

تکنولوژی محاسبات توزیع شده، مفاهیم و کاربردهای آن به همراه R-9، برنامه‌ای جهت ترسیم طیف‌های پاسخ به کمک این تکنولوژی

محمد تبرکی

دانشجوی کارشناسی مهندسی عمران، دانشکده مهندسی، دانشگاه خلیج فارس، بوشهر
mohammad@tabarroki.com

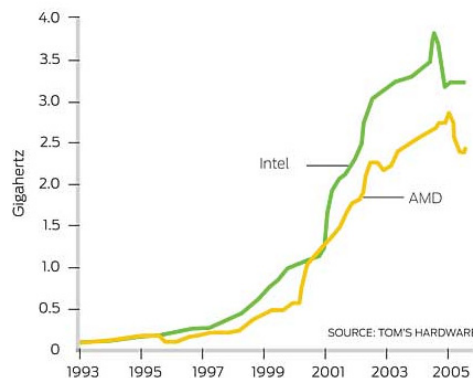
خلاصه

بسیاری از محاسبات علمی، به مقدار قابل ملاحظه‌ای زمان برای پردازش نیاز دارند. که انجام آنها توسط کامپیوترهای شخصی، امری بسیار طولانی و در برخی موارد عملاً غیرممکن است. استفاده از ابرکامپیوترها نیز به دلیل هزینه‌ی سرسام‌آور آنها، برای هر گروه تحقیقاتی امکان‌پذیر نیست. راه‌حل، تکنولوژی محاسبات توزیع شده می‌باشد که در طی آن بار محاسباتی بین چندین کامپیوتر که از طریق شبکه به هم متصل شده‌اند، توزیع می‌گردد. در این مقاله، مفاهیم و کاربردهای محاسبات توزیع شده بررسی می‌شود و سپس به عنوان نمونه، برنامه‌ای جهت ترسیم طیف‌های پاسخ به کمک این تکنولوژی نوشته شده است. که زمان آنالیز آن توسط یک کامپیوتر ۱۴۵۱ ثانیه و هنگام توزیع بین هشت کامپیوتر، به ۲۰۷ ثانیه کاهش می‌یابد.

کلمات کلیدی: محاسبات توزیع شده، طیف‌های پاسخ

مقدمه

بسیاری از محاسبات علمی، به مقدار قابل ملاحظه‌ای زمان برای پردازش نیاز دارند. هر چند که طبق قانون مور [1] سرعت پردازنده‌ها هر دو سال یکبار بدون افزایش چشم‌گیر قیمت، دو برابر می‌شود ولیکن همزمان با آن پارامترهای بیشتری نیز هر ساله در محاسبات وارد شده و نتیجه اینکه سرعت پردازنده‌های هر دوره نسبت به مسائل دوره‌ی خود کافی به نظر نمی‌رسد. در شکل ۱، نمودار فرکانس ساعت پردازنده‌های تولید شده توسط دو شرکت Intel و AMD در طول سال‌های ۱۹۹۳ تا ۲۰۰۵ ملاحظه می‌شود. البته علت کاهش فرکانس پردازنده‌ها در انتهای نمودار، گرمای بیش از حد تولید شده و همچنین افزایش میزان برق مصرفی در پردازنده‌های با فرکانس بالا می‌باشد. لذا طبق گزارش Intel [2] با ۲۰ درصد کاهش در فرکانس ساعت پردازنده، می‌توان ۵۰ درصد در میزان برق مصرفی صرفه جویی کرد و این در حالی است که فقط ۱۳ درصد از کارایی پردازنده کاهش می‌یابد. بر همین اساس با کنار هم قرار دادن دو پردازنده که هر کدام ۸۰ درصد فرکانس ساعت حالت اولیه خود را دارند، می‌توان ۷۳ درصد کارایی را افزایش داد، در صورتی که میزان برق مصرفی آنها نیز معادل حالتی است که تنها یک پردازنده با ۱۰۰ درصد فرکانس ساعت خود، مشغول به پردازش باشد. ضمن اینکه گرما به جای متمرکز شدن در یک نقطه، از دو نقطه‌ی مختلف توزیع و منتشر خواهد شد و همه‌ی این‌ها، مویذ پا برجا بودن قانون مور (البته به شکل غیر سنتی آن) پس از گذشت نیم دهه است.



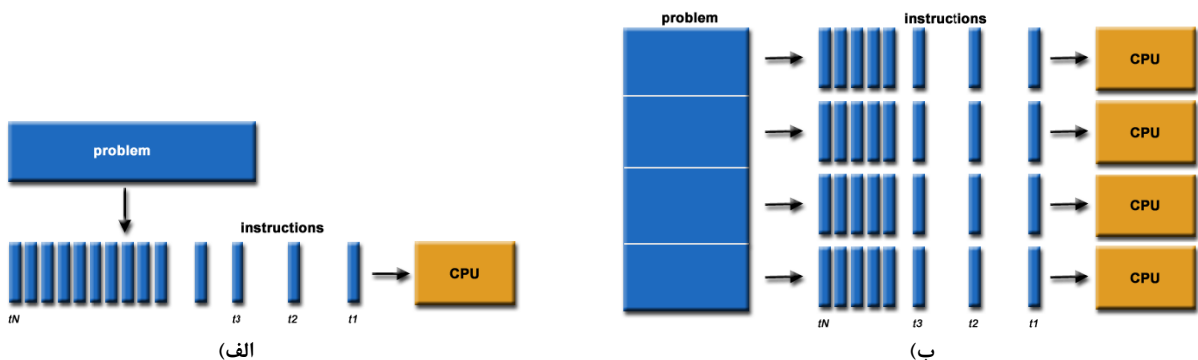
شکل ۱ - فرکانس ساعت پردازنده‌های Intel و AMD در طول سال‌های ۱۹۹۳ تا ۲۰۰۵

با همهی این تفاسیر، همانطور که در ابتدا نیز گفته شد، هر ساله در علوم مختلف مسائل جدیدی مطرح می‌گردد که بی‌شک اکثر آنها به انجام محاسبات زمانبر ختم می‌شوند و انجام این محاسبات با کامپیوترهای شخصی، امری بسیار طولانی و در برخی موارد عملاً غیرممکن است. برای غلبه بر این مشکل دو راه حل وجود دارد. راه حل اول، استفاده از کامپیوترهای بسیار قدرتمند، مانند ابرکامپیوترها می‌باشد که شدیداً هزینه‌ی بالایی دارند و هر موسسه یا گروه تحقیقاتی نمی‌تواند آن را تهیه کند. برای مثال، ابرکامپیوتر ایران که به سفارش سازمان هواشناسی کشور توسط محققین «مرکز تحقیقات پردازش‌های فوق سریع دانشگاه امیرکبیر» در سال ۱۳۸۶ ساخته شد، ۵۰۰ میلیون تومان هزینه در بر داشته است. (این ابرکامپیوتر دارای توان پردازشی ۸۶۰ گیگافلاپس - ۸۶۰ میلیارد عملیات اعشاری در ثانیه - می‌باشد)

راه حل دوم، استفاده از تکنولوژی محاسبات توزیع شده (Distributed Computing) می‌باشد. این روش به دلیل کارایی بالا و هزینه‌ی کمی که دارد، با استقبال محققین در علوم مختلف مواجه شده است. در زمینه‌ی مهندسی عمران نیز کارهای بسیار زیادی بر اساس همین تکنولوژی صورت گرفته که برای مثال می‌توان به این موارد اشاره نمود: آنالیز المان محدود و بهینه سازی طراحی در یک محیط محاسبات توزیع شده [3]، آنالیز آئروالاستیک و میزان حساسیت سرعت لرزش پل‌های معلق با دهانه بزرگ با استفاده از محاسبات توزیع شده [4]، شبیه سازی ترافیک در مقیاس بزرگ از طریق محاسبات توزیع شده paramics [5]، آنالیز المان محدود توزیع شده در کامپوزیت‌های لایه‌ای [6].

محاسبات توزیع شده

قبل از تعریف محاسبات توزیع شده، ابتدا لازم است که اجرای سری و اجرای موازی از هم تمییز داده شوند. به طور معمول، یک مسئله به مجموعه‌ای از دستورالعمل‌ها تقسیم می‌شود (مثلاً مسئله‌ی مساحت دوزنقه، به سه دستورالعمل: ۱- جمع دو قاعده‌ی آن ۲- تقسیم عدد بدست آمده بر دو ۳- ضرب آن در ارتفاع، تقسیم می‌گردد) سپس این دستورالعمل‌ها به ترتیب و به طور متوالی در پردازنده مورد پردازش قرار می‌گیرند، طوری که در هر لحظه تنها یک دستورالعمل توسط پردازنده در حال پردازش است. به این نوع پردازش، اجرای سری می‌گویند (شکل ۲ - الف). حال اگر بتوان همان مسئله را به چندین بخش تقسیم کرد، به طوری که حل همزمان این بخش‌ها با هم امکان‌پذیر باشد. آنگاه هر بخش را به پردازنده‌ی جداگانه‌ای سپرده و مطابق شکل ۲ - ب، دستورالعمل‌های هر بخش به طور همزمان بر روی پردازنده‌های مختلف مورد پردازش قرار می‌گیرند. به این نوع پردازش، اجرای موازی می‌گویند.



شکل ۲- الف) اجرای سری ب) اجرای موازی

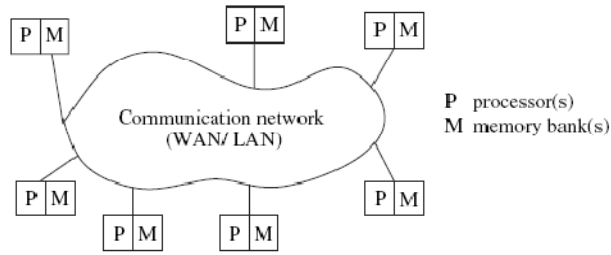
به عنوان ساده‌ترین مثال، می‌توان به ضرب ماتریسی اشاره نمود. در حالت اجرای سری، هر کدام از درایه‌های ماتریس حاصله، به ترتیب یکی پس از دیگری از حاصلضرب سطر و ستون‌های متناظر ماتریس‌ها بدست می‌آیند. اما همانطور که در شکل ۳ نیز ملاحظه می‌شود، محاسبه‌ی هر کدام از این درایه‌ها، هیچ‌گونه وابستگی به محاسبه‌ی دیگر درایه‌های ماتریس حاصله ندارد و لذا می‌توان محاسبه‌ی هر کدام از آنها را به یک پردازنده‌ی جداگانه‌ای سپرد (البته در این مثال هر چقدر ابعاد ماتریس بزرگتر باشد، مزیت اجرای موازی نمایان‌تر می‌گردد).

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 \\ 5 & 6 & 7 & 8 \\ 10 & 11 & 12 & 10 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix} = \begin{pmatrix} 70 & 80 & 90 \\ 158 & 184 & 210 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 \\ 5 & 6 & 7 & 8 \\ 10 & 11 & 12 & 10 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix} = \begin{pmatrix} 70 & 80 & 90 \\ 158 & 184 & 210 \end{pmatrix}$$

شکل ۳ - عدم وابستگی محاسبه‌ی درایه‌ها به همدیگر در ضرب ماتریسی

طبق تعریف [7]، محاسبات توزیع شده به دسته‌ای از محاسبات کامپیوتری اطلاق می‌شود که در آن دو یا چند کامپیوتر مستقل، از طریق شبکه‌ی کامپیوتری به هم متصل شده و با استفاده از یک الگوریتم موازی بار محاسباتی بین آنها توزیع می‌گردد. در شکل ۴، تصویری شماتیک از یک محیط محاسبات توزیع شده ملاحظه می‌شود. در این شکل هر کامپیوتر مستقل، دارای پردازنده (P) و حافظه‌ی (M) مخصوص به خود است. شبکه‌ای که این

کامپیوترها را به هم متصل می‌کند، می‌تواند در مقیاس کوچک مانند یک شبکه‌ی محلی (LAN) و یا در مقیاس بزرگ (WAN) و یا اینترنت باشد.



شکل ۴ - تصویری شماتیک از یک محیط محاسبات توزیع شده

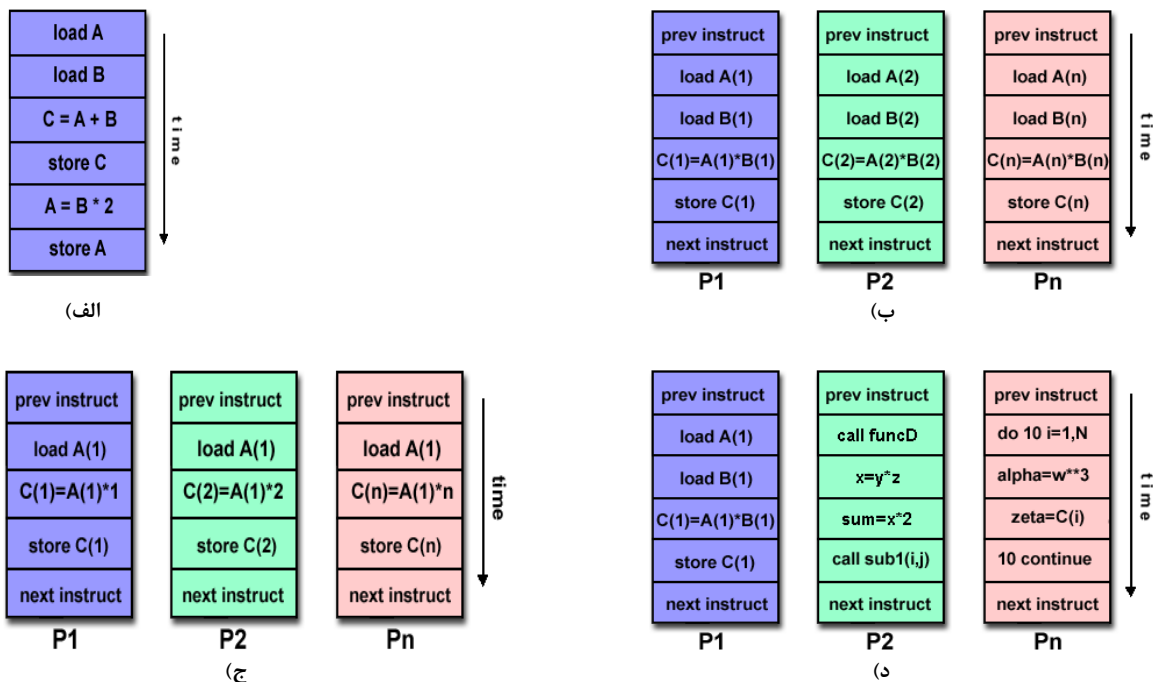
البته در همه‌ی موارد نمی‌توان از محاسبات توزیع شده استفاده نمود، بلکه مسئله‌ای که قرار است از طریق این تکنولوژی حل گردد، باید قابلیت تقسیم شدن به بخش‌های کوچکتر و مجزا را داشته باشد. آشنا شدن با طبقه‌بندی Flynn می‌تواند راهنمای خوبی برای بررسی امکان استفاده از محاسبات توزیع شده در حل یک مسئله باشد. Flynn [8]، ساختار برنامه‌های کامپیوتری را بر اساس دو فاکتور دستورالعمل (Instruction) و داده (Data) طبقه‌بندی کرد. که هر کدام از این دو فاکتور می‌تواند دارای وضعیت تک (Single) یا چندگانه (Multiple) باشد. لذا چهار حالت ممکن طبقه‌بندی به این شرح است: ۱- SISD، ۲- SIMD، ۳- MISD و ۴- MIMD (شکل ۵)

در حالت SISD (تک دستورالعمل، تک داده - Single Instruction, Single Data)، تنها یک جریان داده به عنوان ورودی برنامه استفاده شده و همچنین تنها یک جریان دستورالعمل توسط پردازنده اجرا می‌شود (شکل ۵ - الف). منظور از جریان داده یا جریان دستورالعمل، مجموعه‌ای از داده‌ها یا دستورالعمل‌ها می‌باشد که فقط و فقط می‌بایست به ترتیب و پشت سر هم اجرا شوند.

در حالت SIMD (تک دستورالعمل، چندین داده - Single Instruction, Multiple Data)، تمامی پردازنده‌ها یک جریان دستورالعمل را اجرا می‌کنند، اما برای هر کدام از پردازنده‌ها یک جریان داده‌ی مجزایی به عنوان ورودی استفاده می‌شود (شکل ۵ - ب).

در حالت MISD (چندین دستورالعمل، تک داده - Multiple Instruction, Single Data)، تنها یک جریان داده به عنوان ورودی برنامه استفاده می‌شود. اما هر کدام از پردازنده‌ها یک جریان دستورالعمل مجزایی را بر روی این جریان داده اجرا می‌کنند (شکل ۵ - ج).

در حالت MIMD (چندین دستورالعمل، چندین داده - Multiple Instruction, Multiple Data)، چندین جریان داده به عنوان ورودی برنامه استفاده می‌شود و هر کدام از پردازنده‌ها یک جریان دستورالعمل مجزایی را بر روی جریان داده‌ی خاص خود اجرا می‌کنند (شکل ۵ - د).



شکل ۵ - طبقه‌بندی Flynn (الف) SISD (ب) SIMD (ج) MISD (د) MIMD

استفاده از محاسبات توزیع شده در برنامه‌هایی که ساختار آنها در یکی از حالت‌های SIMD، MISD و MIMD قرار گیرد، امکان پذیر است. اما برای حالت SISD این امکان وجود ندارد. البته چنانچه بتوان برنامه را طوری تغییر داد که از حالت SISD به دیگر حالت‌ها تبدیل شود، آنگاه برای آن برنامه نیز امکان استفاده از محاسبات توزیع شده وجود دارد. برای مثال، روش‌های رایج در بدست آوردن معکوس ماتریس به گونه‌ای بودند که در حالت SISD قرار می‌گرفتند، اما با پدید آمدن تکنولوژی محاسبات توزیع شده و آشنا شدن با فواید آن، برای بهره‌مندی از این تکنولوژی روش‌هایی برای بدست آوردن معکوس ماتریس ارائه شد که آن را از حالت SISD به دیگر حالت‌ها تبدیل می‌کند. به عنوان نمونه می‌توان به [9] و [10] اشاره نمود.

کاربردهای محاسبات توزیع شده

به طور خلاصه می‌توان گفت در هر زمینه‌ای که به محاسبات زمانبر نیاز باشد، کاربرد محاسبات توزیع شده یک امر بدیهی به نظر می‌رسد. در ادامه به برخی از این زمینه‌ها که در مقیاس بزرگ کار شده‌اند، اشاره شده است:

- رمزگشایی
- اقتصاد، پیش‌بینی‌های مالی
- گرافیک و بصری، رندر کردن تصاویر
- فیزیک، شبیه سازی‌ها، صحت سنجی نظریات
- شیمی و بیولوژی، یافتن درمان برای ایدز، سرطان، آلزایمر
- جغرافی، پیش‌بینی وضعیت هوا، شبیه سازی حرکت‌های زمین

بزرگ‌ترین پروژه‌ای که با استفاده از محاسبات توزیع شده در حال انجام شدن است، پروژه Folding@home [11] وابسته به دپارتمان شیمی دانشگاه استنفورد می‌باشد. این پروژه که بر روی شبکه‌ی اینترنت کار می‌کند، به منظور شناسایی دقیق ساختار پروتئین جهت یافتن راهی برای درمان بیماری‌هایی از قبیل آلزایمر، سرطان، پارکینسون و غیره می‌باشد. در واقع کاربران به طور داوطلبانه از سراسر دنیا با دانلود نرم‌افزار مربوطه و اجرای آن بر روی کامپیوتر، اوقات بیکاری پردازنده‌ی خود را در اختیار این پروژه قرار می‌دهند. هم اکنون توان پردازشی این پروژه ۲ پتافلاپس (۲ هزار تریلیون عملیات اعشاری در ثانیه) می‌باشد که در مقایسه با قوی‌ترین ابرکامپیوتر جهان به اسم Roadrunner که دارای توان پردازشی ۱ پتافلاپس است، رقم جالبی به نظر می‌رسد. لازم به ذکر است که Roadrunner با هزینه‌ی ۱۳۳ میلیون دلار در ژوئن ۲۰۰۸ به بهره‌برداری رسید (اگر کل جمعیت شش میلیاردی زمین، به طور شبانه‌روز با ماشین حساب مشغول حساب کردن باشند، ۴۶ سال طول می‌کشد تا کاری که Roadrunner تنها در یک روز انجام می‌دهد را به سرانجام برسانند [12]).

R-9

معادله دیفرانسیل حاکم بر جابجایی یک سیستم تک‌درجه آزادی الاستیک خطی با استهلاک لزج تحت اثر نیروی دینامیکی خارجی $p(t)$ به صورت رابطه‌ی زیر می‌باشد [13].

$$m\ddot{u} + c\dot{u} + ku = p(t) \quad (1)$$

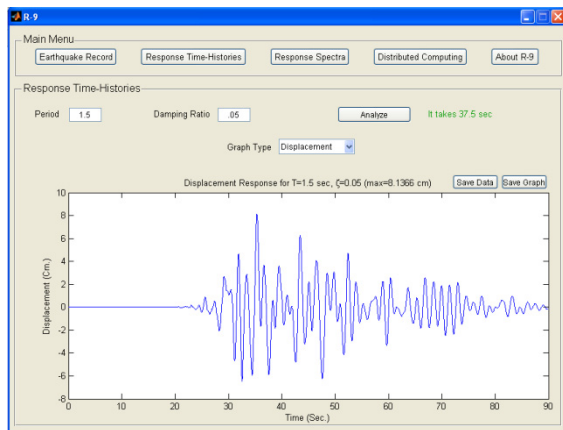
که در آن، m جرم سیستم، c ضریب میرایی، k سختی و u جابجایی است. در مهندسی زلزله، نیروی $p(t)$ در واقع همان شتاب ناشی از زلزله می‌باشد که به جرم سیستم اعمال می‌شود. لذا با جایگذاری $-m\ddot{u}_g(t)$ و با فرض شرایط اولیه سکون، حل معادله بالا به کمک انتگرال دوهمال به صورت رابطه‌ی (2) در خواهد آمد.

$$u(t) = -\frac{1}{\omega_D} \int_0^t \ddot{u}_g(\tau) e^{-\xi\omega(t-\tau)} \sin(\omega_D(t-\tau)) d\tau \quad (2)$$

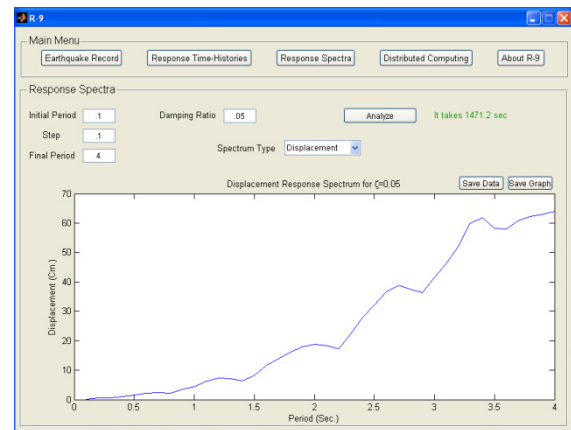
در این رابطه، ω_D فرکانس طبیعی با میرایی، ξ نسبت میرایی و ω فرکانس طبیعی می‌باشد. شتاب‌های ناشی از زلزله توسط دستگاه شتاب‌نگاشت در بازه‌های زمانی خیلی کوتاه ثبت می‌گردند و در نهایت در قالب یک فایل کامپیوتری در دسترس می‌باشد. که عموماً به آن رکورد زلزله گفته می‌شود. چنانچه رکورد زلزله دارای تعداد زیادی نقاط (شتاب‌های ثبت شده) باشد، آنگاه فرآیند انتگرال‌گیری رابطه‌ی (2) زمانبر خواهد بود. خصوصاً اینکه جهت ترسیم طیف پاسخ، این فرآیند انتگرال‌گیری باید به ازای تک تک فرکانس‌ها تکرار شود (البته اگر قصد استفاده از روش‌های دیگری همچون روش اختلاف مرکزی نباشد). برای مثال، رکورد زلزله‌ی چی چی تایوان که از سایت «مرکز تحقیقات مهندسی زلزله پاسیفیک» (PEER) دانلود شد، دارای ۱۸۰۰۰ نقطه است و این به منزله‌ی ۱۸۰۰۰ بار انتگرال‌گیری جهت یافتن حداکثر جابجایی مطلق به ازای یک فرکانس مشخص می‌باشد. حال اگر قرار باشد طیف پاسخ آن به ازای ۴۰ فرکانس مختلف (مثلاً از ۲ تا ۰.۵، با گام‌های ۱.۵) ترسیم شود، نیاز است که ۷۲۰۰۰۰ بار عملیات انتگرال‌گیری صورت پذیرد (۷۲۰۰۰۰ = ۴۰ × ۱۸۰۰۰).

با اندکی تأمل در مسئله‌ی ترسیم طیف پاسخ به کمک انتگرال دوهمال، به این نکته پی برده می‌شود که ساختار این مسئله به صورت SIMD (تک دستورالعمل، چندین داده) می‌باشد. زیرا تنها یک جریان دستورالعمل، یعنی انتگرال دوهمال و چندین جریان داده که در واقع از ضرب فرکانس‌های مختلف در رکورد زلزله ایجاد می‌شوند، وجود دارد. در نتیجه می‌توان از محاسبات توزیع شده بهره گرفت و محاسبه‌ی حداکثر جابجایی مطلق به ازای هر

فرکانس را به کامپیوتر جداگانه‌ای سپرد و در نهایت نتایج را به کامپیوتر مرکزی ارسال کرد. در همین راستا، برنامه‌ای به اسم R-9 به زبان MATLAB نوشته شد. این برنامه که برای درس مهندسی زلزله ارائه گردید، با دریافت رکورد زلزله و وارد نمودن پارامترهای مربوطه، تاریخچه‌های زمانی پاسخ جابجایی، سرعت و شتاب آن را ترسیم می‌کند. برای مثال در شکل ۶-الف، تاریخچه‌ی زمانی پاسخ جابجایی زلزله‌ی چی چی تایوان به ازای تناوب ۱.۵ ثانیه و نسبت میرایی ۰.۰۵ ملاحظه می‌شود. همچنین می‌توان با وارد نمودن بازه‌ای از تناوب‌ها و یک نسبت میرایی مشخص، طیف‌های پاسخ جابجایی، سرعت، شتاب و شبه شتاب را نیز ترسیم کرد. البته برای ترسیم این طیف‌های پاسخ، R-9 می‌تواند به صورت محاسبه بر روی تک کامپیوتر و یا چند کامپیوتر (محاسبات توزیع‌شده) عمل کند. در شکل ۶-ب، طیف پاسخ جابجایی همان زلزله در بازه‌ی تناوب‌های ۰.۱ تا ۴ ثانیه (با گام‌های ۰.۱) با نسبت میرایی ۰.۰۵ و به صورت محاسبه بر روی تک کامپیوتر ملاحظه می‌شود.



(الف)

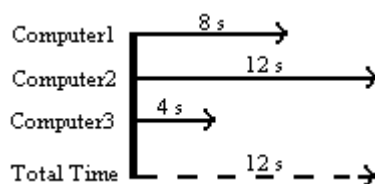


(ب)

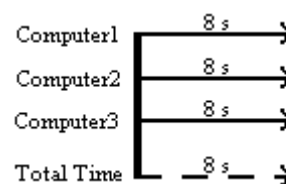
شکل ۶- برنامه‌ی R-9 (الف) تاریخچه زمانی پاسخ جابجایی (ب) طیف پاسخ جابجایی

در حالت استفاده از R-9 به صورت محاسبات توزیع‌شده، می‌بایست بر روی کامپیوترهایی که قرار است بار محاسباتی بین آنها توزیع گردد، برنامه‌ی R9worker اجرا شود (وظیفه‌ی این برنامه، دریافت رکورد زلزله، نسبت میرایی، تناوب یا بازه‌ای از تناوب‌ها از کامپیوتر مرکزی - کامپیوتری که R-9 بر روی آن در حال اجرا شدن است - سپس بدست آوردن حداکثر جابجایی مطلق به ازای تک تک تناوب‌های دریافت شده و در نهایت ارسال نتایج به کامپیوتر مرکزی می‌باشد). پس از اجرای R9worker بر روی آن کامپیوترها، اکنون می‌توان در کامپیوتر مرکزی بار محاسباتی تک تک کامپیوترها را تعیین کرد.

نحوه‌ی تعیین و توزیع بار محاسباتی، به تعداد کامپیوترها و توان پردازنده‌ی آنها بستگی دارد. برای مثال چنانچه قرار باشد طیف پاسخ جابجایی به ازای ۴۰ تناوب مختلف ترسیم گردد و تعداد کامپیوترها نیز ۴۰ عدد باشد، آنگاه به راحتی می‌توان پردازش هر کدام از این تناوب‌ها را به کامپیوتر جداگانه‌ای سپرد. اما چنانچه فقط ۸ کامپیوتر در دسترس باشد، آنگاه می‌بایست هر کدام از آن کامپیوترها، ۵ تناوب مختلف را مورد پردازش قرار دهند. ضمن اینکه اگر کامپیوتری دارای پردازنده‌ی ضعیف‌تری نسبت به بقیه باشد، می‌بایست بار محاسباتی کمتری نیز به آن سپرده شود. زیرا زمانی طیف پاسخ ترسیم خواهد شد که همه‌ی کامپیوترها نتایج را به مرکز ارسال کرده باشند و حتی اگر نتایج یک کامپیوتر باقی مانده باشد، طیف پاسخ ترسیم نخواهد شد. لذا با سپردن بار محاسباتی کمتری به کامپیوتر با پردازنده‌ی ضعیف‌تر، مدت زمان آنالیز بر روی همه‌ی کامپیوترها تقریباً مساوی در خواهد آمد. به طور خلاصه می‌توان گفت در محاسبات توزیع‌شده، مدت زمان کل آنالیز برابر است با مدت زمان آنالیز در کامپیوتری که دیرتر از سایر کامپیوترها نتایج را به مرکز ارسال می‌کند. برای مثال چنانچه ۳ کامپیوتر با پردازنده‌های مختلف، مدت زمان ارسال نتایج در آنها به ترتیب: ۸، ۴ و ۱۲ ثانیه باشد آنگاه زمان کل آنالیز برابر با ۱۲ ثانیه است (شکل ۷-الف). اما اگر بار محاسبات را به گونه‌ای توزیع کرد که کامپیوتری که فقط ۴ ثانیه درگیر محاسبات بوده، بار بیشتری به آن تعلق گیرد و کامپیوتری که ۱۲ ثانیه درگیر محاسبات بوده، بار کمتری به آن تعلق گیرد، آنگاه می‌توان مدت زمان ۸ ثانیه را برای همه‌ی آنها پیش بینی کرد که در نتیجه مدت زمان کل آنالیز نیز به ۸ ثانیه کاهش می‌یابد (شکل ۷-ب).



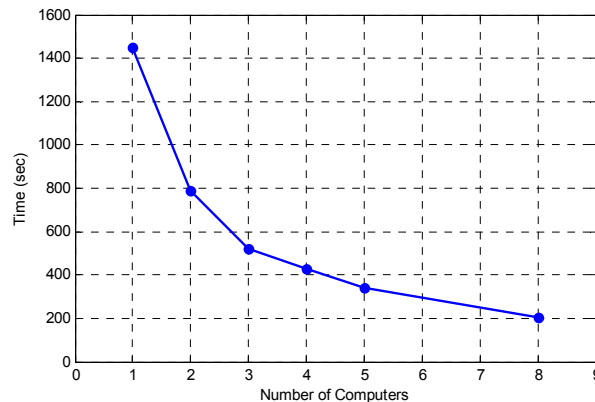
(الف)



(ب)

شکل ۷- تاثیر نحوه‌ی توزیع بار در افزایش یا کاهش زمان کل محاسبات (الف) توزیع ناکارا (ب) توزیع بهینه

پس از آشنایی با نحوه‌ی توزیع بار، اکنون به راحتی می‌توان از R-9 به صورت توزیع‌شده استفاده نمود. در آزمایشی که برای ترسیم طیف پاسخ صورت گرفت، ابتدا R-9 بر روی تک کامپیوتر و سپس بر روی دو، سه، چهار، پنج و هشت کامپیوتر توزیع گردید که زمان آنالیز آنها به ترتیب ۱۴۵۱، ۷۹۰، ۵۲۳، ۴۲۹، ۳۴۰ و ۲۰۷ ثانیه بدست آمد. در شکل ۸، نمودار همین آزمایش ملاحظه می‌شود.



شکل ۸ - تعداد کامپیوترها و زمان آنالیز متناظر آنها در ترسیم طیف پاسخ توسط R-9

نتیجه‌گیری

با اندکی دقت در محیط پیرامون خود، به این نکته پی می‌بریم که کامپیوترهای زیادی در اطرافمان وجود دارند که حداقل نیمی از اوقات بیکار بوده و یا اینکه در حال انجام کارهای ساده‌ای می‌باشند که درصد بسیار کمی از وقت پردازنده را به خود اختصاص می‌دهد. تکنولوژی محاسبات توزیع شده با بهره‌گیری از همین منابع ارزشمند که به تعبیری می‌توان گفت در حال هدر رفتن می‌باشند، یک ابرکامپیوتر خانگی برای ما فراهم می‌کند که با استفاده از آن می‌توان به پرسش‌هایی جواب داد که ذهن کنجکاومان بی‌صبرانه منتظر شنیدن آن است.

قدردانی

با تشکر صمیمانه از والدین عزیزم، فاطمه و عباس تبرکی که با دلگرمی و پشتیبانی همیشگی خود، بی‌شک دوست داشتنی‌ترین محیط توزیع شده برای من هستند.

مراجع

1. Moore, G. E. (1965) Cramming More Components onto Integrated Circuits. *Electronics*, 38, 8.
2. Ross, P.E. (2008) Why CPU Frequency Stalled. *Spectrum*, IEEE, 45, 4, 72-72.
3. Rajan, S.D. and Belegundu, A.D. (2004) Finite Element Analysis & Design Optimization in a Distributed Computing Environment, *Proceedings, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY Aug 2004.
4. S. Hernández, J. A. Jurado & F. Nieto. (2005) Aeroelastic analysis and sensitivity of the flutter speed of long span suspension bridges with distributed computing. *Computer Aided Optimum Design in Engineering IX* 269.
5. Henry X. Liu, Wenteng Ma, R. Jayakrishnan, and Will Recker. (2004) Large-Scale Traffic Simulation Through Distributed Computing of Paramics. *California Partners for Advanced Transit and Highways (PATH)*.
6. Patel, P. V. Patodi, S. C. (2006) Distributed Finite Element Analysis of Laminated Composites. *Journal of institution of engineers India part cv Civil Engineering Division*, 87, 17-22.
7. Ajay D. Kshemkalyani and Mukesh Singhal (2008) *Distributed Computing; Principles, Algorithms, and Systems*, Cambridge University Press.
8. Flynn, M. (1972) Some Computer Organizations and Their Effectiveness, *IEEE Trans. Comput.*, 21, 948.
9. Charlie Obimbo, Behzad Salami (2007) A Parallel Algorithm for determining the inverse of a matrix for use in blockcipher encryption/decryption. *The Journal of Supercomputing*. 39, 2, 113-130.
10. Mahmood, A.; Dichter, J. (2002) An efficient parallel algorithm for the inverse of a banded matrix in the maximum entropy sense. *IEEE International Conference on Systems, Man and Cybernetics*. 4, 6.
11. <http://folding.stanford.edu/>
12. The New York Times, June 9, 2008 (<http://www.nytimes.com/2008/06/09/technology/09petaflops.html>).
13. Anil K. Chopra (1995) *Dynamics of structures : theory and applications to earthquake engineering*.