

۱۴۸۴۰۱۶
۹۵-۱۴۱۸

به نام خدا



مؤسسه فرهنگی هنری
دییگران تهران

دنیای متفاوت کلان داده و چارچوب هدوپ

مؤلفان :

امیر مولوی خایانی

(کارشناس ارشد فناوری اطلاعات)

دکتر امیر مسعود رحمانی

(دکتری تخصصی مهندسی کامپیوتر)

هر گونه چاپ و تکثیر از محتویات این کتاب بدون اجازه کتبی
ناشر ممنوع است. متخلفان به موجب قانون حمایت حقوق
مؤلفان، مصنفان و هنرمندان تحت پیگرد قانونی قرار می گیرند.

دنیای متفاوت کلان داده

و چارچوب هادوپ

مؤلفان: مهندس امیر مولوی خیائی

دکتر امیر سعود رحمانی

ناشر: مؤسسه فرهنگی نشر دیباگران تهران

حروفچینی و صفحه آرایی: نازنین صیری

طرح روی جلد: مجتبی حبیبی

چاپ: دانشجو

نوبت چاپ: اول

تاریخ نشر: ۱۳۹۵

تیراژ: ۱۰۰ جلد

قیمت: ۲۵۰۰۰۰ ریال

شابک: ۹۷۸-۶۰۰-۱۲۴-۵۶۳-۳

نشانی واحد فروش: تهران، میدان انقلاب،

خ کارگر جنوبی، روبروی پاساژ مهستان،

پلاک ۱۲۵۱

سرشناسه: مولوی خیائی، امیر، ۱۳۷۰-

عنوان و نام پدید آور: دنیای متفاوت کلان داده و چارچوب
هادوپ/مؤلف: امیر مولوی خیائی

مشخصات نشر: تهران- دیباگران تهران- ۱۳۹۵

مشخصات ظاهری: ۲۴۱ ص. مصور.

شابک: ۹۷۸-۶۰۰-۱۲۴-۵۶۳-۳

وضعیت فهرست نویسی: فیبا

یادداشت: واژه نامه.

یادداشت: کتابنامه: ص: ۲۳۱

موضوع: آپاچی هادوپ

موضوع: Apache Hadoop

موضوع: داده های کلان

موضوع: Big data

موضوع: سازنده: آ.ا.ها (کامپیوتر)

موضوع: file organization (computer science)

شناسه افزوده: رحمانی، امیر، مس: ۱۳۵۲-

رده بندی کنگره: ۸۱۳۹۵ م ۸۲۲ TK ۵۱۰۰۸۸۸

رده بندی دیویی: ۰۰۵/۳۱۳۷۶

شماره کتابشناسی ملی: ۴۶۲۷۴۷۹

تلفن: ۲۲۰۸۵۱۱۱-۶۶۴۱۰۰۴۶ کد پستی: ۱۳۱۴۹۸۳۱۸۵

پست الکترونیکی: bookmarket@mftmail.com

فروش اینترنتی:

www.mftshop.com

www.mftbook.ir

نشانی اینستاگرام: Dibagaran_publishing

نشانی تلگرام: @mftbook

فهرست مطالب

۱.....	عنوان کتاب
۲.....	شناسنامه کتاب
۱۴.....	مقدمه ناشر
۱۵.....	پیشگفتار
۱۷.....	مقدمه
۱۷.....	سیستم‌های موانع
۱۸.....	سیستم‌های ترزیع شده
۱۸.....	مزایای استفاده از سیستم‌های ترزیع شده
۱۹.....	محاسبات گرید
۲۰.....	محاسبات داوطلبانه
۲۱.....	محاسبات ابری
۲۲.....	مدل‌های سرویس پردازش ابری
۲۳.....	محاسبات به صورت مه
۲۴.....	خط زمانی کلان داده
۲۷.....	معماری کلان داده
۲۷.....	نقشه راه کتاب
۲۹.....	فصل اول
۲۹.....	مقدمه‌ای بر کلان داده و هدوپ
۲۹.....	مقدمه‌ای بر کلان داده و هدوپ
۳۰.....	مدل تعریفی کلان داده

۳۱	حجم
۳۲	سرعت تولید
۳۳	تنوع
۳۳	ارزش
۳۳	صحت
۳۳	اعتبار
۳۴	اضافات
۳۴	آسیب پذیری
۳۴	تطبیق پذیری
۳۴	چسبناکی
۳۵	نوسان
۳۵	مصورسازی
۳۵	درک کلان داده
۳۷	NoSQL
۳۹	پایگاه داده‌های تحلیلی
۳۹	کلان داده چگونه ایجاد شده است؟
۴۰	یک مورد کاربرد از کلان داده
۴۲	الگوهای مورد کاربرد کلان داده
۴۲	استفاده از کلان داده به عنوان الگوی ذخیره سازی
۴۳	استفاده از کلان داده به عنوان الگوی تبدیل داده‌ها
۴۵	استفاده از کلان داده به عنوان الگوی تحلیل داده‌ها
۴۵	استفاده از کلان داده به عنوان الگوی داده‌های بلادرنگ
۴۶	استفاده از کلان داده به عنوان الگوی استفاده از حافظه نهان با تاخیر کم

۴۹	تاریخچه هدوپ
۵۰	مزایای هدوپ
۵۱	کاربردهای هدوپ
۵۲	چارچوب هدوپ
۵۳	آپاچی هدوپ
۵۴	توزیع های هدوپ
۵۶	ارکان هدوپ
۵۶	مؤلفه های دسترسی به داده
۵۷	مؤلفه ذخیره سازی داده
۵۷	مؤلفه های فروریز داده و هدوپ
۵۸	مؤلفه های تحلیل بلادرنگ و جریانی
۵۸	جمع بندی
۵۹	فصل دوم
۵۹	چارچوب هدوپ
۵۹	چارچوب هدوپ
۶۰	سیستم های مرسوم
۶۱	مسیر حرکتی پایگاه داده ها
۶۲	موارد کاربرد هدوپ
۶۳	جریان داده ای هدوپ
۶۵	یکپارچگی هدوپ
۶۶	چارچوب هدوپ
۶۶	سیستم فایل توزیع شده
۶۷	HDFS
۶۷	برنامه نویسی توزیع شده

۶۹	 پایگاه داده‌های غیر رابطه‌ای
۶۹	 آپاچی HBase
۶۹	 فروبری داده
۷۰	 سرویس برنامه‌نویسی
۷۲	 زمان‌بندی
۷۲	 تحلیل داده و یادگیری ماشین
۷۳	 مدیریت سیستم
۷۴	 جمع‌بندی
۷۵	 فصل سوم
۷۵	 ارکان هدوپ
۷۵	 ارکان هدوپ
۷۶	 HDFS
۷۶	 شاخصه‌ای HDFS
۷۷	 معماری HDFS
۷۸	 NameNode
۷۹	 DataNode
۸۰	 Secondary NameNode یا Checkpoint NameNode
۸۰	 BackupNode
۸۰	 طبقه‌بندی داده درون HDFS
۸۱	 جریان خطی پردازش خواندن
۸۳	 جریان خطی پردازش نوشتن
۸۵	 خودآگاهی قفسه
۸۵	 مزایای خودآگاهی قفسه در HDFS
۸۶	 HDFS

۸۶	محدودیت‌های HDFS
۸۸	مزیت فدراسیون HDFS
۸۹	MapReduce
۸۹	معماری MapReduce
۹۰	JobTracker
۹۰	TaskTracker
۹۰	سریال‌سازی نوع‌های داده
۹۱	رابط Writable
۹۱	رابط WritableComparable
۹۲	مثالی از عملکرد MapReduce
۹۳	فرآیندهای MapReduce
۹۵	Mapper
۹۵	Sorting و Shuffle
۹۵	Reducer
۹۶	اجرای گمانی
۹۷	قالب‌های فایل
۹۷	قالب‌های ورودی
۹۸	RecordReader
۹۹	قالب‌های خروجی
۱۰۰	RecordWriter
۱۰۱	مراحل کمکی
۱۰۱	Combiner
۱۰۲	Partitioner
۱۰۲	افراز کننده سفارشی

۱۰۴.....	YARN
۱۰۵.....	معماری YARN
۱۰۶.....	ResourceManager
۱۰۶.....	NodeManager
۱۰۶.....	ApplicationMaster
۱۰۷.....	برنامه‌های قدرت گرفته از YARN
۱۰۷.....	جمع‌بندی
۱۰۹.....	فصل چهارم
۱۰۹.....	مؤلفه‌های دسترسی داده
۱۰۹.....	مؤلفه‌های دسترسی داده
۱۱۰.....	نیاز ابزار پردازشی در هادوپ
۱۱۰.....	Pig
۱۱۱.....	نوع‌های داده‌های Pig
۱۱۲.....	معماری Pig
۱۱۲.....	طرح منطقی
۱۱۳.....	طرح فیزیکی
۱۱۳.....	طرح MapReduce
۱۱۳.....	حالت‌های Pig
۱۱۴.....	Hive
۱۱۵.....	معماری Hive
۱۱۶.....	Metastore
۱۱۶.....	کامپایلر پرس و جو
۱۱۷.....	موتور اجرایی
۱۱۷.....	

۱۱۸.....	جمع‌بندی
۱۲۱.....	فصل پنجم
۱۲۱.....	مؤلفه ذخیره‌سازی غیر رابطه‌ای
۱۲۱.....	مؤلفه ذخیره‌سازی غیر رابطه‌ای
۱۲۲.....	نگاه کلی به HBase
۱۲۳.....	مزایای HBase
۱۲۴.....	معماری HBase
۱۲۵.....	Master Server
۱۲۶.....	Region Server
۱۲۶.....	WAL
۱۲۷.....	BlockCache
۱۲۷.....	LRUBlockCache
۱۲۸.....	SlabCache
۱۲۸.....	BucketCache
۱۲۸.....	ناحیه‌ها
۱۲۹.....	MemStore
۱۲۹.....	Zookeeper
۱۳۰.....	مدل داده‌ای HBase
۱۳۰.....	مؤلفه‌های منطقی به عنوان مدل داده‌ای
۱۳۳.....	خصوصیه‌های ACID
۱۳۳.....	نظریه CAP
۱۳۵.....	طراحی شما
۱۳۶.....	جریان خطی پردازش نوشتن
۱۳۷.....	جریان خطی پردازش خواندن

۱۳۷.....	فشرده‌سازی
۱۳۸.....	سیاست فشرده‌سازی
۱۳۸.....	فشرده‌سازی جزئی
۱۳۹.....	فشرده‌سازی عمده
۱۳۹.....	قطعه‌بندی
۱۳۹.....	پیش قطعه‌بندی
۱۴۰.....	قطعه‌بندی خودکار
۱۴۰.....	قطعه‌بندی اجزای
۱۴۰.....	یکپارچه‌سازی HBase و Hive
۱۴۱.....	تنظیم عملکرد
۱۴۲.....	فشرده‌سازی
۱۴۲.....	فیلتر کردن
۱۴۳.....	شمارنده
۱۴۳.....	پردازشگرهای همکار
۱۴۵.....	جمع‌بندی
۱۴۷.....	فصل ششم
۱۴۷.....	فروبری داده‌ها در هدوپ
۱۴۷.....	فروبری داده‌ها در هدوپ
۱۴۸.....	منابع داده
۱۴۹.....	چالش‌های موجود در فروبری داده
۱۴۹.....	Sqoop
۱۵۰.....	اتصال دهنده‌ها و راه‌اندازها
۱۵۱.....	معماری نسخه اول Sqoop
۱۵۱.....	Sqoop

۱۵۲.....	Sqoop معماری نسخه دوم
۱۵۵.....	Flume
۱۵۵.....	قابلیت اطمینان
۱۵۶.....	Flume معماری
۱۵۷.....	توپولوژی چندردهای
۱۵۸.....	Flume Master
۱۵۸.....	Flume Node ها
۱۵۹.....	مؤلفه‌های عامل
۱۶۳.....	جمع‌بندی
۱۶۵.....	فصل هفتم
۱۶۵.....	جریان‌سازی و تحلیل بلادرنگ
۱۶۵.....	جریان‌سازی و تحلیل بلادرنگ
۱۶۶.....	Storm معرفی
۱۶۷.....	Storm معماری مدل
۱۶۸.....	Storm معماری داده‌ای
۱۶۹.....	Storm توپولوژی
۱۷۰.....	Storm در کنار YARN
۱۷۰.....	Storm معرفی
۱۷۲.....	SPARK خصوصیات
۱۷۲.....	SPARK چارچوب
۱۷۳.....	SPARKSQL
۱۷۳.....	GraphX
۱۷۳.....	MLib
۱۷۳.....	SPARK Streaming

۱۷۵.....	معماری SPARK
۱۷۸.....	معماری SPARK
۱۷۹.....	عملیات قابل انجام در SPARK
۱۸۰.....	جمع‌بندی
۱۸۱.....	فصل هشتم
۱۸۱.....	مورد مطالعاتی؛ فناوری خط تولید
۱۸۱.....	مورد مطالعاتی؛ فناوری خط تولید
۱۸۲.....	چه مک‌ایز می‌برای کاهش هزینه و افزایش بهره‌وری خط تولید پیاده‌سازی میشوند
۱۸۶.....	نگاهی به آینده داده
۱۸۷.....	حریم خصوصی در مجموعه داده‌ها، عمومی
۱۸۸.....	اینترنت اشیاء
۱۸۸.....	توسعه نرم‌افزار
۱۸۹.....	نتیجه‌گیری
	پیوست شماره ۱: نحوه‌ی نصب چارچوب هادوپ بر روی سیستم عامل لینوکس توزیع
۱۹۱.....	Ubuntu
۱۹۵.....	نصب و پیکربندی هادوپ به صورت شبه توزیع شده بر روی یک گ
۲۰۶.....	پیوست شماره ۲: پورت‌های کاری HDFS و نحوه‌ی عملکرد آن‌ها
۲۰۸.....	پیوست شماره ۳: نحوه‌ی کدنویسی برای یک کار MapReduce
۲۰۸.....	Mapper
۲۰۹.....	Reducer
۲۱۰.....	Driver
۲۱۴.....	پیوست شماره ۴: تنظیمات پیکربندی سیاست‌های فشرده‌سازی HBase
۲۱۴.....	فشرده‌سازی جزئی
۲۱۸.....	

۲۱۶.....	پیوست شماره ۵: یک مثال از پیکربندی توپولوژی Storm
۲۱۶.....	پیاده‌سازی Spout ها
۲۱۸.....	پیاده‌سازی Bolt ها
۲۲۰.....	توپولوژی
۲۲۲.....	پیوست شماره ۶: تکه کد شمارش کلمات برای موتور پردازشی Spark به زبان جاوا
۲۲۳.....	لیست واژگان تخصصی از فارسی به انگلیسی
۲۳۱.....	مراجع

خط مشی کیفیت انتشارات مؤسسه فرهنگی هنری دیباگران تهران در عرصه کتاب الکترونیک است که بتواند خواسته های به روز جامعه فرهنگی و علمی کشور را تا حد امکان پوشش دهد.

حمد و سپاس ایزد منان را که با الطاف بیکران خود این توفیق را به ما ارزانی داشت تا بتوانیم در راه ارتقای دانش عمومی و فرهنگی این مرز و بوم در زمینه چاپ و نشر کتب علمی دانشگاهی، علوم پایه و به ویژه عرصه کامپیوتر و انفورماتیک گام‌هایی هرچند کوچک برداشته و در انجام رسالتی که بر عهده داریم، مؤثر واقع شویم.

گسترده‌گی علوم و ترسعه ررافقه آن، شرایطی را به وجود آورده که هر روز شاهد تحولات اساسی چشمگیری در سطح جهان هستیم. این گسترش و توسعه نیاز به منابع مختلف از جمله کتاب را به عنوان قدیمی‌ترین و راحت‌ترین دست‌یابی به اطلاعات و اطلاع‌رسانی، بیش از پیش روشن می‌نماید.

در این راستا، واحد انتشارات مؤسسه فرهنگی هنری دیباگران تهران با همکاری جمعی از اساتید، مؤلفان، مترجمان، متخصصان، پژوهشگران، محققان و نیز پرسنل ورزیده و ماهر در زمینه امور نشر درصدد هستند تا با تلاش‌های مستمر خود برای رفع کمبودها و نیازهای موجود، منابعی پربار، معتبر و با کیفیت مناسب در اختیار علاقمندان قرار دهند.

کتابی که در دست دارید با همت "آقایان مهندس میر ولوی، خایانی - دکتر امیر مسعود رحمانی" و تلاش جمعی از همکاران انتشارات میسر گشته که شایسته است از یکایک این گرامیان تشکر و قدردانی کنیم.

کارشناسی و نظارت بر محتوا: زهره قزلباش

در خاتمه ضمن سپاسگزاری از شما دانش‌پژوه گرامی درخواست می‌نماید که مراجعه به آدرس dibagaran.mft.info (ارتباط با مشتری) فرم نظرسنجی را برای کتابی که در دست دارید تکمیل و ارسال نموده، انتشارات دیباگران تهران را که جلب رضایت و وفاداری مشتریان را هدف خود می‌داند، یاری فرمایید.

امیدواریم همواره بهتر از گذشته خدمات و محصولات خود را تقدیم حضورتان نماییم.

مدیر انتشارات

مؤسسه فرهنگی هنری دیباگران تهران

Publishing@mftmail.com

امروزه با گره خوردن زندگی بشر با دنیای دیجیتال و ورود اینترنت اشیاء به زندگی روزمره ما با پدیده‌هایی روبرو شدیم که از آن با نام انفجار اطلاعات خام یاد می‌شود. این حجم انبوه اطلاعاتی نیاز به راهکارهای نوین ذخیره‌سازی دارند و از طرف دیگر شیوه‌های مرسوم پردازشی راهکار متناسبی برای پردازش و کشف دانش از این اطلاعات خام ارائه نمی‌کنند. از این رو شیوه‌های نوین ذخیره‌سازی همانند سیستم فایل توزیع شده و پایگاه داده‌های غیر رابطه‌ای به کمک این پدیده آمده‌اند و از طرف دیگر با بهره‌گیری از سیستم‌های توزیع شده نیازهای پردازشی تا حدی رفع شده است. با ارائه مفهومی که با نام نگاشت - تجمیع از آن یاد می‌شود، توانست‌اند فرآیندهای پردازشی را در میان طیف وسیعی از سیستم‌های کامپیوتری توزیع کنند و پس از اتمام پردازش، نتایج به دست آمده را با هم ادغام کرده و به نتیجه‌ی نهایی دست پیدا می‌کنند.

این کتاب با این هدف نگاشته شده است که بتواند مباحث علمی کلان داده و فناوری‌های وابسته را پوشش دهد و از این رو متداول‌ترین چالش‌ها و پرسش‌ها را با نام هدوب از لحاظ فنی بررسی شده است و سعی شده تمام موارد ضروری که در زمان نگارش کتاب موجود بوده است در آن گنجانده شود.

خواهشمندم هرگونه اشکال و ناراسایی در مطالب، به ایمیل زیر ارسال کنید تا اصلاحات لازم انجام شود؛

امیر مولوی خایانی

امیر مسعود رحمانی

زمستان ۱۳۹۵

books@amirinfo.ir

با اینکه هدف از نگارش این کتاب معرفی کلان داده، سیستم‌های مرتبط و کالبد شکافی مؤلفه‌های آن است، قصد داریم مقدمه کتاب را به معرفی سیستم‌های توزیع شده و تعاریف بنیادی آن اختصاص دهیم. از این رو که ممکن است شما خواننده گرامی دانش نسبی کافی با این مقوله نداشته باشید، سعی شده به صورت خلاصه دید مناسبی از این مقوله را در اختیار شما بگذاریم و این فصل بتواند سنگ بنایی برای باقی فصل‌های کتاب قرار گیرد.

سیستم‌های موازی

بدیهی است که در یک پردازنده تک هسته‌ای، عمل پردازش موازی انجام نمی‌شود. هر هسته پردازنده در آن واحد قادر به انجام یک پردازش است و در واقع سرعت بالای پردازش فرآیندها و جابجایی میان آن‌ها است که به کاربر این حس را القا می‌کند که تمام عملیات به صورت موازی صورت می‌گیرد اما این تصور اشتباه است و در واقع زمانی پردازشی یک پردازنده، بر اساس تعداد عملیات و اولویت میان آن‌ها تقسیم می‌شود و تا اتمام پردازش این عمل صورت می‌گیرد. حال اگر از الگوی برنامه نویسی موازی استفاده کنیم در واقع برنامه‌ای ایجاد کرده‌ایم که فرآیندهای آن به تعدادی نخ تقسیم می‌شود زمانی که پردازنده برای اجرای کامل این برنامه صرف می‌کند، بیشتر شود، زیرا اتلافی را افزایش داده‌ایم که در جابجایی بین نخ‌ها به وجود می‌آید. با توجه به این نکته وجود پردازنده‌های چند هسته‌ای شکل گرفت که در آن واحد قادر باشیم فرآیندهای مختلف را در این هسته‌های یک پردازنده توزیع کنیم و سرعت پردازش را بالاتر ببریم. این اولین قدم در مورد پیچیده‌تر شدن سخت‌افزارها بود تا مفهومی با نام سیستم‌های موازی^۱ همانند چند پردازنده‌ای، چند هسته‌ای و ... شکل بگیرد؛ سیستم‌هایی که توانایی اجرای همزمان چندین عملیات را دارند که همه‌ی آن‌ها از یک حافظه اصلی به صورت مشترک استفاده می‌کند.

اشکال عمده این نوع سیستم‌ها، وابستگی کامل به قدرت سخت‌افزار و پیاده‌سازی آن‌ها است. یعنی در صورت نیاز به افزایش کارایی، تنها راه‌های ممکن ارتقاء سخت‌افزار یا بهینه کردن الگوریتم‌های برنامه است. این عمل را که در اصطلاح، ارتقای عمودی^۲ می‌نامند، روش مناسبی برای پیاده‌سازی موازی سازی

^۱ Parallel Systems

^۲ Vertical Scaling

نیست چون هزینه‌های گزافی به همراه دارد و یک سیستم به تنهایی قادر به پشتیبانی درخواست‌های زیاد نیست. همین اتفاق است که موجب می‌شود مفهوم سیستم‌های توزیع شده به یاری سیستم‌های موازی بیاید و باعث شود که نقاط ضعف آن را پوشش دهد که در اصطلاح، ارتقای افقی^۱ نامیده می‌شود.

سیستم‌های توزیع شده

تفاوتی که در سیستم‌های موازی و سیستم‌های توزیع شده^۲ وجود دارد، عدم وجود حافظه اصلی اشتراکی است. از سیستم‌های توزیع شده اینگونه یاد می‌شود؛ مجموعه‌ای از سخت‌افزارها و نرم‌افزارها که برای رسیدن به یک هدف واحد از طریق شبکه با یکدیگر در ارتباط هستند. تعریف دیگری از این سیستم‌ها نیز می‌توان داشت. مجموعه‌ای از سیستم‌های کامپیوتری مستقل از هم و خودمختار که از نظر کاربر یک سیستم کامپیوتری واحد را منسجم به نظر می‌رسد.

مزایای استفاده از سیستم‌های توزیع شده

کارایی بالا: به دلیل همزمانی اجرای عملیات در سخت‌افزارهای مختلف، کارایی این نوع سیستم‌ها در مقابل سیستم‌های متمرکز افزایش می‌یابد.

قابلیت همکاری: به دلیل این که ذات این سیستم‌ها به صورت توزیع شده است، قابلیت اتصال و همکاری میان باقی سیستم‌ها وجود دارد.

قابلیت دسترس پذیری و قابلیت اطمینان: با وجود روش‌های مختلفی همچون چند نسخه‌سازی، در این سیستم‌ها می‌توان به آسانی به این ویژگی‌ها دست یافت.

قابلیت گسترش: به منظور گسترش سیستم و رفع کردن نیازمندی‌های جدید، به راحتی می‌توان مؤلفه‌های مختلف و زیرسیستم‌های جدید را به کل زیرساخت اضافه کرد بدون اینکه سیستم مختل شود.

^۱ Horizontal Scaling

^۲ Distributed Systems

افزایش بهره‌وری: به دلیل ایجاد ساختار سلسله‌مراتبی و شکست سیستم به زیرسیستم‌ها بهره‌وری کلی سیستم افزایش می‌یابد.

قابلیت استفاده مجدد: امکان تعویض زیرسیستم‌ها به راحتی در این سیستم‌ها تعبیه شده است.

کاهش هزینه: به علت قابلیت گسترش راحت و همچنین استفاده مجدد از زیرسیستم‌ها، هزینه‌ها کاهش می‌یابد، اما باید توجه داشت که کل معماری پیاده‌سازی باید به صورت صحیح انتخاب شود.

یک سیستم توزیع شده باید در وهله‌ی اول منابع را به آسانی در اختیار کاربران قرار دهد. دوم شفاف باشد و تمام پیچیدگی‌های یک سیستم توزیع شده را از دید کاربر مخفی کند. سوم اینکه سیستم باز و قابل گسترش باشد؛ به آسانی و با شفافیت کامل قادر به اضافه کردن مؤلفه‌های جدید به سیستم باشیم. چهارم، سیستم مقیاس‌پذیر باشد؛ به صورتی که بدون اینکه سیستم مختل شود، بتوانیم منابع موجود سیستم را افزایش دهیم.

محاسبات گرید^۱

انجمن‌های محاسبات با کارایی بالا^۲ (HPC) در محاسبات گرید سال‌های متوالی است که به پردازش داده‌ها در مقیاس بالا با استفاده از چندین رابط برنامه‌کار ردی^۳ (APIs) به عنوان رابط تبادل پیام^۴ (MPI) مشغول هستند. به طور گسترده، روش مورد استفاده در HPC باعث توزیع عملیات در خوشه‌ای از ماشین‌ها با دسترسی به سیستم‌های فایل اشتراکی با پهنای باند شبکه گسترده ذخیره‌سازی^۵ می‌شود. این کارکرد به طور عمده برای کارهای حساس محاسباتی مناسب است، اما زمانی مشکل ساز می‌شود که یک گره نیاز به دسترسی حجم‌های داده‌ای زیادی دارد، زیرا پهنای باند در این نوع محاسبات یک گلوگاه محسوب می‌شود و پردازش گره‌ها به حالت تعلیق در می‌آید.

MPI کنترل مناسبی به برنامه‌نویس می‌دهد، اما نیازمند به کار بردن صریح مکانیزم‌های جریان داده، در معرض گذاشتن روال‌های زبان C، سازهایی همچون سوکت‌ها و همچنین الگوریتم‌های سطح بالا

^۱ Grid Computing

^۲ High-Performance Computing

^۳ Application Program Interfaces

^۴ Message Passing Interface

^۵ Storage Area Network

برای تحلیل دارد. ایجاد هماهنگی در بین پردازشگرها هنگام محاسبات توزیع شده در مقیاس بزرگ یک چالش است. سخت‌ترین جنبه کار رسیدگی ساده به خرابی‌های جزئی (زمانی که مشخص نیست کدامین پردازش‌ها با خطا مواجه شده است) و هنگامی است که هنوز در حال پیشرفت در کلیات محاسبات کاری هستید. برنامه های MPI به طور صریح به مدیریت نقاط ذخیره و بازیابی می پردازند که موجب اختیارات بیشتری برای برنامه نویس می‌شود و از سوی دیگر نوشتن برنامه را برای آن‌ها سخت می‌کند.

محاسبات داوطلبانه^۱

این نوع مدل پردازشی را با معرفی یک پروژه‌ی موفق در این زمینه شرح خواهیم داد و آن پروژه چیزی جز SETI@home نیست. SETI به دنبال یافتن هوش فرا زمینی است، پروژه اجرایی SETI@home با استفاده از اختیار گذاشتن داوطلبانه زمان پردازنده کامپیوترهای دیگر در زمان بیکاری برای تحلیل داده‌ی امواج رادیویی به منظور یافتن نشانه‌هایی از زندگی هوشمند خارج از زمین استفاده می‌کند. SETI@home شناخته‌ترین پروژه محاسبات داوطلبانه از میان دهه‌ها پروژه مشابه است؛ به علاوه می‌توان به دو پروژه دیگر یعنی اعداد اول مرسن^۲ به کمک دنیای اینترنت برای یافتن اعداد اول بزرگ و Folding@home برای درک ترکیبات پروتئین و چگونگی رابطه آن‌ها با بیماری اشاره کرد.

پروژه های محاسبات داوطلبانه با شکستن مسائل به تکه‌های کوچک و ساده‌تری بر حل آن‌ها کار می‌کنند که واحدهای کاری نامیده می‌شوند و برای تحلیل به کامپیوترهای گوناگون یا فرستاده می‌شوند. برای مثال، واحد کاری SETI@home در حدود ۰.۳۵ مگابایت از داده‌های رادیویی است که ساعت‌ها یا روزها بر روی کامپیوترهای معمولی برای تحلیل کردن زمان نیاز دارد. زمانی که تحلیل کامل شد، نتایج به سرور ارسال می‌شوند و کلاینت واحد کاری دیگری دریافت می‌کند. به منظور احتیاط و جلوگیری از تقلب، هر واحد کاری برای سه ماشین مختلف ارسال می‌شوند و زمانی که حداقل دو نتیجه یکسان دریافت شود، آن نتیجه قابل قبول است.

^۱ Volunteer Computing

مسائل SETI@home خیلی به پردازنده حساس هستند هر کدام مناسب اجرا بر روی صدها یا هزاران کامپیوتر در جهان ساخته می شوند،^۱ هر چند در برخی مسائل زمان انتقال واحدهای کاری به مراتب بیشتر از زمان اجرای محاسبات بر روی آنها است و داوطلبان سیکل‌های پردازنده را اهدا می کنند، نه پهنای باند. از این رو SETI@home برای محاسبات همیشگی خود بر روی ماشین‌های غیر مطمئن در اینترنت با پهنای باند ارتباطی خیلی متغیر و داده‌های غیر محلی مناسب است.

محاسبات ابری

محاسبات ابری با محاسبات با کارایی بالا شباهت‌های زیادی دارد اما این دو در تعاریف بنیادی خود تفاوت‌های چشمگیر دارند که مهم‌ترین آنها مورد استفاده و معماری این دو سیستم توزیع شده با هم است. محاسبات با کارایی بالا بیشتر در پژوهش‌ها و کاربردهای درون سازمانی استفاده دارد و هزینه‌ی زیادی در بحث راه‌اندازی و نگهداری آنها اعمال می‌شود. در مقابل پردازش ابری با این فرضیه پدید آمده است که کاربران به جای ذخیره و حتی پردازش اطلاعات بر روی سیستم‌های شخصی، این موارد را بر روی مجموعه‌ای از سیستم‌ها انبار کنند که دسترسی کاربران به طور معمول از طریق شبکه‌ی اینترنت با آنها میسر است و به این صورت سهولت دسترسی به اطلاعات در هر زمان و هر مکان را ایجاد می‌کند و با ارائه مدل تجاری دریافت هزینه در برابر خدمات برای سازمان‌های فراهم کننده ابر درآمد زایی داشته باشد.

شکل گیری این نوع سیستم‌های پردازشی و ذخیره‌سازی به اوایل دهه ۹۰ بر می‌گردد و بدین شکل تعریف می‌شود: یک الگوی توزیع شده پردازشی در مقیاس بالا است که به ارائه مجازی‌سازی با قابلیت گسترش پویا، کنترل قدرت محاسباتی، بسترهای ذخیره‌سازی و دسترسی است که در صورت درخواست کاربران و دریافت هزینه مطابق این خدمات، آنها را به کاربر آن عرضه می‌کند.

^۱ در ژانویه سال ۲۰۰۸، طبق گزارشی پروژه SETI@home روزانه ۳۰۰ گیگابایت داده را با استفاده از ۳۲۰۰۰۰ کامپیوتر

پردازش می‌کند.