دانشکده مهندسی مکانیک

عنوان پایان نامه جهت دریافت درجهٔ کارشناسی ارشد
در رشتهٔ مهندسی مکانیک - تبدیل انرژی

تهیهٔ یک کد پردازش موازی
برای حل جریان سیال دو بعدی

آزاده نصیری

استاد راهنما : دکتر سید مصطفی حسینعلی پور

آبان 1384

تقدیر و تشکر

پس از خدا که به خاطرتمام نعمتهایش ، او را شاکرم ؛ بر خود لازم میدانم از کسانی که در انجام این پایان نامه مرا یاری کرده اند تشکر کنم.

از جناب آقای دکتر حسینعلی پور به علت راهنماییهای مفیدشان در انجام این تحقیق بسیار سپاسگزارم ، همچنین از پدر و مادرم که بهترین دوست و همراه من در همهٔ مراحل زندگیم بوده اند ، تشکر می کنم .

چکیده

این پروژه که موازی سازی یک کد با قابلیت حل عددی جریان آرام / آشفته،پایا/ناپایاو غیر قابل تراکم است،حرکتی در جهت پیشرفت و ارتقای کدهای موجود و استفاده از حد اکثر توانایی آنها ، میباشد.

موازی سازی کد به روش انتقال پیام با کتابخانهٔ MPI انجام و مسایلی جهت بررسی راندمان کد تولیدی اجرا شده است، که همگی شناخته شده ترین مسایل CFD به عنوان نمونه هایی در پدیده های مختلف هستند. در اجرای موازی علاوه بر رابطهٔ راندمان و تعداد پردازشگرهای فعال ، چگونگی تقسیم منطقه بر راندمان بررسی شد.

به طور کلی کاهش زمان و افزایش speed up با افزایش تعداد پردازشگرها کاملاً واضح و مطابق با پیش بینی است.در مسایلی تعداد بهینه ای برای پردازشگرها به دست آمد که تعداد بیشتر، منجر به کاهش speedup می شد. تفاوت رفتار جریان آرام و آشفته نیز نسبت به موازی سازی بررسی گردید.

این پروژه نشان می دهد که در تفکیک منطقه ، نه تنها اصل حداقل ارتباط، بلکه فیزیک مسأله نیز شدیداً باید مورد توجه قرار گیرد ، کما اینکه ممکن است ، فیزیک مسأله سرعت بیشتر را با تفکیک منطقه ای که ارتباط بیشتری دارد ، تحمیل کند.

فصل یک ، مقدمه ای کلی در رابطه با پردازش موازی و تاریخچهٔ آن و رویکرد امروزی آن ،و فصل دوم آشنایی با مبانی و اصطلاحات پردازش موازی و سخت افزار و نرم افزار هستند. در فصل سوم معادلات حاکم بر فیزیک مسایل سیالاتی و چگونگی گسسته سازی آنها در این کد ، آورده شده اند. فصل چهارم اختصاص یافته است به چگونگی موازی سازی و مسایل حل شده با کد تولید شده . فصل پنجم ، نتیجه گیری و پیشنهادات ارائه شده اند.

در نهایت مراجع اصلی این پایان نامه آورده شده اند.

فهرست

عنوان	صفحه
شکلها و منحنیها	III
فصل اول : مقدمه	1
فصل دوم : پردازش موازی چیست؟	۶
2-1- محاسبات موازی	۷
2-2-چرا پردازش موازی؟	۷
2-3-واژه شناسی	۱۱
2-4-سخت افزار	15
2-4-1- طبقه‌بندی (taxonomy).	۱۶
2-4-2- معماری کامپیوتر	۱۶
2-4-2-1- SIMD	۱۷
2-4-2-2- MIMD	۲۱
2-4-2-2-1- حافظه مشترک (Shared – Memory)	۲۲
2-4-2-2-2- حافظه توزیع شده (Distributed Memory)	۲۳
2-4-2-3- نوع Hybrid Memory Distributed – Shared	۲۵
2-5-نرم افزار	۲۶
2-5-1- الگوریتم‌های پردازش موازی	۲۷
فصل سوم : معادلات حاکمه و گسسته سازی	۳۰
3-1-معادلات حاکم بر سیال	۳۱
3-2- گسسته سازی معادلات	۳۳
فصل چهارم : چگونگی موازی سازی	۳۹
4-1- معرفی کد	۴۰
4-2-معرفی CLUSTER مورد استفاده	۴۱
4-3-تفکیک منطقه(Domain Decomposition)	۴۲
4-4- مسایل حل شده	۴۶
الف)جریان آرام در کانال	۴۶

ب)جریان آشفته در کانال ۴۷
ج)جریان آرام روی پله(Backward step) ۴۸
د)جریان آشفته روی پله ۵۰
ه)جریان داخل حفره(cavity) ۵۱
و)جریان داخل جعبه با مرز ناپایای سینوسی ۵۵
ز) خشک کن با بخار سوپرهیت ۵۶
ز-1- هیدرودینامیک تک جت ۵۷
ز-2- انتقال حرارت ۶۰
فصل پنجم :نتایج و پیشنهادات ۶۶
مراجع ۶۹

فهرست شکلها

عنوان **صفحه**

شکل۲-۱: نمودار شماتیک speed up بر حسب تعداد پردازشگرهای فعال 14
شکل2-۲: یک مدل ون نیومن 18
شکل۲-۳: مدل سخت افزاری برداری 19
شکل۲-۴: مدل SIMD 20
شکل۲-۵: مدل MIMD 22
شکل۲-۶: مدل shared memory 23
شکل۲-۷: مدل ایده آل 23
شکل۲-۸: مدل distributed memory 24
شکل۲-۹: مدل هیبرید 25
شکل۴-۱: تصویر cluster با node 8 41
شکل۴-۲: شکل نمادین چگونگی تفکیک منطقه با مرز مشترک 42
شکل ۴-۳: فلوچارت برنامه در حالت سریال 44
شکل ۴-۴: پروفیل سرعت در خروجی یک کانال با جریان آرام 43
شکل ۴-۵: منحنی speedup جریان آرام داخل کانال 47
شکل ۴-۶: پروفیل سرعت در خروجی کانال با جریان آشفته 48
شکل ۴-۷: speedup جریان آشفته در کانال 48
شکل ۴-۸: کانتور جریان آرام گذرنده از روی پلّه 49
شکل ۴-۹: speedupجریان آرام روی پلّه 50
شکل۴-۱۰: کانتور جریان آشفته روی پلّه 50
شکل۴-۱۱: speedup جریان آشفته روی پلّه 51
شکل۴-۱۲: کانتور جریان داخل حفره 52
شکل۴-۱۳: speedupجریان داخل حفره ،■Ghia و— این کد 52
شکل۴-۱۴: گرافیک سه نوع تفکیک منطقه در cavity 53
شکل۴-۱۵: مقایسهspeedup جریان داخل حفره در سه مدل تقسیم منطقه 54
شکل۴-۱۶: کانتور خط جریان در جعبه در زمان t = 0/07 55
شکل۴-۱۷: speed up حاصل از جریان داخل جعبه 56
شکل۴-۱۸: شکل شماتیک از تفکیک منطقهٔ impinging در راستای عمودی 57

شکل۴-۱۹: کانتور جریان یک تک جت پاششی 59
شکل۴-۲۰: منحنی فشار نسبی نسبت به فاصله از خط مرکزی جت 62
شکل۴-۲۱: منحنی Nu بر حسب فاصله از مرکز جت 63
شکل۴-۲۲: منحنی speedup تک جت پاششی 64

فصل اوّل

مقدمه

توجه به پردازش موازی به عنوان رویکردی تقریباً جدید در کد نویسی ، نه تنها به عنوان آسان کننده و سرعت دهنده ، بلکه به عنوان هدف سازندگان سخت افزار به عنوان معماری یک PC خانگی، بدون شک یک ضرورت است. مکانیک سیالات از شاخه هایی در علوم و فنون است که معادلات پیچیده و فیزیک بحث بر انگیز پدیده های تحت بررسی اش آن را شدیداً وابسته به محاسبات کامپیوتری و حلهای عددی حجیم و دقیق کرده است.

از آنجا که به طور کلی ، کمتر از نیم قرن از عمر پردازش موازی ، می گذرد،هنوز در تمام جهات ، قابل پیشرفت و تحقیق می باشد. در زمینهٔ سخت افزار با انواع معماریها روبرو میشویم و همچنان ، و البتّه متأثر از پیشرفت در سایر شاخه های سخت افزار ، در حال پیشرفت میباشد. در زمینهٔ نرم افزار نیز ، نه تنها بسیاری از کامپایلرها ، به صورت High Performance دوباره ویرایش شده اند، کتابخانه های بسیار کارآمد و مفید جهت اتصال به برنامهٔ سریال از قبل نوشته شده، طراحی و تولید شده اند. از دیدگاه برنامه نویسی ، به معنی استفاده از نرم افزار و سخت افزار موجود ، جهت تولید یک کد و یا نرم افزار کاربردی، می توان در بسیاری از شاخه های علوم پایه ، فنی و مهندسی ، هنری ، آمار و اقتصاد و شاخه های دیگر ، تحول و پیشرفت ناشی از به کار بردن این نوع تفکّر و فن آوری را مشاهده کرد. به طور خاص ، در مکانیک سیالات ، موازی سازی و تفکر موازی در برنامه نویسی ، تحول بزرگی در حل مسایل متعدد، خصوصاً مسایل grand challenging ، ایجاد کرده است. امروزه متخصصان CFD نه تنها به دنبال روشهایی بهینه در grid generation بلکه در تقسیم مناطق (Domain Decomposition) بین پردازشگرها هستند . به علاوه یکی از کارهایی که انجام می شود ، بهبود عملکرد کدهای کارآمد موجود با استفاده از موازی سازی است. دیدگاه موازی ، جسارت اندیشیدن به مسایل بزرگ ، حجیم و دقیق ، از جهات مختلف را به متخصصان این شاخه از علم داده است. از آنجا که تنوع مسایلی که به این روش برنامه نویسی می شوند و تنوع رویکردها نسبت به موازی سازی و چگونگی استفاده از این فن بسیار متنوع و متعدد است ، نمی توان مورد خاصی را جهت رویکرد امروز،

نام برد . از بزرگترین شاخه های سیالات که بیشترین بهره را از پردازش موازی برده اند ، شاخهٔ هوا-فضا است که چه در زمینهٔ آیرو دینامیک و طراحی انواع هواپیماها و موشکها و چه در زمینهٔ هواشناسی به واسطهٔ این فن آوری پیشرفتهای قابل ملاحظه ای داشته اند.

امروزه ، تحقیقات بسیاری شامل بررسی اثرات انواع متدهای CFD روی انواع سخت افزار ها و ابداع الگوریتمها و یا انواع کتابخانه ها برای اتصال و ارتقای برنامه های موجود می باشد . تحقیقهای Satofuta و همکارانش مبنی بر ارائه حلهای جدید برای معادلهٔ پواسون ، Garanzha و همکارانش با ارائهٔ متدهای تقریب نوع پاده مرتبهٔ بالا برای حل مسایل سه بعدی غیر قابل تراکم CFD از این جمله اند. از دیگر محققان در این زمینه در سال 2004 می توان به افراد زیر اشاره کرد.

I. Boursier, D. ؛ M. Soria, F. X. Trias, C. D. Perez-Segarra and A. Oliva F. X. Trias, M. Soria,O. ؛Tromeur-Dervout and Y. Vassilevsky –Aitken B.N. Chetverushkin, V.A. Gasilov, S.V. ؛ Lehmkuhl and C.D. Perez-Segarra A. Frullone and D. ؛ Polyakov, M.V. Iakobovski, E.L. Kartashera, I.V. Tromeur-Dervout Abalakin, I.V. , N.Yu Romanyukha, S.A. Sukov and A.S. .Minkin

در بسیاری از موارد تحقیقاتی و صنعتی در دینامیک سیالات ، کدهای موجود با کمک فن آوری موازی سازی ارتقای راندمان و یا گسترهٔ کاربرد یافته اند ، مانند شبیه سازی آیرودینامیکی نوعی جنگنده که توسط G.Shrewsbury و همکارانش ارائه شد و یا شبیه سازی جریان اطراف فضاپیماها که توسط Matsushima انجام شد .

به طور مثال می توان به تعدادی از این تحقیقات انجام شده در سال 2004 ، اشاره کرد .

جریانهای آشفته و آکوستیک و سیّالات غیر نیوتونی (A. V. Alexandrov, V. G. Y. Kim, S. H. Park, K. W. Cho and J. H. Kwon Bobkov، ؛J. Jimenez شبیه سازی جریان DNS،large eddy (J. M. McDonough and T. Yang، ؛؛G. Amati and F.

R. ؛، V. Ivannikov, S. Gaissaryan, A. Avetisyan and V. Padaryan؛،Massaioli FLOW K. آیرودینامیک و هیدرودینامیک ، (K. Matsuno ؛ ،Sentis and F. Duboc -. M. ؛ I. H. Tuncer and M. Kaya؛Chiba, S. Obayashi and K. Nakahashi Kremenetsky and T. Larsson L.Gonzalez, E.J. Whitney, K. Srinivas, J. K. Nakahashi and L-S. ؛ M. Yamakawa and K. Matsuno) شبکه بندی (Periaux I. A.) تشعشع ، (H. Tomita, M. Satoh and T. Nasuno؛ S. N. Boldyrev؛Kim B.N.) lattice-boltzman متد (Graur, T. A. Kudryashova and S. V. Polyakov X. ؛ G. Brenner and A. Al-Zoubi ؛ Chetverushkin and N. Yu. Romanyukha E. Laucoin and C. ؛ J. Gu, D. R. Emerson, D. Bradley and P. H. Gaskell ؛ B. N. Chetverushkin, N. G. Churbanova and M. A. Trapeznikova؛ Calvin K. Shimano, K. Okudera, T. Anaguchi, N. Utsumi, M. Saito, ؛L. A. Barba D. ؛ J. Sahu and K. R. Heavey) کنترل جریان ، (C. Sumie and Y. Enomoto (Tromeur-Dervout

شاخۀ اکولوژی و هواشناسی از این فنآوری بهرۀ بسیار برده است. A.Dubey و همکارانش در آزمایشگاه آرگون و به سفارش ناسا ، کد موجود برای حل جریان گاز تراکم پذیر با انتقال حرارت هدایتی را ارتقا داده و آن را با استفاده از این فنآوری تبدیل به کدی برای حل جریانهای همرفتی بسیار آشفته اطراف ستارگان کردند. از دیگر فعالان این شاخه در سال 2004 می توان از افراد زیر نام برد .

B. Galvan, B. ؛ M. Tsugawa, Y. Tanaka, M. Sakashita and M. Kogi M. J. Castro, J. A. ؛ Gonzalez, A. Padron, H. Carmona and G. Winter C. ؛ M. Zijlema؛ Vida and C. Pares ؛ Garcia-Rodriguez, J. M. Gonzalez A. I. Sukhinov, A. A. ؛ C. Levi Ciortan, C. Guedes Soares, J. Wanderley A. I. Sukhinov and A. A. Sukhinov؛ Sukhinov and S.V. Kirilchik

در خصوص آزمایشگاههای فعال در پردازش موازی ، در شاخۀ مکانیک سیّالات ، لازم است اشاره شود که امروزه تقریباً تمام دانشکده های فعال و بنام در این شاخۀ مکانیک سیالات در سطح جهان ، از

امکانات پردازش موازی بهره مندند و مراکز تحقیقاتی و آزمایشگاههایی که قبلاً امکانات سوپرکامپیوترها را داشته اند ، به صورت هیبرید از امکانات پردازش موازی بهره میبرند. به طور مثال از آزمایشگاههای زیر نام برد:

آزمایشگاه ملّی آرگون ؛آزمایشگاه CFDدر دانشگاههای مهندسی Purdue ، کلرادو ,UCB و Minnesota، UCD ، Illinois، UCLA ، مرکز ملی آمریکا در تحقیقات اتمسفریک؛آزمایشگاه ملّی هوا-فضا در توکیو ؛ انستیتو تحقیقاتی انرژی اتمی ژاپن در توکیو ؛ مرکز تحقیقاتی محاسبات پیشرفته در صنایع دفاع و مهندسی هوا-فضا ، دانشگاه مینسوتا ، آمریکا ؛ آزمایشگاه ملی Oak Ridge ؛ (GMD) مرکز تحقیقاتی ملی آلمان در IT ، سنت آگوستین ، آلمان ؛ انستیتو تحقیقات پتروشیمی فرانسه ؛ انستیتو تحقیقات نیروگاه و مهندسی فرآیند در دانشگاه اشتوتگارت ، مرکز پروازهای فضایی گودارد ناسا ، مرکز تحقیقاتی شیفبا ، دانشگاه هامبورگ . در ایران این فن آوری بسیار جدید است و از عمر آن در دانشگاههای ایران بیش از سه سال نمی گذرد و در حال حاضر دانشکده های هوا-فضا و مکانیک دانشگاه صنعتی شریف ، دانشکدهٔ مکانیک دانشگاه شیراز و دانشکدهٔ مکانیک دانشگاه علم و صنعت دارای این فن آوری هستند. تمرکز پروژه های انجام شده بر ارتقای کدهای موجود در آزمایشگاهها بوده است. به طور مثال ارتقای کد های die-cast ، حل اجزاء محدود تزریق پلاستیک؛ البته با تکنولوژی روز دنیا فاصلهٔ زیادی دارد ، زیرا تعداد پردازشگرهای موازی در آزمایشگاههای ایران کمتر از پنجاه عدد است ؛ در حالیکه بسیاری از آزمایشگاههای پیشرفتهٔ دنیا با چند صد پردازشگر ، یا سیستم هیبرید ، فعالیت میکنند.

فصل دوم

پردازش موازی چیست؟

موازی سازی یک استراتژی برای انجام کارهای پیچیده و با حجم زیاد است. برای حصول به نتیجه با سرعت و دقت بیشتر. در حقیقت هر عملیاتی را می‌توان به بخشهای کوچکتر تقسیم کرد، که این بخشها می‌توانند پی‌درپی انجام شوند و یا می‌توان این بخشها را بین واحدهای عملیاتی به گونه‌ای تقسیم کرد که در هر زمان بیش از یک بخش انجام شود. در رویکرد دوم که در حقیقت همان موازی سازی است، روند به صورت تقسیم کار به بخشهای کوچکتر، اختصاص دادن این وظایف به چند واحد عملیاتی (علمگر) و هماهنگی و ارتباط بین آنها تا حصول نتیجه است. استفاده از این استراتژی در زندگی روزمره بسیار معمول است، ساخت ساختمانها ، ادارهٔ یک مؤسسه و سازمان بزرگ، تولید دستگاهها و مواد مصرفی مختلف. با کمی دقت متوجه می‌شویم که در هر کدام از پروژه‌های فوق در آنِ واحد چند نفر برای رسیدن به هدف مطلوب در حال انجام وظیفهٔ مخصوص به خود هستند و هماهنگی صحیح بین آنها تا رسیدن به نتیجه ، انجام می‌گیرد.

2-1- محاسبات موازی

به طور خاص و آنچه در اینجا منظور ماست، پردازش موازی است. پردازش موازی استفاده از بیش از یک واحد پردازش مرکزی (CPU) در یک زمان ، برای حل یک مسأله است. امروزه، طیف گسترده‌ای از ماشینها در زمرهٔ سیستم‌های پردازش موازی قرار می‌گیرند، از دو کامپیوتر شخصی (PC) که از طریق یک شبکهٔ داخلی به هم وصلند تا هزاران پردازشگر قوی در یک سوپر کامپیوتر.

2-2-چرا پردازش موازی؟

همان‌گونه که قبلاً نیز گفته شد، نیاز به افزایش سرعت و دقت ، علت کلی رویکرد به موازی سازی است. به طور خاص در شاخهٔ کامپیوتری و محاسباتی آن می‌توان به نکات زیر اشاره کرد:

تصور کنید که نیاز به افزایش سرعت معادلات حجیم داریم، مثل حل مسایل هواشناسی، اما افزایش

سرعت در یک کامپیوتر سری دارای محدودیت است. حد هدایت سیم مسی برابر $9\times10^5 m/s$ و حد سرعت نور $3\times10^9 m/s$ است.

امروزه با مسایلی مواجه هستیم که کامپیوترهای غیرموازی توان انجام آن را ندارند، مسایلی مانند:

- بسیاری از مسایل شبیه سازی و مدل‌سازی به علت نیاز به دقت
- مسایلی که براساس تعداد بسیار زیادی از اطلاعات محاسبه شوند. مانند پردازش تصویر، زلزله‌شناسی و کاوش اطلاعات
- مسایل Grand Challenging مثل مدل‌سازی هواشناسی، ژنوم انسان، انتشار آلودگی کرودینامیک کوانتومی و گردش اقیانوسها

در این مسایل مقیاس محاسبات «ترافلاپ» است. فلاپ به یک عمل محاسباتی در هر ثانیه گفته می‌شود وترا یعنی 10^{12}. عمل محاسباتی به معنی متوسط عمل جمع و ضرب است. (تقسیم که زمان بیشتری می‌گیرد، در محاسبات فلاپ در نظر گرفته نمی‌شود. در معماریهای امروزی زمان مصرفی جمع و ضرب برابر است.)اما مشکل تنها به سرعت و حجم زیاد محاسبات ختم نمی‌شود. اگر کامپیوتری ساخته شود که بتواند تریلیون محاسبه در ثانیه انجام دهد ولی به اندازهٔ چند میلیون کلمه حافظه گنجایش داشته باشد، سودی نخواهد داشت. پس حجم بالای محاسبات، سرعت و حجم حافظه بالا را نیز نیازمند است. حال تصور کنید که بخواهیم کامپیوتری با این توان یعنی توان یک تریلیون عمل محاسبه در هر ثانیه بسازیم. شاید واضح‌ترین دیدگاه به طور ساده توسعهٔ فناوری‌های موجود است. یعنی سعی کنیم یک کامپیوتر ون نیومن کم و بیش متناسب با پردازشگر بی‌نهایت سریع و حجم فوق‌العاده بالای حافظه بسازیم. اگر این کامپیوتر بخواهد کد زیر را اجرا کند:

```
/* x,y and z are arrays of floats, each containing one*/
/* trillion entries*/
```

```
for (i=0 ; i<ONE-TRILLION ; i++)
    Z[i] = x [i] + y[i]
```

روند کار به طور معمول به این صورت است که y[i] و x[i] از حافظه گرفته شده، جمع می‌شوند و نتیجه در z[i] ذخیره می‌شود. پس، برای اجرای این کد، حداقل 3×10^{12} کپی بین حافظه و محل ثبت در هر ثانیه باید انجام شود. اگر داده (data) از حافظه به CPU با سرعت نور $(3\times10^{8}m/s)$ حرکت کند، و اگر r، فاصلهٔ متوسط یک کلمهٔ حافظه از واحد پردازش مرکزی (CPU) باشد، آنگاه r باید معادلهٔ زیررا ارضا کند:

$$3\times10^{12}\times r(m) = 3\times10^{8}(m/s)\times1(s)$$

یعنی $r=10^{-4}(m)$

کامپیوتر سریع ما باید حداقل سه تریلیون کلمه حافظه گنجایش داشته باشد، تا آرایه‌های z,y,x را ذخیره کند. سخت‌افزار معمول شبکهٔ حافظه، به صورت مستطیل طراحی شده‌اند. اگر یک شبکهٔ مستطیلی با طول S در نظر گرفته شود و CPU در مرکز آن قرار داشته باشد، آنگاه فاصلهٔ متوسط از یک مکان حافظه تاCPU حدود S/2 است . پس می‌خواهیم$S/2=r=10^{-4}(m)$ باشد، یا $S=2\times10^{-4}(m)$. اگر کلمات حافظه یک شبکهٔ مربعی را شکل دهند، یک ردیف کلمه حافظه شامل $\sqrt{3\times10^{12}} = \sqrt{3}\times10^{6}$ کلمه خواهد بود. پس ما یک کلمه حافظه را باید داخل یک مربع با طول $\frac{2\times10^{-4}}{\sqrt{3}\times10^{-6}} \approx 10^{-10}(m)$ جا بدهیم. این مقدار اندازه یک اتم کوچک است! به عبارت دیگر، اگر یک کلمه یعنی 32 بیت (یا معمول‌تر 64 بیت) را با یک اتم نشان دهیم، متوجه می‌شویم که ساخت این کامپیوتر در حال حاضر غیرممکن است. البته از نظر اقتصادی نیز با امکانات امروز، ساخت ترانزیستورهاو قطعاتی تا آن حد ریز مقرون به صرفه نیست . پس چگونه باید پاسخ صورت مسأله اصلی حل مسایل پیچیده و حجیم با سرعت بالا را داد؟ پاسخ بدیهی است. همان گونه که در زندگی روزمره وقتی حجم کار از توان یک نفر خارج می‌شود، چند نفر به کار گرفته می‌شوند، در اینجا باید چند پردازشگر و واحد حافظه به

کار گرفته شود. البته پیشرفت و سرعت شبکه‌ها و صنایع سخت‌افزار کامپیوتر نیز در رسیدن به این ایده بسیار موثر بوده است و قطعاً آیندهٔ محاسبات، محاسبات موازی است.

پس دلایل به طور عمده عبارتند از:

۱- صرفه‌جویی زمانی

۲- حل مسایل بزرگ و پیچیده

۳- استفاده از امتیاز منابع غیرمحلی، با استفاده از محاسبه روی شبکه‌ای از کامپیوترها یا حتی شبکهٔ جهانی ، وقتی منابع محلی محدود است.

٤- صرفه‌جویی در هزینه(حتی شامل خرید چند منبع محاسباتی ارزان‌تر به جای خرید یک سوپر کامپیوتر می‌شود.)

٥- غلبه بر محدودیت حافظه

البته راه حل به اینجا ختم نمی‌شود. پس از تصمیم‌گیری برای به کارگیری چند پردازشگر و حافظه , هنوز حجم زیادی از عملیات باید انجام شود:

۱- طراحی و اجرای یک شبکه ارتباطی بین پردازشگرها و واحدهای حافظه

۲- طراحی و اجرای یک سیستم نرم‌افزار برای این سخت‌افزار

۳- نوشتن الگوریتم و دستورالعملهای اطلاعاتی برای حل مسأله

٤- تقسیم الگوریتم و دستورالعملهای اطلاعاتی به زیر مسأله‌ها و بخشهای کوچکتر.

٥- مشخص کردن ارتباطاتی که بین این بخشهای کوچک نیاز است.

٦- تخصیص دادن این زیر مسأله‌ها و بخشهای کوچک به پردازشگرها و واحدهای حافظه

۷- مدیریت و هماهنگی در زمان انجام مسأله

۸- جمع‌آوری نتایج و ارائهٔ آن

و آنچه گفته شد روندی است که باید طی شود و مسایلی است که باید حل شوند تا از محاسبات سری به محاسبات موازی ،که نیاز حتمی است، رسید .

2-3- واژه شناسی (Terminology)

در این بخش اصطلاحاتی را معرفی می‌کنیم که در فرهنگ پردازش موازی بسیار به کار می‌روند:

۱) Communication

وظایف موازی شده، عموماً نیاز به تبادل اطلاعات دارند. راههای متعددی در این جهت وجود دارد. مانند به اشتراک گذاشتن حافظه، از طریق انواع شبکه و به طورکلی باید گفت، آنچه در انتقال اطلاعات مهم است همین مورد Communication یا ارتباط است که سایر شاخه‌های موازی سازی را تحت‌الشعاع قرار می‌دهد.

۲) Synchranization

یعنی برنامه‌نویسی به گونه‌ای باشد که در نقطه‌ای یک پردازش متوقف و منتظر می‌شود تا پردازش دیگر به همان قسمت برنامه برسد، یا به او اطلاعاتی بدهد و یا اطلاعاتی بگیرد. این عمل معمولاً زمان اجرای برنامه را افزایش می‌دهد، اما در جهت صحت اجرای برنامه گاهی لازم می‌باشد.

۳) Granularity

در محاسبات موازی، مفهومی کیفی است که به معنی نسبت محاسبات به ارتباطات است. دو نوع Fine و Coarse دارد.

٤) Observed Speed up

این پارامتر در یک کد به صورت:

زمان ظاهری اجرای برنامه به صورت سری/ زمان ظاهری اجرای برنامه به صورت موازی

(Wallclock time of serial execution/Wallclock time of parallel execution)

یکی از ساده‌ترین و پر استفاده‌ترین مشخصه‌ها برای تعیین بازده یک برنامهٔ موازی است.

Speed up ،به طورکلی، به نوعی بازده برنامه موازی را نشان می‌دهد و چگونگی محاسبهٔ آن از موارد مورد بحث است. به طور معمول به صورت S(p)=Ts/Tp محاسبه می‌شود که زمان می‌تواند از نظر کامپیوتری تعریف‌های مختلفی داشته باشد.

۵) Parallel Overhead

هزینه‌ای که برای هماهنگی و اجرای وظایف موازی مصرف می‌شود شامل:

- زمان شروع task startup time
- هماهنگی زمانی Synchronization
- ارتباطات و تبادل مقادیر Data Communication
- هزینهٔ تحمیلی ناشی از کامپایلر، کتابخانه و ابزار موازی
- پایان اجرای وظیفه (task)

پس می‌توان گفت منبع Overhead، ارتباطات، تأخیرهای زمانی، انتظار و محاسبات اضافی ناشی از موازی شدن است.

۶) Massively Parallel

از دیدگاه سخت‌افزاری شامل چند پردازشگر می‌شود .کلمه چند مقادیر مختلفی به خود می‌گیرد ولی در اینجا به معنی بیش از 1000 است.

۷) Latency

زمانی است که صرف می‌شود تا حداقل پیام (صفر بایت) از نقطهٔ A به نقطهٔ B برود، معمولاً با میکروثانیه گفته می‌شود.

۸) Band width

مقداری از اطلاعات که در واحد زمان قابل انتقال است. معمولاً به صورت مگابایت در ثانیه توضیح داده می‌شود. ارسال تعداد زیاد پیام کوتاه باعث می‌شود که افزایش latency هزینه را بالا ببرد. گاهی بهتر است، پیامهای کوچک بسته‌بندی شده و به صورت یک پیام بزرگتر ارسال گردد.

۹) وضوح ارتباطات (Visibility of Communication)

- با مدل انتقال پیام، ارتباطات صریح و معمولاً بسیار واضح و تحت کنترل برنامه نویس هستند.
- با مدل داده موازی، ارتباطات در برخی موارد برای برنامه نویس شفاف است. در مواردی روی معماریهای حافظهٔ توزیع شده، برنامه نویس حتی نمی‌تواند بفهمد که دقیقاً چقدر ارتباط بین پردازشگرها انجام شده است.

12) قانون آمدال(Amdahl's law)

اگر T_s زمان اجرای برنامه روی یک تک پردازشگر و r نسبتی از برنامه که قابلیت موازی شدن دارد و P تعداد پردازشگرهای مورد استفاده باشند، زمان اجرای قسمت موازی برنامه $r\frac{Ts}{p}$ و قسمت غیرموازی (1-r)Ts است که وابستگی به P ندارد. پس Speed up به دست می‌آید:

$$S(p)=\frac{Ts}{(1-r)Ts+rTs/p}=\frac{1}{1-r+r/p} \qquad (1)$$
$$p\to\infty \qquad \Rightarrow S(p)=\frac{1}{1-r}$$

در نتیجه برای Speed up از نظر تئوری محدودیت از بالا تعریف می‌شود. این قانون آمدال است. البته می‌دانیم که به دلیل اتلافات ناشی از ارتباطات نیز حد بالای Speed up محدود می‌شود.

13) process (پردازش)

پردازش یک بلوک پایه و اساسی در بیشتر محاسبات موازی است. یک پردازش، نمونه‌ای از یک برنامه یا زیربرنامه است که کم و بیش به صورت مستقل روی یک پردازشگر اجرا می‌شود. در حقیقت یک برنامهٔ پردازش موازی، اجرای برنامه در هر لحظه با بیش از یک پردازش فعال، است. پس قابلیت برنامه, تحت تأثیر چگونگی تخصیص اطلاعات به پردازش‌ها، تولید و هماهنگی پردازش‌ها می‌باشد.

14) راندمان موازی (Parallel Efficiency)