
بسم الله الرحمن الرحيم

دانشگاه یزد

دانشکده‌ی ریاضی

گروه علوم کامپیوتر

پایان‌نامه

پایان‌نامه جهت دریافت درجه کارشناسی ارشد

علوم کامپیوتر

بررسی الگوریتم‌های هندسی موازی برای کامپیوترهای چندهسته‌ای

استاد راهنما:

دکتر محمد فرشی

استادان مشاور:

دکتر سید ابوالفضل شاهزاده‌فاضلی

دکتر فضل ا... ادیب نیا

پژوهش‌گر:

فاطمه دهقانی فیروزآبادی

اسفند ۱۳۹۰

تقدیم به پدرم

که در پناهش، نه از سرمای زمستان، بر من گزندی رسید؛ و نه از گرمای آتشین تابستان...

تقدیم به مادرم

به پاس تمام نیایش‌های شبانه‌اش، مهربی‌دریغش و عشق بی‌نهایتش

تقدیم به همسر عزیزم

برای تمام صبر، اندیشه و ایمانی که نثارم کرد...

سپاس‌گزاری

سپاس خداوند را که مرا به راه دانش رهنمون شد و هموست که فرصت شاگردی اساتید گرامی را بر من ارزانی داشت تا چراغی بر تاریکی جهلم باشند.

در آغاز وظیفه خود می‌دانم از زحمات بی‌دریغ استاد راهنمای خود، جناب آقای دکتر محمد فرشی، صمیمانه تشکر و قدردانی کنم که قطعاً بدون راهنمایی‌های ارزنده ایشان، این مجموعه به انجام نمی‌رسید. سعه‌ی صدر، اخلاق و نکته‌سنجی این بزرگوار را سرلوحه خود در تمام مراحل زندگی قرار داده و از خدای منان برای ایشان، توفیق و سربلندی روزافزون را خواستارم.

از همراهی و راهنمایی‌های استادان فرهیخته، جناب آقای دکتر سید ابوالفضل شاهزاده‌فاضلی و جناب آقای دکتر فضل ا... ادیب‌نیا که زحمت مطالعه و مشاوره این پایان‌نامه را تقبل فرمودند و در آماده‌سازی این رساله، به نحو احسن اینجانب را مورد راهنمایی قرار دادند، کمال سپاس و امتنان را دارم.

چکیده

غشای محدب یک مجموعه‌ای از نقاط، کوچکترین مجموعه محدبی است که همه‌ی نقاط را شامل می‌شود. غشای محدب یک ساختار اولیه در ریاضیات و هندسه محاسباتی است و در مسائلی مانند تشخیص الگو، شکل‌شناسی و پردازش تصویر کاربرد فراوانی دارد.

در این پایان‌نامه، یک الگوریتم موازی مقیاس‌پذیر برای ساخت غشای محدب مجموعه‌ای از n نقطه در صفحه بررسی می‌گردد. این الگوریتم برای مدل چندکامپیوتری دانه‌درشت طراحی شده است که در این مدل، p پردازنده هر یک با حافظه محلی $O\left(\frac{n}{p}\right) \gg O(1)$ ، توسط شبکه‌های ارتباطی دلخواه به یکدیگر متصل می‌شوند. این الگوریتم برای دامنه وسیعی از مقادیر n و p مقیاس‌پذیر است یعنی برای مقادیر $\frac{n}{p} \geq p^{\epsilon}$، $(\epsilon > 0)$ کارا و قابل اجراست. زمان مورد اجرای این الگوریتم برابر $O\left(\frac{T_{sequential}}{p} + T_s(n,p)\right)$ است که $T_{sequential}$ زمان اجرای بهترین الگوریتم ترتیبی و $T_s(n,p)$ زمان مرتب‌سازی کلی n داده بر روی یک ماشین p پردازنده‌ای است. علاوه بر این، الگوریتم تنها از تعداد ثابتی دورهای ارتباطی کلی استفاده می‌کند.

فهرست مطالب

لیست تصاویر

لیست الگوریتم‌ها

لیست رویه‌ها

پیشگفتار

با گسترش روزافزون استفاده از کامپیوتر برای کارهای تحقیقاتی در دانشگاه‌ها، مراکز تحقیقاتی و شرکت‌های تجاری، نیاز به پردازش سریع‌تر افزایش یافته و به یک نیاز اساسی تبدیل شده است. امروزه پردازش موازی نقش بسیار جدی در مرتفع‌سازی این نیاز ایفا می‌کند.

سرعت کامپیوترهای شخصی کنونی نسبت به اجداد خود به طور سرسام‌آوری افزایش یافته است اما حتی این سرعت نجومی نیز در اجرای برخی از برنامه‌های پیشرفته، کند است. از جمله عرصه‌هایی که احتیاج به کامپیوترهایی با سرعت پردازش بسیار بالا دارند می‌توان به برنامه‌های شبیه‌سازی در تحقیقات هسته‌ای، فناوری نانو، برنامه‌های پیش‌بینی وضعیت هوا، برنامه‌های فیلم‌سازی کامپیوتری، برنامه‌های ساخت انیمیشن حرفه‌ای و بسیاری از زمینه‌های مختلف دیگر را نام برد که همگی به سرعت پردازش بسیار زیاد نیاز دارند تا در یک زمان مناسب به نتیجه برسند.

یک راه حل برای این معضل، استفاده از سوپرکامپیوترها است. درست است که سرعت پردازش سوپرکامپیوترها بسیار بالاتر از کامپیوترهای شخصی است اما استفاده از آنها در در همه موارد مقرون به صرفه نیست؛ ضمن آنکه این فناوری در انحصار بعضی از کشورهای توسعه‌یافته است و سایر کشورها از دسترسی به این تجهیزات استراتژیک محروم هستند.

راه حل دیگر در دستیابی به سرعت پردازش بسیار بالا استفاده از روش پردازش موازی است. به بیان ساده در این روش چند پردازنده معمولی با همکاری یکدیگر به اجرای یک برنامه می‌پردازند که طی این همکاری، برنامه با سرعت بالاتری اجرا می‌شود.

برای اینکه بتوان از امکانات و تجهیزات یک سیستم موازی به درستی استفاده نمود، نیاز به سیستم عامل‌هایی است که موازی سازی را پشتیبانی کنند، همچنین برای حل مسائلی که قابلیت موازی‌سازی را دارند، لازم است که الگوریتم‌ها و برنامه‌هایی موازی ارائه گردند. اولین گام در برنامه‌نویسی موازی این است که طراح برنامه،

بتواند به صورت موازی فکر کند، مسئله را به درستی شناخته و بر روی قسمت‌هایی از مسئله که قابلیت اجرای همزمان دارند، تمرکز کند. در این پایان‌نامه سعی شده است تا ملزومات و پیش‌نیازهای برنامه‌نویسی موازی معرفی شود و سپس به عنوان نمونه، برای حل یک مسئله هندسی، الگوریتم موازی ارائه می‌گردد.

در فصل اول مقدمه‌ای بر مفاهیم برنامه‌نویسی موازی بیان می‌شود. در فصل دوم مدل موازی مورد بحث این تحقیق معرفی می‌گردد و در فصل سوم، ابتدا مسئله هندسی محاسبه غشای محدب معرفی شده و سپس برای حل این مسئله هندسی، الگوریتمی موازی ارائه و تحلیل می‌شود.

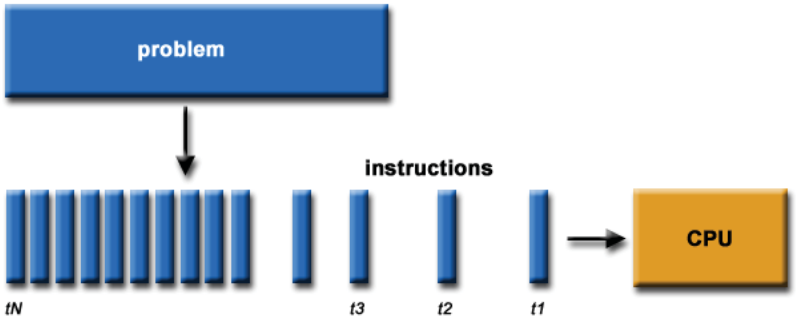
فصل ۱

مقدمه‌ای بر برنامه‌نویسی موازی

این فصل مروری اجمالی بر مفهوم پردازش موازی، اهمیت و کاربردهای آن و همچنین مروری بر مفاهیم و ملزومات برنامه‌نویسی موازی خواهد بود. اکثر مفاهیمی که در ادامه فصل آورده شده از مراجع [۸، ۲۰، ۲۳] استخراج گردیده است. در انتهای فصل نیز، تکنولوژی چندهسته‌ای که در چند سال اخیر پیشرفت چشمگیری داشته است، معرفی می‌گردد.

۱.۱ محاسبه موازی

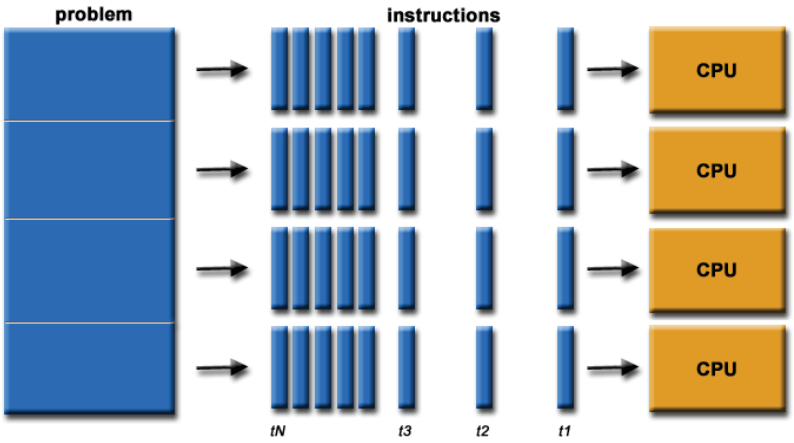
محاسبه موازی به معنای اجرای همزمان قسمت‌های مختلف یک برنامه در چند پردازنده به منظور حصول سریع‌تر نتایج است. در پردازش ترتیبی، دستورات به ترتیب در پردازنده اجرا می‌شوند و سرعت اجرا متناسب با سرعت پردازنده است (شکل ۱.۱). در پردازش موازی دستورات در چند پردازنده اجرا می‌شوند ولی سرعت اجرا الزاماً برابر با تعداد پردازنده‌ها ضربدر سرعت یک پردازنده نیست (شکل ۲.۱). محاسبه موازی در بخش‌های مختلف کامپیوتر اعم از سخت‌افزار و نرم‌افزار شکل می‌گیرد، بنابراین برای بررسی کلیات محاسبه موازی باید به جنبه‌های مختلف سخت‌افزاری و نرم‌افزاری آن پرداخت.



شکل ۱.۱: محاسبات سریال

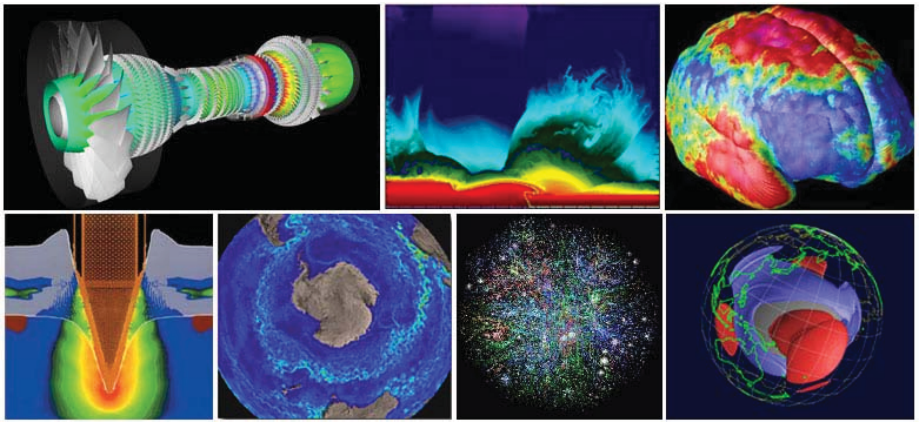
از پردازش موازی در جهت افزایش قدرت کامپیوترها استفاده می‌شود. اما اصلی‌ترین استفاده از آن در حل مسائل و مدل‌های علمی و مهندسی است (شکل ۳.۱). از جمله این حوزه‌ها می‌توان به موارد زیر اشاره کرد:

- فیزیک کاربردی، هسته‌ای، ذرات بنیادی، ماده چگال، گداخت هسته‌ای، فوتونیک و نانو
- اتمسفر، زمین و محیط زیست
- فناوری زیستی و ژنتیک
- زمین‌شناسی و زلزله‌شناسی



شکل ۲.۱: محاسبات موازی و کاربرد آن

- مهندسی مکانیک؛ از اندام مصنوعی تا مصنوعات فضایی
- مهندسی الکترونیک؛ طراحی مدار، میکروالکترونیک
- علوم کامپیوتر و ریاضی



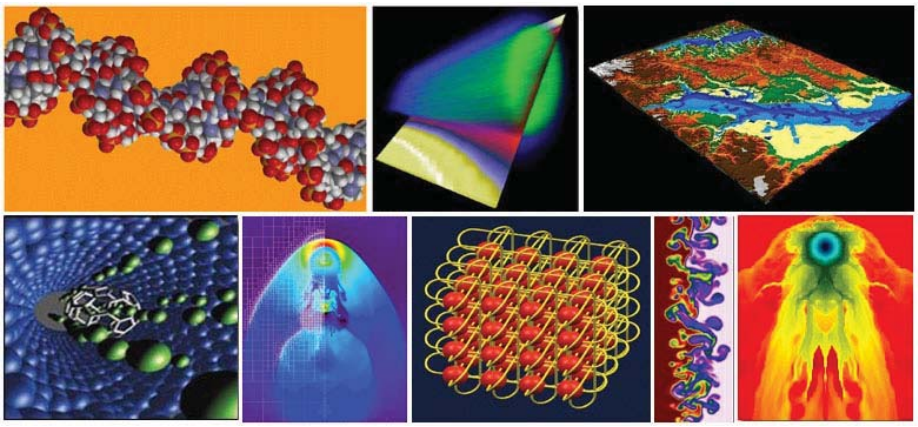
شکل ۳.۱: تصاویری از داده‌های محاسباتی در رشته‌های مختلف

اما امروزه نه تنها حل مسایل علمی احتیاج به پردازش موازی دارد بلکه برخی از نرم‌افزارهای تجاری نیز به کامپیوترهای سریع نیاز دارند. بسیاری از این برنامه‌ها احتیاج به پردازش حجم زیادی از داده به شکل بسیار پیچیده دارند. از جمله این برنامه‌ها می‌توان به موارد زیر اشاره کرد:

- پایگاه‌های عظیم داده و عملیات داده کاوی[۱]
- اکتشاف نفت

[۱] Data Mining

- موتورهای جستجوی وب، سرویس‌های تجاری تحت وب
- تصویربرداری و تشخیص پزشکی (شکل ۴.۱)
- طراحی و شبیه‌سازی دارو
- مدیریت شرکت‌های ملی و چند ملیتی
- مدل‌سازی مالی و اقتصادی
- گرافیک پیشرفته و فناوری واقعیت مجازی[۲] [۹]
- فناوری چند رسانه‌ای و شبکه ویدیویی



شکل ۴.۱: تصاویر ایجاد شده توسط نرم‌افزارهای تجاری

دلایل اصلی استفاده از محاسبات موازی عبارتند از:

صرفه‌جویی در زمان و هزینه: استفاده از منابع بیشتر در یک کار، زمان انجام کار را کاهش می‌دهد. علاوه بر این، استفاده از چندین منبع ارزان به جای استفاده از یک منبع گران‌قیمت سبب کاهش هزینه‌ها می‌گردد.

حل مسائل بزرگتر: بسیاری از مسائل بزرگ و پیچیده که حل آن‌ها توسط یک کامپیوتر، خصوصاً در کامپیوترهای با حافظه محدود، غیرعملی یا غیرممکن است، با استفاده از کامپیوترهای موازی قابل حل می‌باشند. به عنوان مثال در موتورهای جستجوگر وب و پایگاه داده‌ها لازم است که میلیون‌ها تراکنش در یک ثانیه پردازش شوند.

[۲] Virtual Reality Technology

فراهم نمودن همزمانی: در منابع محاسباتی منفرد، در یک زمان تنها یک کار را می‌توان انجام داد، در حالی که در منابع محاسباتی چندگانه، چندین کار به طور همزمان قابل انجام است. به عنوان مثال AccessGrid[۳] یک شبکه همکاری جهانی است که مردم از سراسر جهان می‌توانند به طور همزمان و به صورت مجازی با یکدیگر ملاقات نموده و مراودات کاری خود را انجام دهند.

استفاده از منابع غیرمحلی: در مواقعی که منابع محاسباتی محلی برای حل مسئله مورد نظر، محدود هستند می‌توان به‌واسطه شبکه‌های گسترده و یا اینترنت از منابع غیرمحلی، برای حل مسئله کمک گرفت. به عنوان مثال در Folding@home[۴] که یک پروژه محاسبات توزیع‌شده است، غالب بر ۳۴۰۰۰۰ کامپیوتر متصل به اینترنت به کار گرفته می‌شود تا در مسئله تشخیص چین‌خوردگی‌های پروتئین، توان محاسباتی ۴.۲ ترافلاپس[۵] حاصل گردد (هر ترافلاپس معادل یک تریلیون عملیات ممیز شناور در ثانیه است).

۲.۱ معماری کامپیوتر موازی

پرطرفدارترین طبقه‌بندی معماری سیستم‌های کامپیوتری در سال ۱۹۶۶ توسط فلین[۶] تعریف شد. طرح طبقه‌بندی فلین[۷] براساس جریان اطلاعات صورت گرفته است. اطلاعاتی که پردازنده با آن سر و کار دارد را می‌توان به دو دسته دستورالعمل و داده تقسیم نمود. طبق طبقه‌بندی فلین جریان دستورالعمل یا داده می‌تواند به یکی از صورت‌های تکی یا چندتایی باشد. بر این اساس معماری سیستم‌های کامپیوتری را می‌توان به چهار دسته تقسیم نمود:

۱. جریان یک دستورالعمل، یک داده (SISD): این معماری برای یک کامپیوتر ترتیبی (غیر موازی) است. در این روش تنها یک جریان دستوری توسط یک پردازنده در طول هر پالس ساعت مورد عمل واقع می‌شود و همین‌طور یک جریان داده در طول پالس ساعت مورد استفاده قرار می‌گیرد. در این روش دستورها به طور قطعی انجام می‌شوند و وابسته به عمل پردازنده دیگر نیستند. کامپیوترهای ترتیبی متعارف

---

[۳] www.accessgrid.org

[۴] folding.stanford.edu

[۵] TeraFLOPS

[۶] Flynn

[۷] Flynn's Taxonomy