



پردازش موازی

برنامه نویسی با GPU

نویسندگان

دکتر اسدالله شادبهرامی
(هیأت علمی مهندسی کامپیوتر دانشگاه گرج)

صدیقه جم

ناشر: کتب مهندسی، کامپیوتر، دانشگاهی، علمی-تخصصی

نص

سرشناسه	: شاه بهرامی، اسدالله، ... ۱۳۴۷-
عنوان و نام پدیدآور	: پردازش موازی: برنامه نویسی با GPU/نویسندگان اسدالله شاه بهرامی، صدیقه جم.
مشخصات نشر	: تهران: نشر نص، ۱۳۹۶.
مشخصات ظاهری	: ۴۲۸ ص.؛ مصور، جدول، نمودار.
شابک	: 978-964-410-383-4 ریال ۲۸۰۰۰۰
وضعیت فهرست نویسی	: فیا
یادداشت	: واژنامه.
موضوع	: برنامه نویسی با GPU.
موضوع	: parallel processing (Electronic computers)
موضوع	: پردازش موازی
موضوع	: parallel processing (Computer science)
شناسه افزوده	: جم، صدیقه، ۱۳۶۶-
رده بندی کنگره	: ۲۱۳۹۶ ش ۴ پ AQ۶۷/۸۵
رده بندی دیویی	: ۰۰۴/۳۵
شماره کتابشناسی ملی	: ۴۸۵۸۳۶۷

این کتاب در تاریخ ۱۳۹۶/۳/۱۰ در کمیته نشر دانشگاه گیلان تصویب گردیده است



موسسه علمی فرهنگی

پردازش موازی، اصول برنامه نویسی با GPU
 اسدالله شاه بهرامی، صدیقه جم
 چاپ اول: پاییز ۹۶
 تیراژ: ۲۰۰۰
 ناشر: «نص»
 طراح، آماده سازی، چاپ و صحافی: موسسه علمی فرهنگی «نص»
 قیمت: ۲۸۰۰۰ تومان

دفتر انتشارات: تهران، میدان انقلاب، خ منیری جاوید، بن بست مبین، شماره ۶

تلفن: ۶۶۴۱۲۳۸۵ - ۶۶۴۶۵۶۷۴ - ۶۶۹۵۳۸۸۳

فروشگاه: ضلع جنوب شرقی میدان انقلاب، شماره ۲۵، تلفن: ۶۶۴۰۵۳۷۲

ISBN: 978-964-410-383-4

شابک: ۹۷۸-۹۶۴-۴۱۰-۳۸۳-۴

ایمیل: info@nasspub.om

وب سایت: www.nass.ir



فهرست

پیشگفتار ۱۵

بخش اول: مفاهیم، موازی، ازی، اصول و تکنیک‌ها

فصل اول : توسعه سیستم‌های کامپیوتری	۱۹
۱-۱ روند تکامل سخت‌افزار	۲۰
۱-۱-۱ پیدایش ماشین‌های محاسباتی ساده	۲۰
۲-۱-۱ طراحی ماشین‌های محاسباتی توسط دانشمندان	۲۰
۳-۱-۱ کارت‌های پانچ	۲۱
۴-۱-۱ دستگاه تقاضای و تحلیلی اعداد	۲۱
۵-۱-۱ پیشرفت در استفاده از کارت‌های پانچ	۲۲
۶-۱-۱ پیدایش شرکت IBM	۲۲
۲-۱ نسل‌های کامپیوتر	۲۳
۳-۱ روند تکامل ریزپردازنده‌های Intel	۲۳
۴-۱ تشخیص تعداد هسته‌های پردازنده یک کامپیوتر	۲۵
۵-۱ روند تکامل نرم‌افزار	۲۷
۶-۱ نتیجه‌گیری	۲۸

۲۸ تمرین‌ها.

فصل دوم: پردازش موازی

۱-۲ انگیزه‌ی موازی‌سازی

۳۳ ۱-۱-۲ توان محاسباتی

۳۳ ۲-۱-۲ سرعت حافظه/دیسک

۳۴ ۳-۱-۲ ارتباطات داده‌ای

۲-۲ مدل‌های پردازش

۳۴ ۱-۲-۲ پردازش سریال

۳۵ ۲-۲-۲ پردازش موازی

۳-۲ جمع‌بندی فلین

۳۶ ۱-۲-۲ معماری یک دستورالعمل، یک داده

۳۷ ۲-۳-۲ معماری یک دستورالعمل، چند داده

۳۸ ۳-۳-۲ معماری چند دستورالعمل، یک داده

۳۸ ۴-۳-۲ معماری چند دستورالعمل، چند داده

۴-۲ انواع مدل‌های پردازش موازی

۳۹ ۱-۴-۲ مدل ماشین

۳۹ ۲-۴-۲ مدل معماری

۴۰ ۳-۴-۲ مدل محاسباتی

۴۱ ۴-۴-۲ مدل برنامه‌نویسی

۵-۲ الگوریتم‌های موازی

۴۱ ۱-۵-۲ الگوریتم

۴۲ ۲-۵-۲ مدل‌های طراحی الگوریتم‌های موازی

۴۲ ۱-۲-۵-۲ مدل تولید فرایند یا نخ

۴۲ ۲-۲-۵-۲ مدل انشعاب-الحاق

۴۲ ۳-۲-۵-۲ مدل شروع پاراگراف-پایان پاراگراف

۴۳ ۴-۲-۵-۲ مدل یک برنامه، چند داده

۴۳ ۵-۲-۵-۲ مدل کارفرما-کارگر

۴۳ ۶-۲-۵-۲ مدل خدمتگذار-مشتري

۴۴ ۷-۲-۵-۲ مدل پردازش خط لوله

۴۵ ۸-۲-۵-۲ مدل مخزن وظیفه

۴۶ ۹-۲-۵-۲ مدل تولید کننده-مصرف کننده

۶-۲ دامنه‌ی محاسبات موازی

۴۶ ۱-۶-۲ برنامه‌های کاربردی در مهندسی و طراحی

۴۷	برنامه‌های کاربردی علمی ۲-۶-۲
۴۷	برنامه‌های کاربردی تجاری ۳-۶-۲
۴۷	برنامه‌های کاربردی در سیستم‌های کامپیوتری ۴-۶-۲
۴۸	نتیجه‌گیری ۷-۲
۴۸	تمرین‌ها

فصل سوم: نحوه موازی‌سازی الگوریتم‌ها ۴۹

۵۰	۱-۳ سطوح موازی‌سازی
۵۰	۱-۱-۳ موازی‌سازی در سطح دستورالعمل
۵۲	۲-۱-۳ موازی‌سازی در سطح داده
۵۲	۳-۱-۳ موازی‌سازی در سطح حلقه
۵۳	۴-۱-۳ موازی‌سازی در سطح تابع
۵۳	۲-۳ موازی‌سازی الگوریتم‌های سریال
۵۷	۳-۳ تکنیک‌های تجزیه
۵۷	۱-۳-۳ تجزیه با شش
۵۹	۲-۳-۳ تجزیه داده
۶۶	۴-۳ انتخاب وظایف و نکات آینه‌ای پدازنده‌ها
۶۹	۵-۳ تکنیک‌های بهینه‌سازی کامپایلر
۶۹	۱-۵-۳ ترکیب حلقه
۷۰	۲-۵-۳ تعویض حلقه
۷۰	۳-۵-۳ مورب‌سازی حلقه
۷۱	۴-۵-۳ کاشی‌کاری فضای تکرار
۷۲	۵-۵-۳ گشودن حلقه
۷۳	۶-۵-۳ آزادسازی حلقه
۷۳	۷-۵-۳ نرمال‌سازی حلقه
۷۳	۸-۵-۳ معکوس‌سازی حلقه
۷۴	۶-۳ نتیجه‌گیری
۷۴	تمرین‌ها

فصل چهارم: مدل‌های برنامه‌نویسی موازی ۷۷

۷۸	۱-۴ مدل یک دستورالعمل چند داده
۷۹	۲-۴ مدل برنامه‌نویسی OpenMP
۸۲	۳-۴ مدل برنامه‌نویسی MPI
۸۲	۱-۳-۴ رابط ارسال پیام (MPI)

۸۴	۲-۳-۴ مفاهیم اصلی MPI	۷۲
۸۴	۱-۲-۳-۴ حافظه توزیع شده	۷۲
۸۵	۲-۲-۳-۴ تبادل پیام	۷۷
۸۵	۳-۲-۳-۴ فرایندها	۸۱
۸۵	۴-۲-۳-۴ کتابخانه تبادل پیام	۸۵
۸۵	۵-۲-۳-۴ ارسال/دریافت	۸۵
۸۶	۶-۲-۲-۴ هم زمانی/ناهم زمانی	۸۹
۸۶	۷-۲-۳-۴ بافر برنامه کاربردی	۹۵
۸۶	۸-۲-۳-۴ بافر سیستم	۹۵
۸۶	۹-۳-۴ ارتباط مسدودکننده	۹۵
۸۶	۱۰-۲-۳-۴ ارتباطات غیر مسدودکننده	۹۵
۸۶	۱۱-۲-۳-۴ پارسا ها و گروه ها	۹۵
۸۶	۱۲-۲-۳-۴ رتبه	۹۵
۸۷	۴-۴ مدل برنامه نویسی OpenMP	۹۵
۸۸	۵-۴ مدل برنامه نویسی PCP/XT/Theta	۹۵
۸۹	۶-۴ مدل های برنامه نویسی موازی	۹۵
۹۰	۱-۶-۴ محاسبات همه منظوره بر روی GPU	۹۹
۹۰	۲-۶-۴ مدل برنامه نویسی GPU	۹۹
۹۲	۷-۴ مدل برنامه نویسی OpenCL	۹۹
۹۳	۱-۷-۴ مدل زیرساخت OpenCL	۹۹
۹۳	۲-۷-۴ مدل اجرای OpenCL	۹۹
۹۶	۳-۷-۴ مدل حافظه OpenCL	۹۹
۹۷	۴-۷-۴ مدل برنامه نویسی OpenCL	۹۹
۹۷	۱-۴-۷-۴ مدل برنامه نویسی موازی داده ای	۹۹
۹۸	۲-۴-۷-۴ مدل برنامه نویسی موازی وظیفه ای	۹۹
۹۸	۵-۷-۴ شباهت های CUDA و OpenCL	۹۹
۹۹	۶-۷-۴ مدل برنامه نویسی مبتنی بر دستور سطح بالا بر روی GPU	۹۹
۱۰۰	۸-۴ موازی سازی الگوریتم ضرب ماتریس در هر یک از مدل های برنامه نویسی موازی	۹۹
۱۰۰	۱-۸-۴ الگوریتم سریال ضرب ماتریس	۹۹
۱۰۲	۲-۸-۴ الگوریتم های موازی ضرب ماتریس	۹۹
۱۰۲	۱-۲-۸-۴ الگوریتم موازی مربوط به مدل برنامه نویسی SIMD	۹۹
۱۰۴	۲-۲-۸-۴ الگوریتم موازی مربوط به مدل برنامه نویسی SIMD+OpenMP	۹۹
۱۰۵	۳-۲-۸-۴ الگوریتم موازی مربوط به مدل برنامه نویسی MPI	۹۹

۱۱۱	۴-۲-۸-۴ الگوریتم موازی مربوط به مدل ناهمگن برنامه‌نویسی OpenCL	۱۹۱
۱۱۳	۹-۴ نتیجه‌گیری	۲۹۲
۱۱۴	تمرین‌ها	۲۹۱

بخش دوم: اصول GPU و برنامه‌نویسی بر روی آن

۱۱۷	فصل پنجم: معماری GPU	۱۹۱
۱۱۸	۱-۵ معماری برخی از کامپیوترهای کنونی	۱۹۱
۱۲۴	۲-۵ معماری کلی واحد پردازشگر گرافیکی	۲۹۲
۱۲۸	۳-۵ تاریخچه GPU	۲۹۱
۱۳۰	۴-۵ تفاوت CPU و GPU	۲۹۱
۱۳۴	۵-۵ نشان‌های مختلف GPU	۲۹۱
۱۳۵	۱-۵-۵ قابلیت محاسباتی ۱.x	۲۹۱
۱۳۶	۲-۵-۵ قابلیت محاسباتی ۲.x	۲۹۱
۱۴۰	۳-۵-۵ قابلیت محاسباتی ۳.x	۲۹۱
۱۴۱	۴-۵-۵ قابلیت محاسباتی ۴.x	۲۹۱
۱۴۳	۵-۵-۵ قابلیت محاسباتی ۵.x	۲۹۱
۱۴۷	۶-۵ کاربردهای GPU	۲۹۱
۱۴۷	۱-۶-۵ جلوه‌های ویژه یا افکت‌های نیمه‌هوا سینمایی و انیمیشن	۲۹۱
۱۴۸	۲-۶-۵ اقتصادی و مالی	۲۹۱
۱۴۸	۳-۶-۵ پزشکی	۲۹۱
۱۴۹	۴-۶-۵ صنایع دفاعی و دولتی	۲۹۱
۱۴۹	۵-۶-۵ امنیت	۲۹۱
۱۵۰	۶-۶-۵ سیستم‌های جغرافیایی	۲۹۱
۱۵۰	۷-۶-۵ نفت و گاز	۲۹۱
۱۵۱	۸-۶-۵ الکترومغناطیس	۲۹۱
۱۵۱	۷-۵ نتیجه‌گیری	۲۹۱
۱۵۱	تمرین‌ها	۲۹۱

فصل ششم: مفهوم موازی سازی از منظر GPU

۱۵۳	۱-۶ مسائل سریال/موازی	۲۹۱
۱۵۶	۲-۶ همروندی و ویژگی محلی بودن	۲۹۱
۱۵۶	۱-۲-۶ همروندی عملیات	۲۹۱
۱۵۸	۲-۲-۶ ویژگی محلی بودن	۲۹۱

۱۶۱	 CUDA و NVIDIA	۳-۶
۱۶۲	 سخت افزار GPU	۴-۶
۱۶۴	 انواع موازی سازی در GPU	۵-۶
۱۶۴	 موازی سازی مبتنی بر وظیفه	۱-۵-۶
۱۶۶	 موازی سازی مبتنی بر داده	۲-۵-۶
۱۶۸	 نتیجه گیری	۶-۶

فصل هفتم : راه اندازی CUDA

۱۷۲	 Visual Studio	۱-۷
۱۷۵	 نصب اشکال زدا	۲-۷
۱۷۹	 ال پی ال	۳-۷
۱۸۰	 نتیجه گیری	۴-۷

فصل هشتم : آشنایی با مفاهیم شبکه ها، بلاک ها و نخ ها

۱۸۲	 CUDA	۱-۸
۱۸۲	 تجزیه مسئله	۲-۸
۱۸۴	 نخ بندی در GPU	۳-۸
۱۸۶	 نگاهی به سخت افزار	۴-۸
۱۸۹	 کرنل های CUDA	۵-۸
۱۹۰	 بلاک ها	۶-۸
۱۹۲	 آرایش بلاک ها	۱-۶-۸
۱۹۷	 شبکه ها	۷-۸
۱۹۸	 گام و آفست	۱-۷-۸
۲۰۰	 اندیس های نخ X و Y	۲-۷-۸
۲۰۷	 تار	۸-۸
۲۰۸	 انشعاب (شرط)	۱-۸-۸
۲۰۹	 بهره برداری از GPU	۲-۸-۸
۲۱۰	 زمان بندی بلاک	
۲۱۲	 نتیجه گیری	
۲۱۳	 تمرین ها	

فصل نهم : مدیریت حافظه توسط CUDA

۲۱۷	 حافظه نهان	۱-۹
۲۱۹	 کاربرد رجیسترها	۲-۹

۲۲۹.....	حافظه‌ی اشتراکی	۳-۹	۵۶۲
۲۳۰.....	مرتب‌سازی با استفاده از حافظه‌ی اشتراکی	۱-۳-۹	۸۶۲
۲۳۴.....	مرتب‌سازی مبنایی	۲-۳-۹	۸۶۶
۲۴۰.....	ادغام لیست‌ها	۳-۳-۹	۶۶۲
۲۴۵.....	ادغام چندین بردار به روش موازی	۴-۳-۹	۱۰۶
۲۴۷.....	کاهش موازی	۵-۳-۹	۹۰۶
۲۵۰.....	حافظه‌ی اشتراکی در انواع GPU	۶-۳-۹	۹۱۶
۲۵۱.....	خلاصه حافظه اشتراکی	۷-۳-۹	۹۰۶
۲۵۱.....	حافظه ثابت	۴-۹	۱۱۶
۲۵۲.....	ویژگی ذخیره محتوا در حافظه نهان توسط حافظه ثابت	۱-۱-۹	۲۱۶
۲۵۲.....	دستگاه‌های با قابلیت محاسباتی ۱.x	۱-۲۰	۶۱۶
۲۵۳.....	دستگاه‌های با قابلیت محاسباتی ۲.x و بالاتر	۲-۱-۹	۶۱۶
۲۵۴.....	ویژگی از حافظه ثابت	۲-۴-۹	۶۱۶
۲۵۴.....	حافظه سری	۵-۹	۶۱۶
۲۶۳.....	رتبه‌بندی	۱-۵-۹	۶۱۶
۲۶۳.....	مرتب‌سازی حافظه سری	۲-۵-۹	۶۱۶
۲۶۶.....	مرتب‌سازی نمونه	۳-۵-۹	۶۱۶
۲۶۸.....	انتخاب نمونه‌ها	۴-۵-۹	۶۱۶
۲۷۰.....	مرتب‌سازی نمونه‌ها	۵-۵-۹	۶۱۶
۲۷۱.....	شمارش بخش‌های نمونه	۶-۵-۹	۶۱۶
۲۷۴.....	جمع پیشوندی	۷-۵-۹	۶۱۶
۲۷۹.....	دسته‌بندی به بخش‌ها	۸-۵-۹	۶۱۶
۲۸۱.....	مرتب‌سازی بخش‌ها	۹-۵-۹	۶۱۶
۲۸۴.....	تجزیه و تحلیل نتایج	۱۰-۵-۹	۶۱۶
۲۸۷.....	حافظه بافت	۶-۹	۶۱۶
۲۸۷.....	ذخیره‌سازی در حافظه نهان حافظه بافت	۱-۶-۹	۶۱۶
۲۸۸.....	دست‌کاری سخت‌افزار در فراخوانی‌های حافظه	۲-۶-۹	۶۱۶
۲۸۹.....	محدودیت‌های استفاده از حافظه بافت	۳-۶-۹	۶۱۶
۲۸۹.....	نتیجه‌گیری	۷-۹	۶۱۶
۲۹۰.....	تمرین‌ها		۶۱۶
۲۹۱.....	فصل دهم: برنامه‌نویسی CUDA C/C++		
۲۹۲.....	برنامه سریال و موازی	۱-۱۰	
۲۹۲.....	اهداف طراحی CPU و GPU	۱-۱-۱۰	

- ۲-۱-۱۰ الگوریتم‌هایی با عملکرد بهتر بر روی CPU ۲۹۵
- ۲-۱۰ نمونه‌هایی از برنامه‌های مبتدی تا پیشرفته CUDA ۲۹۸
- ۱-۲-۱۰ چاپ جمله "Hello World" به زبان CUDA ۲۹۸
- ۲-۲-۱۰ محاسبه‌ی حاصل جمع دو عدد ۲۹۹
- ۳-۱۰ برنامه‌نویسی موازی با CUDA ۳۰۱
- ۱-۳-۱۰ جمع ماتریس‌ها ۳۰۶
- ۱-۱-۳-۱۰ بررسی کلاس کار با ماتریس‌ها به صورت موازی ۳۰۶
- ۲-۱-۳-۱۰ بررسی کلاس کار با ماتریس‌ها به صورت سریال ۳۰۹
- ۳-۱-۳-۱۰ بررسی کلاس اصلی ۳۱۱
- ۲-۱-۳-۱۰ تحلیل رفتارهای نرم‌افزار در زمان اجرا به صورت سریال و ۳۱۲
- ۱-۳-۳-۱۰ تحلیل رفتار نرم‌افزار در پیشروی در سطر با پیشروی در ستون با رویکرد سریال ۳۱۳
- ۲-۲-۳-۱۰ تحلیل رفتار نرم‌افزار در پیشروی در سطر با پیشروی در ستون با رویکرد موازی ۳۱۳
- ۳-۲-۳-۱۰ تحلیل رفتار نرم‌افزار در پیشروی در سطر در دو حالت سریال و موازی ۳۱۳
- ۴-۲-۳-۱۰ تحلیل رفتار نرم‌افزار در پیشروی در دو حالت سریال و موازی ۳۱۴
- ۳-۳-۱۰ بررسی منطق حالت به سطر ۳۱۴
- ۱-۳-۳-۱۰ بررسی توابع مرتبط به جریات سریال، ماتریس‌ها به زبان C سنتی و CUDA C ۳۱۴
- ۴-۳-۱۰ تحلیل رفتارهای نرم‌افزار در زمان اجرا در محیط ووال استودیو و زبان C ۳۱۹
- ۱-۴-۳-۱۰ تحلیل رفتار نرم‌افزار در جمع در سطر به صورت ۳۱۹
- ۲-۴-۳-۱۰ تحلیل رفتار نرم‌افزار در زمان جمع دو ماتریس به صورت ورت سریال و جمع به کمک CUDA ۳۱۹
- ۴-۱۰ محاسبه مدت زمان اجرای برنامه ۳۲۰
- ۱-۴-۱۰ محاسبه مدت زمان اجرای برنامه‌های سریال در زبان C ۳۲۰
- ۲-۴-۱۰ محاسبه مدت زمان اجرای برنامه‌های موازی در زبان C ۳۲۱
- ۵-۱۰ تعیین مشخصات دستگاه ۳۲۵
- ۶-۱۰ پردازش مجموعه داده‌ها ۳۲۹
- ۱-۶-۱۰ استفاده از ballot و سایر عملیات ذاتی ۳۳۱
- ۷-۱۰ نمونه‌ای با استفاده از AES ۳۴۰
- ۱-۷-۱۰ الگوریتم AES ۳۴۱
- ۱-۱-۷-۱۰ جایگزینی ۳۴۲
- ۲-۱-۷-۱۰ چرخاندن سطر به چپ ۳۴۴
- ۳-۱-۷-۱۰ ترکیب ستون‌ها ۳۴۴
- ۴-۱-۷-۱۰ افزودن کلید دور ۳۴۵
- ۵-۱-۷-۱۰ استخراج کلیدهای دور ۳۴۵
- ۲-۷-۱۰ پیاده‌سازی سریال AES ۳۴۸

۳۴۸.....	اندازه دسترسی ۱-۲-۷-۱۰
۳۴۸.....	معاوضه بین حافظه و عملیات ۲-۲-۷-۱۰
۳۴۸.....	کرنل اولیه ۳-۲-۷-۱۰
۳۵۵.....	کارایی کرنل ۳-۷-۱۰
۳۶۰.....	کارایی انتقال ۴-۷-۱۰
۳۶۱.....	نسخه‌ی جریانی ۵-۷-۱۰
۳۶۳.....	مقایسه با CPU ۶-۷-۱۰
۳۷۴.....	ملاحظات برای اجرا روی GPUهای دیگر ۷-۷-۱۰
۳۷۷.....	خلاصه AES ۸-۷-۱۰
۳۷۸.....	۸-۱۰ نتیجه‌ی با استفاده از ALC-PSO
۳۷۸.....	روند اجرا ۱-۱-۱۰
۳۸۰.....	تغییرها ۲-۸-۱۰
۳۸۱.....	مقدار داده اولیه ۳-۸-۱۰
۳۸۱.....	به روز رسانی سرعت، موقعیت و بهترین موقعیت ذرات ۴-۸-۱۰
۳۸۳.....	به روز رسانی برای اجرا ۵-۸-۱۰
۳۸۳.....	پیاده‌سازی الگوریتم برای اجتماعی ذرات با رهبر سالخورده و رقبا بر روی GPU ۶-۸-۱۰
۳۸۳.....	محیط پیاده‌سازی و ابعاد ۱-۶-۸-۱۰
۳۸۴.....	نتایج اجرا بر روی CPUهای مختلف ۲-۶-۸-۱۰
۳۸۵.....	نتایج اجرا بر روی GPUهای مختلف ۳-۶-۸-۱۰
۳۹۳.....	مقایسه کارایی اجرای ALC-PSO بر روی GPU و CPU ۴-۶-۸-۱۰
۳۹۳.....	مقایسه‌ی کارایی GPU نسبت به CPU با تعداد تکرارهای کمتر ۵-۶-۸-۱۰
۳۹۴.....	مقایسه‌ی کارایی GPU نسبت به CPU با تعداد تکرارهای ثابت ۶-۶-۸-۱۰
۳۹۸.....	۹-۱۰ نتیجه‌گیری
۳۹۹.....	تمرین‌ها
۴۰۱.....	پیوست بسط زبان C
۴۱۱.....	واژه‌نامه‌ی فارسی به انگلیسی
۴۱۵.....	واژه‌نامه انگلیسی به فارسی
۴۲۰.....	منابع

پیشگفتار

نرم افزار مجموعه ای از دستورالعمل های یک زبان برنامه نویسی کامپیوتری است که به منظور به کارگیری سخت افزار در جهت برآورد هدف خاصی، توسط برنامه نویسان نوشته می شود. به عبارت دیگر در یک نرم افزار خواسته های کاربر به زبان قابل فهم برای سخت افزار، جمع آوری و منتقل می شود. برای کاربردهای مختلف، زبان های برنامه نویسی متفاوتی طراحی و بهسازی شده اند و معمولاً برنامه نویسان برنامه ها را به صورت سریال پیاده سازی می کنند. در سال های اخیر، انواع مختلف پردازنده های چند هسته ای با قابلیت های پردازش موازی مانند پردازش های موازی دستور، داده، نخ و ... به در سیستم های کامپیوتری عرضه شده اند که باعث تغییرات اساسی در صنعت نرم افزار کامپیوتر و حرکت از دنیای تک هسته ای سریال به سمت برنامه نویسی موازی شده است. برنامه نویسی موازی به منظور پردازش موازی الگوریتم ها با هدف افزایش کارایی محاسبات به کمک چندین پردازنده کوچک یا بزرگ مورد استفاده قرار می گیرد.

برای نوشتن یک برنامه موازی، باید الگوریتم ها و کاربردهای مورد نظر، بررد تجزیه و تحلیل قرار گیرد و انواع سطوح موازی سازی با استفاده از ابزارهای مناسب استخراج شود. سپس سطوح موازی سازی مختلف استخراج شده باید با کتابخانه ها و زبان های برنامه نویسی موازی بر روی پلتفرم ها و پردازنده ها موجود که قابلیت پردازش موازی را با توجه به مدل های استخراج شده دارند، طراحی و پیاده سازی شود. مدل های برنامه نویسی موازی Single Instruction and Multiple Data (SIMD)، Open Multi-Processing (OpenMP)، Open Computing

Language

در پلتفرم‌های سخت‌افزاری مختلف طراحی و مورد استفاده قرار می‌گیرند. همچنین قابلیت برنامه‌نویسی موازی در زبان‌های C#، C++ و غیره فراهم شده است. در میان انواع مدل‌های برنامه‌نویسی موازی، مدل CUDA که توسط شرکت NVIDIA بر روی پلتفرم واحد پردازشگر گرافیکی (Graphics Processing Unit) پیاده‌سازی شده است از اهمیت خاصی برخوردار است. معماری واحد پردازشگر گرافیکی (GPU)، قابلیت موازی‌سازی بالایی را با استفاده از اجرای نخ‌های مختلف بر روی تعداد زیاد هسته‌های پردازشگر فراهم می‌کند و سرعت آن بسیار بیشتر از واحد پردازشگر مرکزی (CPU) است. با توجه به توان پردازشی بالای GPU در کاربردهای غیرگرافیکی، شاخه‌ی جدیدی در علوم کامپیوتر به نام محاسبات با اهداف همه‌منظوره بر روی واحد GPU ایجاد شده است. هدف فعالان این عرصه، بهره‌گیری از GPU به عنوان یک کمک‌پردازنده محاسباتی در برنامه‌های غیرگرافیکی و برنامه‌های عمومی است. پردازنده‌های گرافیکی جدید، از چندین پردازنده با کارایی بالا تشکیل شده‌اند که قادر به انجام محاسبات بسیار سنگین هستند.

در حال حاضر، با توجه به اهمیت مفاهیم پردازش و برنامه‌نویسی موازی، دروس "پردازش موازی" و "معماری سیستم‌های موازی"، جزو دروس اصلی و تخصصی گرایش‌های مختلف کارشناسی ارشد رشته مهندسی کامپیوتر است و در دانشگاه‌های معتبر تدریس می‌شود. در این دروس، سیستم‌های پردازش موازی و کاربردهای آنها و دلایل مورد نیاز برای پردازش موازی و موانع موجود مورد بررسی قرار می‌گیرد. همچنین انواع معماری‌های پردازش موازی و شبکه‌های ارتباطی، روش‌های مختلف تجزیه مسائل، استخراج انواع سطوح موازی‌سازی موجود در الگوریتم‌ها، شناسایی بخش‌های محاسباتی و مان‌گیر کاربردها مورد بحث قرار می‌گیرند.

لذا با توجه به اهمیت دروس پردازش موازی و برنامه‌نویسی موازی در سطوح کارشناسی ارشد و دکتری در رشته‌های مهندسی کامپیوتر، برق و فناوری اطلاعات، لزوم دسترسی بودن منابع مناسب تخصصی در این زمینه ضروری است. با توجه به کمبود منابع فارسی در زمینه پردازش موازی، برنامه‌نویسی موازی با استفاده از پلتفرم GPU، انگیزه‌ای شده تا نویسندگان این مجموعه، با توجه به تجربه چندین ساله‌ی خود در تدریس این دروس، این نوشتار را فراهم کنند و در اختیار جامعه‌ی دانشگاهی کشور قرار دهند. در این کتاب فرض شده است که خوانندگان هیچ‌آشنایی قبلی با CUDA یا برنامه‌نویسی موازی ندارند و فقط با زبان برنامه‌نویسی C/C++ آشنایی دارند. با پیش رفتن در فصل‌های کتاب و ایجاد رغبت نسبت به یادگیری CUDA، موضوعات پیشرفته‌تری پوشش داده شده است. برای برنامه‌نویسانی که با مفهوم برنامه‌نویسی موازی و CUDA آشنایی دارند، جزئیات معماری GPU نیز شرح داده می‌شود.

ساختار کتاب به دو بخش تقسیم شده است. بخش اول شامل فصول اول تا چهارم که مفاهیم اولیه سیستم‌های کامپیوتری، پردازش موازی، نحوی موازی‌سازی الگوریتم‌ها و مدل‌های مختلف برنامه‌نویسی موازی مطرح شده است و بخش دوم شامل فصول پنجم تا دهم، معماری GPU و تفاوت‌های اساسی آن با CPU، کاربردها و برنامه‌نویسی با استفاده از مدل CUDA مورد بحث و بررسی قرار گرفته است. به طور خلاصه فصول کتاب شامل مفاهیم زیر هستند.

فصل اول: توسعه سیستم‌های کامپیوتری، در این فصل سعی شده است اطلاعاتی درباره‌ی تاریخچه‌ی پیدایش سیستم‌های کامپیوتری و روند رشد سخت‌افزار و نرم‌افزار در اختیار خوانندگان قرار گیرد.

فصل دوم: پردازش موازی، این فصل چگونگی پیدایش پردازش موازی را مورد بررسی قرار می دهد و به این سوال پاسخ می دهد که هدف از موازی سازی چیست و چه نوع الگوریتم هایی را می توان موازی سازی کرد.

فصل سوم: نحوه موازی سازی الگوریتم ها، در این فصل انواع روش های موازی سازی شرح داده می شود، همچنین به چگونگی موازی سازی الگوریتم های سریال و تکنیک های بهینه سازی کامپایلر پرداخته می شود.

فصل چهارم: مدل های برنامه نویسی موازی، در این فصل انواع مدل های برنامه نویسی موازی تشریح می شود و برای درک بهتر آنها، چگونگی موازی سازی الگوریتم ضرب ماتریس ها در هر کدام از این مدل ها بیان می شود.

فصل پنجم: معماری GPU، این فصل به شرح جزئیات سخت افزار و معماری دستگاه های CUDA می پردازد. برای دستیابی به کارایی بالا در برنامه نویسی CUDA، اطلاعات کاملی راجع به معماری GPU مورد نیاز است.

فصل ششم: مفهوم موازی سازی از منظر GPU، این فصل مقدمه ای بر مفاهیم برنامه نویسی موازی از جمله تفاوت میان برنامه های سریال و موازی و چگونگی حل مسائل به روش های متفاوت است. در واقع این فصل به برنامه نویسان به صورت سریال برنامه نویسی می کنند کمک می کند که درک ابتدایی از مفاهیمی که در فصل های بعدی در این کتاب بدان پرداخته می شود، به دست آورند.

فصل هفتم: راه انداز CUDA، در این فصل فرایند نصب CUDA SDK تحت سیستم عامل ویندوز شرح داده می شود.

فصل هشتم: آشنایی با بلوک ها و نخ ها، این فصل به شرح جزئیات مدل نخ بندی CUDA می پردازد و شامل مثال هایی است که به انتخاب بهتر این تاثیر گذاری می تواند بر روی کارایی کمک می کند.

فصل نهم: مدیریت حافظه توسط CUDA، تفاوت انواع حافظه و چگونگی به کارگیری آنها توسط CUDA به عنوان بزرگترین عامل تأثیرگذار بر کارایی، موضوع این فصل است. توضیحات همراه با مثال هایی از چگونگی کارکرد انواع مختلف با جزئیات کافی تشریح می شود.

فصل دهم: برنامه نویسی CUDA C/C++، در این فصل به چگونگی همکاری بین CPU و GPU پیرامون مثال های مختلف پرداخته می شود.

اسداله شاه بهرامی

گروه مهندسی کامپیوتر دانشگاه گیلان

صدیقه جم

پاییز ۱۳۹۶

در پایان شایسته است که از کلیه همکاران و دانشجویان گروه مهندسی کامپیوتر دانشگاه گیلان و همی عزیزانی که در تهیه این کتاب همکاری ذی قیمتی نگارندگان داشته اند، سپاسگزاری و قدردانی شود. همچنین تمام خوانندگان گرامی دعوت می شود نظرات، پیشنهادها و انتقادات خود را از طریق ایمیل (shahbahrami@guil.ac.ir) در اختیار نویسندگان قرار داده تا کیفیت کتاب در چاپ های بعدی بهبود داده شود.