده‌اند را برای آزمون مدل پیشنهادی استفاده خواهیم کرد. همچنین الگوریتم‌های مهم در روش برنامه‌نویسی پویا مثل ضرب زنجیره‌ای ماتریس‌ها و الگوریتم یافتن درخت دودویی بهینه استفاده خواهیم کرد. دقت مدل پیشنهادی با استفاده از رابطه‌ی زیر به دست می‌آید. در این معادله Acc دقت پیش‌بینی، t_{real} زمان واقعی کرنل کوداست که حاصل اجرای آن است و $t_{predict}$ زمان پیش‌بینی شده توسط مدل پیشنهاد است.

$$Acc \% = \frac{|t_{real} - t_{predict}|}{t_{real}} \times 100$$

ه - روش‌ها و ابزار تجزیه و تحلیل داده‌ها:

ابزارهای ما برای انجام آزمایشات کامپیوترهای شخصی مجهز به کارت‌های گرافیک سری GeForce محصول شرکت انویدیا است.

و-منابع استفاده شده:

- [1] Pandey, Mohit, et al. "The transformational role of GPU computing and deep learning in drug discovery." *Nature Machine Intelligence* 4.3 (2022): 211-221.
- [2] NVIDIA, C. U. D. A., "CUDA C Programming Guide, Version 11.8.", NVIDIA Corporation, 2022.
- [3] Schoonhoven, Richard, Ben van Werkhoven, and K. Joost Batenburg. "Benchmarking optimization algorithms for auto-tuning GPU kernels." *IEEE Transactions on Evolutionary Computation* (2022).
- [4] Lattuada, Marco, et al. "Performance prediction of deep learning applications training in GPU as a service systems." *Cluster Computing* (2022): 1-24.
- [5] Valiant, Leslie G., "A bridging model for parallel computation.", *Communications of the ACM*, 33.8, 103-111, 1990.
- [6] Valiant, Leslie G. "A bridging model for multi-core computing.", *Journal of Computer and System Sciences* 77.1, 154-166, 2011.
- [7] Amaris, Marcos, et al., "A simple bsp-based model to predict execution time in gpu applications.", 2015 IEEE 22nd International Conference on High Performance Computing (HiPC), IEEE, 2015.
- [8] NVIDIA, C. U. D. A., " Parallel Thread Execution ISA, Version 8.0.", NVIDIA Corporation, 2022.
- [9] NVIDIA, "Whitepaper NVIDIA GeForce GTX 1080 Gaming Perfected", NVIDIA Corp, 2016
- [10] Buttinger-Kreuzhuber, Andreas, et al. "An integrated GPU-accelerated modeling framework for high-resolution simulations of rural and urban flash floods." *Environmental Modelling & Software* 156 (2022): 105480.
- [11] NVIDIA, "NVIDIA Turing GPU architecture: Graphics reinvented", NVIDIA Corp, 2018.
- [12] Yamashita, Kohei, Yasuaki Ito, and Koji Nakano. "Bulk execution of the dynamic programming for the optimal polygon triangulation problem on the GPU." *Concurrency and Computation: Practice and Experience* 31.19 (2019): e4947.
- [13] Hallek, Mohamed, et al. "Color Weighted Rank Transform and Improved Dynamic Programming for Fast and Accurate Stereo Matching." Available at SSRN 4045830.
- [14] Hallek, Mohamed, Fethi Smach, and Mohamed Atri. "Real-time stereo matching on CUDA using Fourier descriptors and dynamic programming." *Computational Visual Media* 5.1 (2019): 59-71.
- [15] Hallek, Mohamed, et al. "Dynamic programming with adaptive and self-adjusting penalty for real-time accurate stereo matching." *Journal of Real-Time Image Processing* 19.2 (2022): 233-245.
- [16] Knap, Marcin, and Paweł Czarnul. "Performance evaluation of unified memory with prefetching and oversubscription for selected parallel cuda applications on nvidia pascal and volta gpus." *The Journal of Supercomputing* 75.11 (2019): 7625-7645.

- [17] Nishida, Kazufumi, Yasuaki Ito, and Koji Nakano. "Accelerating the dynamic programming for the matrix chain product on the GPU." 2011 Second International Conference on Networking and Computing. IEEE, 2011.
- [18] Fichte, Johannes K., Markus Hecher, and Valentin Roland. "Parallel model counting with cuda: Algorithm engineering for efficient hardware utilization." 27th International Conference on Principles and Practice of Constraint Programming (CP 2021). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- [19] Zhu, Zhaoxuan, et al. "A GPU implementation of a look-ahead optimal controller for eco-driving based on dynamic programming." 2021 European Control Conference (ECC). IEEE, 2021.
- [20] Diwan, Tausif, and Jitendra Tembhurne. "A Parallelization of Non-Serial Polyadic Dynamic Programming on GPU." Journal of computing and information technology 27.2 (2019): 55-66.
- [21] Kool, Wouter, et al. "Deep policy dynamic programming for vehicle routing problems." Integration of Constraint Programming, Artificial Intelligence, and Operations Research: 19th International Conference, CPAIOR 2022, Los Angeles, CA, USA, June 20-23, 2022, Proceedings. Cham: Springer International Publishing, 2022.
- [22] Sadiq, Muhammad Umair, and Muhammad Murtaza Yousaf. "Space-efficient computation of parallel approximate string matching." The Journal of Supercomputing (2023): 1-34.
- [23] Alavani, Gargi, Kajal Varma, and Santonu Sarkar. "Predicting execution time of CUDA kernel using static analysis." 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom). IEEE, 2018.
- [24] Guo, Ping, Liqiang Wang, and Po Chen. "A performance modeling and optimization analysis tool for sparse matrix-vector multiplication on GPUs." IEEE Transactions on Parallel and Distributed Systems 25.5 (2013): 1112-1123.
- [25] Amaris, Marcos, et al. "Evaluating execution time predictions on GPU kernels using an analytical model and machine learning techniques." Journal of Parallel and Distributed Computing 171 (2023): 66-78.
- [26] NVIDIA, C. U. D. A., "CUDA C Programming Guide, Version 11.8.", NVIDIA Corporation, 2022.
- [27] Schoonhoven, Richard, Ben van Werkhoven, and K. Joost Batenburg. "Benchmarking optimization algorithms for auto-tuning GPU kernels." IEEE Transactions on Evolutionary Computation (2022).

6- استفاده از امکانات آزمایشگاهی واحد:

- آیا برای انجام تحقیقات نیاز به استفاده از امکانات آزمایشگاهی واحد علوم و تحقیقات می‌باشد؟ بلی

خیر

در صورت نیاز به امکانات آزمایشگاهی لازم است نوع آزمایشگاه، تجهیزات، مواد و وسایل مورد نیاز در این قسمت مشخص گردد.

نوع آزمایشگاه	تجهیزات مورد نیاز	مواد و وسایل	مقدار مورد نیاز

- آیا برای انجام تحقیقات نیاز به حمایت از سایر مراکز خارج از واحد علوم و تحقیقات می باشید؟

بلی خیر

در صورت نیاز نام مراکز و نحوه حمایت (مالی، امکانات و تجهیزات و ..) مشخص گردد.

امضاء مدیر گروه تخصصی:

امضاء استاد راهنما:

7- زمان بندی انجام تحقیق:

الف- تاریخ شروع:..... ب- مدت زمان انجام تحقیق:..... ج- تاریخ اتمام:.....
تذکر: لازم است کلیه فعالیتها و مراحل اجرایی تحقیق (شامل زمان ارائه گزارشات دوره‌ای) و مدت زمان مورد نیاز برای هر یک، به تفکیک پیش‌بینی و در جدول مربوطه درج گردیده و در هنگام انجام عملی تحقیق، حتی‌الامکان رعایت گردد.

پیش‌بینی زمان‌بندی فعالیت‌ها و مراحل اجرایی تحقیق و ارائه گزارش پیشرفت کار

زمان اجرا به ماه																								زمان کل (ماه)	شرح فعالیت			
24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1					
																										1		
																											2	
																											3	
																											4	
																											5	
																											6	
																											7	
																											8	
																											9	
																											10	
																											11	
																											12	

توجه: 1- زمان و نوع فعالیت‌های اجرایی رساله، حتی‌الامکان باید با مندرجات جدول منطبق باشد.

2- حداقل زمان قابل قبول برای پیش‌بینی مراحل مطالعاتی و اجرایی رساله دکتری 12 ماه و حداکثر 24 ماه می‌باشد.

تذکره: اساتید راهنما و مشاور موظف هستند قبل از پذیرش پروپوزال، به سقف ظرفیت راهنمایی و مشاوره خود توجه نموده و در صورت تکمیل بودن ظرفیت پذیرش، از امضاء این فرم و یا در نوبت قرار دادن آن و ایجاد وقفه در کار دانشجویان جداً پرهیز نمایند. بدیهی است در صورت عدم رعایت موازین مربوطه، مسئولیت تأخیر در ارائه پروپوزال و عواقب کار، متوجه گروه تخصصی خواهد بود.

8- صورتجلسه گروه تخصصی

نام و نام خانوادگی دانشجو: _____
امضاء _____ تاریخ _____

نام و نام خانوادگی استاد یا استادان راهنما
1- (عضو هیأت علمی دانشگاه)
2- (عضو هیأت علمی دانشگاه)
امضاء _____ تاریخ _____

نام و نام خانوادگی استاد یا استادان مشاور
1- (عضو هیأت علمی دانشگاه)
2- (عضو هیأت علمی دانشگاه)
امضاء _____ تاریخ _____

نام و نام خانوادگی اعضاء کمیته نظارت بر تحقیق
1- (عضو هیأت علمی دانشگاه)
امضاء _____ تاریخ _____

شورای گروه تخصصی در تاریخ در محل با حضور اعضای مربوطه

تشکیل و موضوع رساله _____ خانم
آقای _____ با عنوان _____
.....

بررسی و به تصویب رسید.

نام و نام خانوادگی اعضای شورا
1- _____
2- _____
3- _____
4- _____
5- _____
6- _____
امضاء _____ تاریخ _____

نام و نام خانوادگی مدیر گروه: _____
امضاء _____ تاریخ _____

تذکره: لازم است پروپوزال دانشجویان از تاریخ تأیید در شورای گروه تخصصی تا زمان طرح در شورای پژوهشی دانشکده بیشتر از یکماه نگذرد.

تذکره: لازم است قبل از تصویب پروپوزال در شورای پژوهشی دانشکده، شرایط احراز و ظرفیت پذیرش اساتید راهنما و مشاور مطابق بخشنامه‌های مربوطه توسط پژوهش دانشکده کنترل شود.

9- صورتجلسه شورای (پژوهشی) دانشکده:

خانم

موضوع و طرح تحقیق رساله آقای

گرایش..... که به تصویب کمیته گروه تخصصی مربوطه رسیده است، در جلسه مورخ شورای (پژوهشی) دانشکده مطرح شد و پس از بحث و تبادل نظر مورد تصویب اکثریت اعضاء قرار گرفت.

ردیف	نام و نام خانوادگی	نوع رأی (موافق یا مخالف)	محل امضاء	توضیحات
1				
2				
3				
4				
5				
6				
7				

نام و نام خانوادگی مدیر/کارشناس پژوهشی دانشکده:

امضاء تاریخ

نام و نام خانوادگی ریاست دانشکده

امضاء تاریخ

این فرم باید توسط دانشجو تکمیل شود

فرم سازمان مرکزی دانشگاه آزاد اسلامی

فرم الف - فرم اطلاعات رساله دکترای تخصصی (Ph.D)

نام واحد دانشگاهی: واحد علوم و تحقیقات	
عنوان رساله دکتری:	
نام و نام خانوادگی دانشجو:	نیمسال تحصیلی:
شماره دانشجویی:	تعداد واحد پایان نامه:
رشته تحصیلی:	گرایش:
<input type="checkbox"/> فنی و مهندسی	<input type="checkbox"/> علوم انسانی
<input type="checkbox"/> کشاورزی	<input type="checkbox"/> هنر
رشته تحصیلی:	<input type="checkbox"/> علوم پایه
نام و نام خانوادگی استاد راهنما 1:	
مرتبه علمی: استادیار <input type="checkbox"/> دانشیار <input type="checkbox"/> استاد <input type="checkbox"/>	رشته تحصیلی:
کد شناسایی استاد راهنما:	رشته تحصیلی:
نام و نام خانوادگی استاد راهنما 2:	
مرتبه علمی: استادیار <input type="checkbox"/> دانشیار <input type="checkbox"/> استاد <input type="checkbox"/>	رشته تحصیلی:
کد شناسایی استاد راهنما:	رشته تحصیلی:
نام و نام خانوادگی استاد مشاور 1:	
مرتبه علمی: استادیار <input type="checkbox"/> دانشیار <input type="checkbox"/> استاد <input type="checkbox"/>	رشته تحصیلی:
کد شناسایی استاد راهنما:	رشته تحصیلی:
نام و نام خانوادگی استاد مشاور 2:	
مرتبه علمی: استادیار <input type="checkbox"/> دانشیار <input type="checkbox"/> استاد <input type="checkbox"/>	رشته تحصیلی:
کد شناسایی استاد راهنما:	رشته تحصیلی:

فرم شماره 1

دستاوردهای حاصل از نتایج حاصل از پایان نامه کارشناسی ارشد یا رساله دکتری

تمامی تولیدات علمی مستخرج از پایان نامه/رساله (در قالب کتاب، مقاله، طرح تحقیقاتی، اختراع، اکتشاف و ...) با رعایت موارد ذیل قابل انتشار است.

الف) نویسنده اول مقاله باید به نام دانشجو و به عنوان تنها آدرس دانشگاه آزاد اسلامی واحد علوم و تحقیقات باشد.

ب) عهده دار مکاتبات (Corresponding Author) می تواند استاد راهنما با آدرس مؤسسه محل خدمت خود یا دانشجو با نشانی دانشگاه آزاد اسلامی واحد علوم و تحقیقات باشد.

ج) فقط اسامی افراد مرتبط با پایان نامه یا رساله (دانشجو -استاد راهنما-استاد مشاور) در مستخرجات ذکر شود.

د) آدرس دقیق دانشگاه و واحد و رعایت ترتیب و توالی آن باید به صورت زیر آورده شود:

به انگلیسی:

Department of تخصصی , Science and Research branch, Islamic Azad University, Tehran, Iran.

به فارسی:

دانشگاه آزاد اسلامی، واحد علوم و تحقیقات، گروه، تهران، ایران.

اینجانبان اساتید راهنما و مشاور پایان نامه / رساله، خانم/آقای، دانشجوی مقطع

..... رشته گرایش..... با عنوان

.....»

«.....»

موارد فوق را می پذیریم

نام و نام خانوادگی استاد راهنما:

تاریخ و امضاء

نام و نام خانوادگی استاد راهنما:

تاریخ و امضاء

نام و نام خانوادگی استاد مشاور:

تاریخ و امضاء

نام و نام خانوادگی استاد مشاور:

تاریخ و امضاء

در تاریخ فرم مزبور که به امضاء اساتید محترم راهنما و مشاور رسیده است، دریافت گردید.

امضاء مدیر پژوهش دانشکده

تذکره: لازم است اساتید راهنما و مشاور انتخابی پس از مطالعه فرم مذکور، نسبت به تکمیل مشخصات و

توشیح آن شخصاً اقدام نمایند تا از هرگونه تخلفات احتمالی جلوگیری گردد.

دستورالعمل نحوه تدوین و استفاده از نتایج حاصل از پایان نامه کارشناسی ارشد یا رساله دکتری

الف) کلیه مطالب و مندرجات پایان نامه/ رساله بر اساس اصول علمی و حاصل از تحقیقات خودم تهیه شود و در صورت استفاده از مطالب، نتایج تحقیقات، نقل قولها، جداول و نمودارهای دیگران در پایان نامه/رساله، منابع و ماخذ آن به نحوی که قابل تشخیص و تفکیک از متن اصلی باشد قید گردد.

ب) در صورتیکه از نتایج تحقیقاتم علاوه بر پایان نامه / رساله، کتاب، مقاله، اختراع، اکتشاف و هر گونه تولیدات علمی حاصل شود، صرفاً بنام دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران بوده و این موضوع صراحتاً در مکاتبات و تولیدات اینجانب درج و بر اساس ضوابط دانشگاه اقدام نمایم.

ج) در صورت استفاده از کمکهای مالی و غیر مالی نهادهای دولتی و غیر دولتی از موضوع تحقیق اینجانب مراتب را کتباً به دانشکده اطلاع دهم در غیر اینصورت دانشکده مجاز به تغییر عنوان پایان نامه یا سایر اقدامات حقوقی می باشد.

د) آدرس دقیق دانشگاه و واحد و رعایت ترتیب و توالی آن به عنوان تنها آدرس در تولیدات علمی مستخرج از پایان نامه/رساله باید به صورت زیر آورده شود:

آدرس دانشگاه و واحد به فارسی: دانشگاه آزاد اسلامی، واحد علوم و تحقیقات تهران، گروه، تهران، ایران.
آدرس دانشگاه و واحد به انگلیسی:

Department of تخصصی , Science and Research branch, Islamic Azad University,
Tehran, Iran.

اینجانب دانشجوی ورودی مقطع رشته
..... گرایش که موضوع پایان نامه/رساله ام تحت عنوان:
.....»
«.....»

در شورای گروه تخصصی مطرح و به تصویب رسیده است موارد فوق را مطالعه کرده و پذیرفتم

نام و نام خانوادگی دانشجو

تاریخ و امضاء

تذکر: بدیهی است چنانچه تحت هر شرایطی و در هر زمان، دانشگاه خلاف موارد ذکر شده را مشاهده نماید نسبت به تصمیم اتخاذ شده هیچگونه ادعایی نداشته و حق هرگونه اعتراضی را از خود سلب و ساقط می نمایم.

در تاریخ فرم مزبور که توسط آقای / خانم به امضاء رسیده است، دریافت گردید.

امضاء مدیر پژوهش دانشکده

فرم شماره 3

فرم تأیید استعلام کتابخانه مرکزی و پژوهشگاه علوم و فناوری اطلاعات ایران
(ایران داک)

اینجانبان اساتید راهنما پایان نامه / رساله آقای / خانم دانشجوی مقطع
دکتری تخصصی / کارشناسی ارشد / دکتری حرفه‌ای رشته با
عنوان استعلام‌های اخذ شده کتابخانه مرکزی
و پژوهشگاه علوم و فناوری اطلاعات ایران (ایران داک) مبنی بر عناوین مشابه عنوان فوق را مطالعه نموده و
با عنایت به کلمات کلیدی عنوان پروپوزال در سایت‌های www.irandoc.ac.ir و sika.iau.ir
تکراری نبودن عنوان پروپوزال مذکور مور تأیید می‌باشد.

نام و نام خانوادگی استاد راهنمای اول:

نام و نام خانوادگی استاد راهنمای دوم: