مسائل زمان‌بندی در بهینه‌سازی پرس‌وجوهای موازی برای محیط‌های ناهمگون

آرش ترمه‌چی
termechi@ce.sharif.ac.ir

محمّد قدسی
ghodsi@sharif.ac.ir

دانشکدهٔ مهندسی کامپیوتر
دانشگاه صنعتی شریف

چکیده

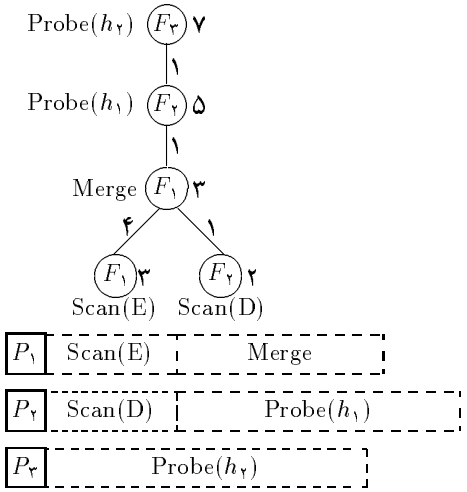
با توجه با گسترش پایگاه‌داده‌های موازی در محیط‌هایی نظیر شبکه‌ای از کامپیوترها، که در آن پردازنده‌ها می‌توانند دارای توان‌های پردازشی متفاوتی باشند؛ زمان‌بندی عملگرهای لازم برای اجرای پرس‌وجوی موازی در این محیط‌ها، یکی از مهم‌ترین قسمت‌های بهینه‌سازی پرس‌وجوها در پایگاه‌داده‌های موازی است. در این مقاله ما یکی از مهم‌ترین مسائل مطرح شده در زمان‌بندی پرس‌وجوهای موازی، مسألهٔ زمان‌بندی درخت عملگر لوله‌ای، را برای محیط‌های ناهمگون تعمیم داده و الگوریتمی برای حل آن در این حالت پیشنهاد کرده‌ایم. بررسی حداکثر ضریب تقریب و ضریب تقریب متوسط این الگوریتم حاکی از کارایی بالا و ضریب تقریب قابل قبول این الگوریتم برای حل مسأله می‌باشد. این الگوریتم حتی در حالاتی که در آنها پردازنده‌ها همگون هستند؛ دارای ضریب تقریب بهتری نسبت به الگوریتم مشابه موجود برای حل مسأله می‌باشد.

کلمات کلیدی: پایگاه‌دادهٔ موازی *Parallel Database*، بهینه‌سازی پرس‌وجوی موازی *Parallel Query Optimization*، زمان‌بندی درخت عملگر لوله‌ای *Scheduling Pipelined Operator Tree*، محیط‌های ناهمگون *Heteregeneous Environment*

۱ مقدمه

یکی از مهم‌ترین قسمت‌های بهینه‌سازی پرس‌وجوی موازی، تعیین چگونگی زمان‌بندی عملگرهای درخت عملگرِ متناظر با پرس‌وجوی ورودی بر روی پردازنده‌های سیستم است [۴] [۱۰]. هر عملگر در درخت عملگر بخشی از یک عملگر رابطه‌ای موجود در پرس‌وجو را پیاده‌سازی می‌کند [۸]. مجموعهٔ مسائل زمان‌بندی درخت عملگر برای نخستین بار توسط و. حسن[۱] و ر. موتوانی[۲] با تعریف مسألهٔ زمان‌بندی درخت عملگر لوله‌ای[۳] در [۱۰] مطرح شد. درخت عملگر لوله‌ای درخت عملگری است که کلیه عملگرها در آن می‌توانند به صورت لوله‌ای با یکدیگر اجرا شوند. به منظور احتساب هزینه‌های اجرای هر عملگر و انتقال داده میان عملگرها، به گره‌های درخت عملگر، وزنی که معادل زمان اجرای عملگر متناظر با آن است؛ و به یال‌های آن وزنی که نشانگر زمان صرف شده برای انتقال داده‌ها میان پردازنده‌های مجری عملگرهای

[۱] W. Hasan
[۲] R. Motwani
[۳] Pipelined Operator Tree



شکل ۱: زمان‌بندی درخت عملگر لوله‌ای

مجاور یال می‌باشد؛ متناظر می‌شود (شکل ۱).

هدف نهایی مسأله، یافتن یک زمان‌بندی از درخت عملگر بر روی پردازنده‌های سیستم، به گونه‌ای است که زمان پاسخ‌گویی کمینه شود. این مسأله نیز مانند اغلب مسائل زمان‌بندی اِن‌پی‌هارد است[۱۰]. و.حسن و دیگران در[۱۰] الگوریتم‌های تقریبی برای حل حالات خاصی از مسأله پیشنهاد کرده‌اند. چکوری[۴] و دیگران در [۳] و و.حسن در[۸] دو الگوریتم تقریبی برای حل مسأله در حالت کلی ارائه کرده‌اند. این الگوریتم‌ها به ترتیب دارای حداکثر ضریب تقریب ۳/۵۶ با پیچیدگی زمانی از مرتبهٔ $O(n \log n)$ و ۲/۸۷$(1+\epsilon)$ با پیچیدگی زمانی از مرتبهٔ $O(\epsilon^{-1} pn \log n)$ که در آنها n تعداد عملگرها، p تعداد پردازنده‌ها و ϵ یک عدد حقیقی دلخواه می‌باشد، است.

با توجه به گسترش روزافزون شبکه‌های کامپیوتری، امروزه طراحی و پیاده‌سازی پایگاه‌داده‌های موازی در محیط‌های چندکامپیوتری به عنوان یک موضوع تحقیقاتی مورد توجه قرار گرفته است [۲] [۱]. در این محیط‌ها هر ایستگاه کاری ممکن است دارای توان پردازشی متفاوت از سایر ایستگاه‌های کاری باشد. جدا از مشخصات سخت‌افزاری متفاوت، امکان استفادهٔ چند منظوره از هر ایستگاه کاری در شبکه که باعث تقسیم توان کاری پردازنده میان کاربردهای مختلف می‌گردد، نیز باعث بروز چنین خصوصیتی می‌شود[۵] [۸]. طراحی الگوریتم‌های لازم برای بهینه سازی پرس‌وجوهای موازی در چنین محیط‌هایی از مسائل باز در این زمینه می‌باشد[۵] [۶] [۸] [۹]. ویژگی ناهمگونی در پردازنده‌های مورد استفاده برای پردازش پرس‌وجو، بیشترین تأثیر را در نحوهٔ زمان‌بندی درخت عملگر حاصل از پرس‌وجو می‌گذارد.

[۴] Chekuri

در این مقاله ابتدا مسألهٔ جدید زمان‌بندی درخت عملگر لوله‌ای برای محیط‌های ناهمگون را تعریف می‌کنیم. در بخش بعد روش کلی ارائه شده برای حل مسأله در محیط‌های همگون را برای حل مسأله در این حالت تعمیم خواهیم داد. با استفاده از این روش در بخش ۴ الگوریتمی برای مسألهٔ جدید پیشنهاد می‌کنیم.

۲ تعریف مسأله

یک زمان‌بندی از درخت عملگر را می‌توان به صورت زیر تعریف کرد[۱۰].

تعریف ۱.۲ یک زمان‌بندی درخت عملگر $T=(V,E)$ بر روی p پردازنده، افراز V، مجموعهٔ گره‌های درخت، به p مجموعهٔ $F_1, F_2, \ldots, F_p$ است، که در آن مجموعهٔ F_k به پردازندهٔ k تخصیص داده می‌شود.

در شکل ۱ یک زمان‌بندی از درخت عملگر بر روی سه پردازنده نمایش داده شده است. در ادامهٔ بحث، طبق قرارداد وزن گره i را با t_i، و وزن یال $(i,j) \in E$ را با C_{ij} نشان می‌دهیم. بنابراین بار پردازندهٔ k، L_k، برابر خواهد بود با[۱۰]

$$L_k = \sum_{i \in F_k} [t_i + \sum_{j \notin F_k} C_{ij}]$$

به دلیل وجود توازی لوله‌ای کلیهٔ عملگرهای درخت باید تقریباً در زمان یکسانی آغاز شده و خاتمه یابند[۱۰]. حال با احتساب ویژگی ناهمگونی درپردازنده‌های محیط، مسألهٔ زمان‌بندی درخت عملگر لوله‌ای را می‌توان به صورت زیر تعریف کرد.

تعریف ۲.۲ مسألهٔ زمان‌بندی درخت عملگر لوله‌ای در محیط‌های ناهمگون

ورودی: درخت عملگر لوله‌ای $T=(V,E)$ که هر گره $i \in V$ آن دارای وزن $t_i > 0$ و هر یال $(i,j) \in E$ آن دارای وزن $C_{ij} \geq 0$ می‌باشد، p پردازنده $m_1, \ldots, m_p$ با سرعتهای $S_1, \ldots, S_p$.

خروجی: یک زمان‌بندی با حداقل زمان اجرا، به عبارت دیگر یک افراز $F_1, \ldots, F_p$ بر روی V به‌طوریکه عبارت زیر کمینه شود.

$$\max_{1 \leq l \leq p} \left[\left(\sum_{i \in F_l} [t_i + \sum_{j \notin F_l} C_{ij}] \right) / S_l \right]$$

از آنجا که در حالت خاصی که سرعت کلیهٔ پردازنده‌ها برابر باشد، مسأله تبدیل به مسألهٔ زمان‌بندی درخت عملگر لوله‌ای در محیط همگون می‌شود، لذا این مسأله نیز ان‌پی‌هارد است. به راحتی می‌توان ثابت کرد که مسأله قابل تبدیل به حالتی است که در آن $S_1 = 1$ و $S_i \leq S_{i+1}, 1 \leq i \leq p-1, 2$ می‌باشند [۷]. بنابراین در ادامه تنها این حالت از مسأله را در نظر می‌گیریم.

همان‌گونه که در [۱۰] اشاره شده، یک یال بی‌ارزش[۵] نشان دهنده بیشتر بودن هزینهٔ انتقال داده‌ها از هزینهٔ اجرای گره‌های همسایهٔ آن یال است. به عبارت دیگر $(i,j) \in E$ بی‌ارزش است اگر و فقط اگر $C_{ij} \geq t_i + \sum_{k \neq j} C_{ik}$ یا $C_{ij} \geq t_j + \sum_{k \neq i} C_{jk}$ باشند. بنابراین اجرای موازی دو پردازهٔ i و j در زمان‌بندی درخت عملگر لوله‌ای هزینه‌ای بیشتر و یا حداقل برابر با اجرای ترتیبی این دو پردازه خواهد داشت [۱۰]. با استدلالی مشابه با [۱۰] می‌توان قضیهٔ زیر را در مورد محیط‌های ناهمگون بیان نمود.

قضیه ۱.۲ . p پردازنده با سرعت‌های $1 \leq i \leq p, S_i \geq 1$ و درخت عملگرلوله‌ای $T=(V,E)$ با یال بی‌ارزش (i,j) را در نظر بگیرید. یک زمان‌بندی بهینه برای T بر روی p پردازنده وجود دارد، به طوری که گره‌های i و j به یک پردازنده واگذار شوند.

بنابراین در این حالت نیز همواره زمان پاسخ‌گویی یک درخت یکنوا[۶]، درختی که دارای یال بی‌ارزش نمی‌باشد [۱۰]، بزرگتر یامساوی زمان پاسخ‌گویی درخت نایکنوای متناظر آن است. پس الگوریتم پیمایش سودجویانه[۷] که در [۱۰] برای پردازنده‌هایی با سرعت‌های برابر طراحی شده است؛ نیز در این وضعیت می‌تواند به عنوان یک مرحلهٔ پیش‌پردازش درخت عملگر ورودی قبل از هر گونه زمان‌بندی به کار رود. الگوریتم برای حذف یال‌های بی‌ارزش از عمل ادغام[۸] استفاده می‌کند.

تعریف ۳.۲ ادغام دو گره i و j، $Collapse(i,j)$، دو گره i و j را با گره جدید i' با وزن $t_{i'} = t_i + t_j$ جایگزین می‌کند. یال‌های متصل به گره‌های i و j در درخت، به گره جدید متصل می‌شوند.

الگوریتم برای انجام کار، به طور ساده درخت عملگر را پیمایش کرده و به محض رسیدن به یال بی‌ارزش، دو گره مجاور آن را در هم ادغام می‌کند.

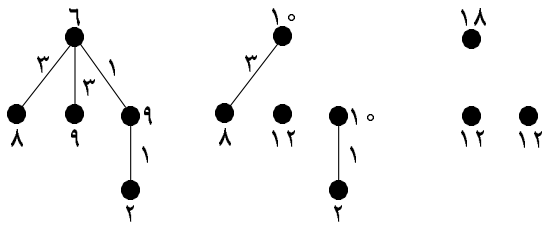
۳ روش بخش‌بندی / زمان‌بندی

یک روش برای یافتن راه حل مناسب مسألهٔ زمان‌بندی درخت عملگر لوله‌ای در حالت کلی، روش دومرحله‌ای می‌باشد [۳] [۸]، که ما از آن با نام روش بخش‌بندی / زمان‌بندی یاد می‌کنیم. این روش برای اولین بار توسط چکوری و دیگران در[۳] ارائه شد و توسط حسن در[۸] تکمیل گردید. در این روش در مرحلهٔ اول درخت ورودی بخش‌بندی[۹] و در مرحلهٔ بعد زمان‌بندی می‌گردد. بخش بندی در حقیقت تقسیم درخت عملگر به مجموعه‌های همبندی از گره‌ها، بدون توجه به تعداد پردازنده‌های سیستم است. بخش‌های حاصل از این مرحله در مرحلهٔ دوم به پردازنده‌های موجود واگذار می‌شوند. بخش‌بندی درخت عملگر با استفاده از دو عمل ادغام و قطع[۱۰] [۳] انجام می‌گیرد.

تعریف ۱.۳ قطع یال (i,j)، $Cut(i,j)$، یال (i,j) را از درخت حذف کرده و وزن آن را به وزن دو گره i و j اضافه می‌کند. به عبارت دیگر $t_i^{new} = t_i^{old} + C_{ij}$ و $t_j^{new} = t_j^{old} + C_{ij}$ خواهد شد.

نقطهٔ قوت این دیدگاه، امکان استفاده از الگوریتم‌های مختلف موجود برای زمان‌بندی چندپردازنده‌ای، برای مرحلهٔ دوم است. ضریب تقریب مناسب و قابل قبول الگوریتم ال‌پی‌تی برای زمان‌بندی پردازه‌ها بر روی پردازنده‌ها با سرعت متفاوت [۷]، اجازه می‌دهد تا از آن برای مرحلهٔ زمان‌بندی استفاده کنیم. پیش از بررسی رابطهٔ میان بخش‌بندی و زمان‌بندی ، ذکر تعاریف زیر ضروری است [۳].

---
[۵] Worthless Edge
[۶] Monotone
[۷] Greedy Chase
[۸] Collapse
[۹] Fragmentation
[۱۰] Cut



شکل ۲: یک بخش‌بندی از درخت عملگر

تعریف ۲.۳ وزن هر بخش F به صورت $M = \sum_{i \in F}[t_i + \sum_{j \notin F} C_{i,j}]$ تعریف می‌شود. بیشترین وزن میان بخش‌های درخت را با M نشان می‌دهیم.

تعریف ۳.۳ وزن شبکه‌ای[11] گره i در درخت عملگر لوله‌ای $T = (V, E)$ به صورت $R_i = t_i + \sum_{j \neq i} C_{ij}$ تعریف می‌شود. بیشترین مقدار وزن شبکه‌ای در درخت با R نمایش داده می‌شود.

در گزاره‌های زیر منظور از الگوریتم بخش‌بندی / اِل‌پی‌تی، الگوریتمی است که با استفاده از روش بخش‌بندی / زمان‌بندی که زمان‌بندی آن به شیوهٔ اِل‌پی‌تی انجام می‌شود، درخت عملگر ورودی را زمان‌بندی می‌کند. $\hat{T}_p$ طول زمان‌بندی به دست آمده توسط این الگوریتم و $T^\star_p$ طول زمان‌بندی بهینهٔ درخت عملگر ورودی بر روی p پردازنده ، هستند. به سادگی می‌توان ثابت نمود:

لم ۱.۳ . یک سیستم $p+1$ پردازنده‌ای که در آن $1 = S_1 \leq S_2 \leq \ldots \leq S_{p+1}$ باشد و درخت عملگر لوله‌ای $T = (V, E)$ را در نظر بگیرید. اگر پردازنده‌ای نظیر m_i وجود داشته باشد، به طوری که هم در زمان‌بندی بهینهٔ T و هم در زمان‌بندی با الگوریتم بخش‌بندی / اِل‌پی‌تی برای T بیکار باشد، آنگاه خواهیم داشت.

$$\hat{T}_{p+1}/T^\star_{p+1} \leq \hat{T}_p/T^\star_p$$

در قضیهٔ زیر، بار کل، L، برابرمجموع کل بارهای تخصیص داده شده به پردازنده‌ها می‌باشد. بار کل در زمان‌بندی بهینه را با L_{opt} نشان می‌دهیم.

قضیه ۱.۳ . اگر برای بخش‌بندی F از درخت عملگر یکنوای T، $L \leq K_2 L_{opt}$ و $M \leq K_1 R$ باشند. در صورت زمان‌بندی بخش‌های این بخش‌بندی با الگوریتم اِل‌پی‌تی بر روی p پردازنده با سرعت‌های مختلف، ضریب تقریب کل الگوریتم برابر با $\max(K_1, \frac{2p-1}{p} K_2)$ خواهد بود.

اثبات. برای $p = 1$ حکم واضح است. فرض می‌کنیم برای کلیهٔ مقادیر $1, 2, \ldots, p-1$ حکم برقرار بوده ولی برای p پردازنده $\hat{T}/T^\star > \max(K_1 \frac{2p-1}{p} K_2)$ باشد. فرض می‌کنیم در این حالت، بخش‌بندی F دارای l بخش بوده و بخش M_i که برای آن داریم $M \geq M_1 \geq \ldots \geq M_i \geq \ldots \geq M_l$، تعیین کنندهٔ طول زمان‌بندی و آخرین بخشی باشد که اجرای آن به پایان می‌رسد. در این صورت کلیهٔ بخش‌های با وزن کمتر از M_i را از بخش‌بندی حذف می‌کنیم. واضح است که در این صورت $T^\star$ افزایش نیافته و $\hat{T}$ نیز بدون تغییر می‌ماند، بنابراین ضریب تقریب، طبق فرض خلف، کماکان بزرگتر از $\max(K_1, \frac{2p-1}{p} K_2)$ خواهد بود. حاصل، بخش‌بندی جدید F' می‌باشد که در آن بخش با کمترین وزن، M_i، تعیین کنندهٔ طول زمان‌بندی است. اگر وزن سنگین‌ترین بخش بخش‌بندی F' را M' و بار کل حاصل از بخش‌بندی را L' بنامیم، طبق فرض $M' \leq M$ و $L' \leq L$ می‌باشند. حال نسبت به وضعیت L' و p نسبت به یکدیگر، دو حالت رخ می‌دهد.

الف) حالتی که تعداد بخش‌ها از تعداد پردازنده‌ها بزرگتر یا با آن مساوی باشند.

اگر $i \geq p$ باشد، آنگاه با توجه به ویژگی الگوریتم اِل‌پی‌تی که پردازه‌ها را به پردازنده‌ای واگذار می‌کند که زمان کامل شدن پردازه در آن کمتر باشد، به هنگام واگذاری M_i خواهیم داشت:

$$\forall 1 \leq j \leq p \; : \; \hat{T} \leq \frac{L_k + M_i}{S_k} \leq \frac{L'_j + M_i}{S_j}$$

که $\hat{T}$ زمان حاصل از الگوریتم اِل‌پی‌تی برای زمان‌بندی بخش‌های F' و L'_j بار واگذار شده به پردازنده j هنگام زمان‌بندی M_i و m_k پردازنده‌ای است که طول زمان‌بندی را تعیین می‌کند. با توجه به این نامساوی برای کلیهٔ مقادیر j، $\hat{T} S_j \leq L'_j + M_i$ می‌باشد. با جمع همهٔ p نامساوی، خواهیم داشت:

$$\hat{T}(\sum_{j=1}^{p} S_j) \leq \sum_{j=1}^{p} L'_j + pM_i$$

[11] Net Weight

و چون $\sum_{j=1}^{p} L'_j = L' - M_i$ است، پس:

$$\hat{T}(\sum_{j=1}^{p} S_j) \leq L' + (p-1)M_i$$

$$\hat{T} \leq \frac{L'}{\sum_{j=1}^{p} S_j} + \frac{(p-1)M_i}{\sum_{j=1}^{p} S_j}$$

با توجه به نامساوی $L' \leq L$ داریم:

$$\hat{T} \leq \frac{L}{\sum_{j=1}^{p} S_j} + \frac{(p-1)M_i}{\sum_{j=1}^{p} S_j}$$

بنابراین طبق فرض خواهیم داشت:

$$\hat{T} \leq \frac{K_2 L_{opt}}{\sum_{j=1}^{p} S_j} + \frac{(p-1)M_i}{\sum_{j=1}^{p} S_j}$$

از طرف دیگر $T^\star \leq \frac{L_{opt}}{\sum_{j=1}^{p} S_j}$ می‌باشد. پس:

$$\hat{T} \leq K_2 T^\star + (p-1)\frac{M_i}{\sum_{j=1}^{p} S_j}$$

با توجه به نامساوی‌های بالا می‌توان نوشت:

$$\sum_{j=1}^{p} S_j \geq \frac{L_{opt}}{T^\star} \Longrightarrow \hat{T} \leq K_2 T^\star + (p-1)\frac{M_i}{L_{opt}} T^\star$$

از سوی دیگر چون $L' \leq L$ است، با توجه به فرض خواهیم داشت

$$\hat{T} \leq K_2 T^\star + \frac{(p-1)K_2 M_i}{L'} T^\star$$

از آنجا که M_i کوچکترین بخش F' است. $L' \geq iM_i$ خواهد بود، پس

$$\hat{T} \leq K_2 T^\star + \frac{(p-1)K_2}{i} T^\star$$

$$\hat{T} \leq K_2 T^\star + \frac{(p-1)K_2}{p} T^\star$$

$$\hat{T} \leq K_2 \frac{2p-1}{p} T^\star$$

$$\hat{T} \leq \max(K_1, K_2 \frac{2p-1}{p}) T^\star$$

ب) حالتی که $i < p$ باشد.

در این حالت در زمان‌بندی حاصل از الگوریتم اِل‌پی‌تی بخش‌های F' حداقل یک پردازنده مانند m_d بیکار خواهد بود. اگر این پردازنده در زمان‌بندی بهینه نیز بیکار باشد. آنگاه اگر زمان حاصل از زمان‌بندی اِل‌پی‌تی بخش‌های F' بر روی $p-1$ پردازنده را $\hat{T'}_{p-1}$، و زمان بهینهٔ زمان‌بندی این بخش‌ها را $T^{\star'}_{p-1}$ بنامیم، داریم

$$\hat{T'}_p / T^{\star'}_p \leq \hat{T'}_{p-1} / T^{\star'}_{p-1}$$

$$\hat{T}_p / T^\star_p \leq T'_{p-1} / T^{\star'}_{p-1}$$

که طبق فرض استقرا، حکم ثابت می‌شود.

در غیر این صورت، اگر این پردازنده در زمان‌بندی بهینهٔ F' بیکار نباشد، آنگاه برای طول زمان‌بندی بهینهٔ F'، $T^{\star'}$، همواره $T^{\star'} \geq \frac{R}{S_d}$ خواهد بود. زیرا R کمترین مقدار بار اعمال شده بر روی یک پردازندهٔ غیر بیکار می‌باشد. ولی چون این پردازنده در زمان‌بندی اِل‌پی‌تی، بیکار است، داریم

$$\hat{T} \leq \frac{M_i}{S_d} \leq \frac{M}{S_d}$$

$$\hat{T} \leq \frac{K_1 R}{S_d} \Longrightarrow \hat{T} \leq K_1 T^{\star'}$$

$$\hat{T} \leq K_1 T^\star \Longrightarrow \hat{T} \leq \max(K_1, K_2 \frac{2p-1}{p}) T^\star$$

پس در هر مورد فرض خلف نادرست و حکم صادق می‌باشد.

□

بنابراین برای کاهش ضریب تقریب الگوریتم بخش‌بندی / اِل‌پی‌تی باید به دنبال یافتن کمترین مقدار برای K_1 و K_2 باشیم، تا به ضریب تقریبی نزدیک به $2 - \frac{1}{p}$ دست پیدا کنیم. ولی یافتن مقادیر بهینه برای K_1 و K_2، مثلاً $K_2 = 1$ و $K_1 = 1$ منتهی به مسأله‌ای اِن‌پی‌هارد خواهد شد [۳]. پس برای یافتن بهترین بخش‌بندی نیز باید از الگوریتم‌های تقریبی استفاده کنیم.

۴ الگوریتم برش‌های محلی گسترش‌یافته

الگوریتم بخش‌بندی برش‌های محلی[۱۲] در [۳] برای حل مسأله برای پردازنده‌های همگون، ارائه شده است. الگوریتم از یک پارامتر ورودی $\alpha > 1$ برای بخش‌بندی استفاده می‌کند. منظور از گره مادر در الگوریتم، گره‌ای است که کلیهٔ فرزندان آن گره برگ باشند.

[۱۲] Local Cuts

الگوریتم ۱.۴ الگوریتم برش‌های محلی

Algorithm Local Cuts.

1.**while** there exist a mother node m with children j **do**
2. **if** $t_j > \alpha C_{jm}$ **then**
3. Cut(j,m);
4. **else**
5. Collapse(j,m);
6.**end while**

برای الگوریتم داریم[۳]:

لم ۱.۴ . کل هزینهٔ انتقال در بخش‌بندی حاصل از الگوریتم برش‌های محلی، حداکثر برابر $\frac{۲W}{\alpha-۱}$ است. به عبارت بهتر $C \leq \frac{۲W}{\alpha-۱}$ می‌باشد.

لم ۲.۴ . هر بخش تولید شده توسط الگوریتم برش‌های محلی، دارای وزنی کمتر از αR می‌باشد. به عبارت دیگر $M < \alpha R$ است.

در[۳] ثابت می‌شود در حالتی که سرعت پردازنده‌ها برابر بوده و برای زمان‌بندی از الگوریتم اِل‌پی‌تی استفاده شود، بهترین ضریب تقریب برای الگوریتم دومرحله‌ای به ازای α برابر $۳+\sqrt{۱۷} \approx ۳/۵۶$ به دست می‌آید که در این حالت بدترین ضریب تقریب همواره کوچکتر یا مساوی ۳٫۵۶ خواهد بود. می‌توان با تغییر مقدار α الگوریتم را به گونه‌ای تغییر داد، که افزون بر آن که از ضریب تقریب مناسبی برای زمان‌بندی بر روی پردازنده‌هایی با سرعت متفاوت برخوردار باشد، در بسیاری از حالاتی که سرعت پردازنده‌ها برابر است، دارای حداکثر ضریب تقریب کمتری نسبت به الگوریتم قبل باشد.

قضیه ۱.۴ . ضریب تقریب الگوریتم بخش‌بندی / اِل‌پی‌تی که از الگوریتم برش‌های محلی با مقدار $\alpha = \frac{۱}{۲}(۳ - \frac{۱}{p} + \sqrt{۱۷ + \frac{۱}{p^۲} - \frac{۱۰}{p}})$ برای α، استفاده کند، همواره کمتر از ۳/۵۶ می‌باشد.

اثبات. همان‌گونه که گفته شد برای هزینهٔ انتقال کل ناشی از بخش‌بندی برش‌های محلی با ضریب ورودی α داریم $C \leq \frac{۲}{\alpha-۱}W$. بنابراین:

$$L = W + C \leq \frac{\alpha+۱}{\alpha-۱}W$$

و چون $W \leq L_{opt}$ است پس:

$$L/L_{opt} \leq \frac{\alpha+۱}{\alpha-۱}$$

از سوی دیگر $M < \alpha R$ می‌باشد، پس طبق قضیهٔ بخش قبل برای ضریب تقریب الگوریتم بخش‌بندی / اِل‌پی‌تی حاصل داریم:

$$\hat{T}/T^* \leq \max(\alpha, \frac{\alpha+۱}{\alpha-۱}(۲ - \frac{۱}{p}))$$

ولی این مقدار زمانی بر حسب α کمینه می‌شود که $\alpha = (۲ - \frac{۱}{p})(\alpha + ۱/(\alpha - ۱)$ شود با حل معادله $\alpha = \frac{۱}{۲}(۳ - \frac{۱}{p} + \sqrt{۱۷ + \frac{۱}{p^۲} - \frac{۱۰}{p}})$ خواهد شد که برای آن داریم:

$$\alpha < \frac{۱}{۲}(۳ + \sqrt{۱۷}) \approx ۳/۵۶$$

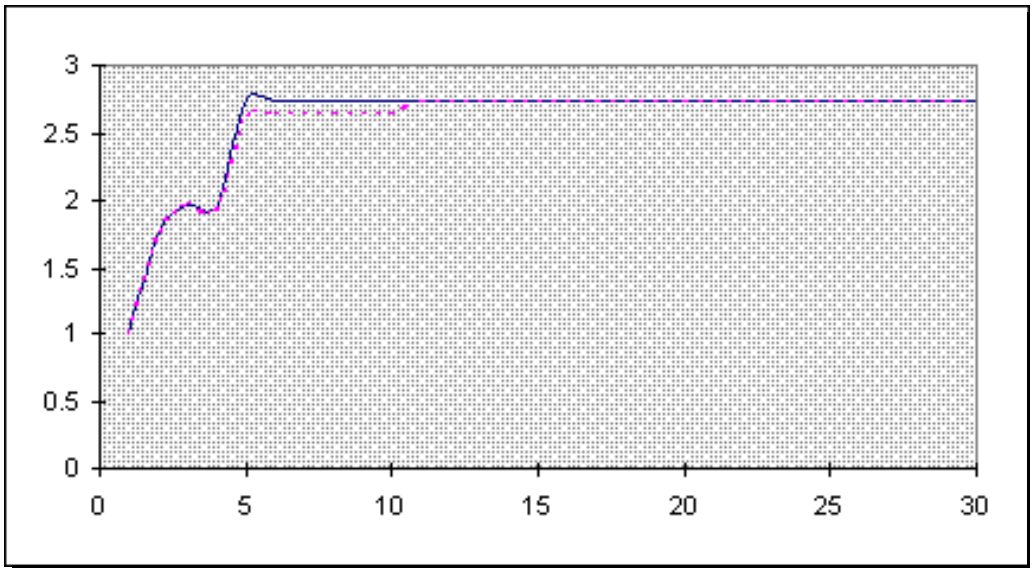
که برابر با حداکثر ضریب تقریب الگوریتم نیز می‌باشد. بنابراین ضریب تقریب الگوریتم همواره کمتر از ۳/۵۶ خواهد شد. □

واضح است که در عمل به دلیل کوچک بودن p مقدار حداکثر ضریب تقریب به عدد ۳ بسیار نزدیک خواهد بود. پس این الگوریتم، که آن را برش‌های محلی گسترش‌یافته می‌نامیم؛ در هر حالتی دارای حداکثر ضریب تقریب کمتری نسبت به الگوریتم برش‌های محلی می‌باشد.

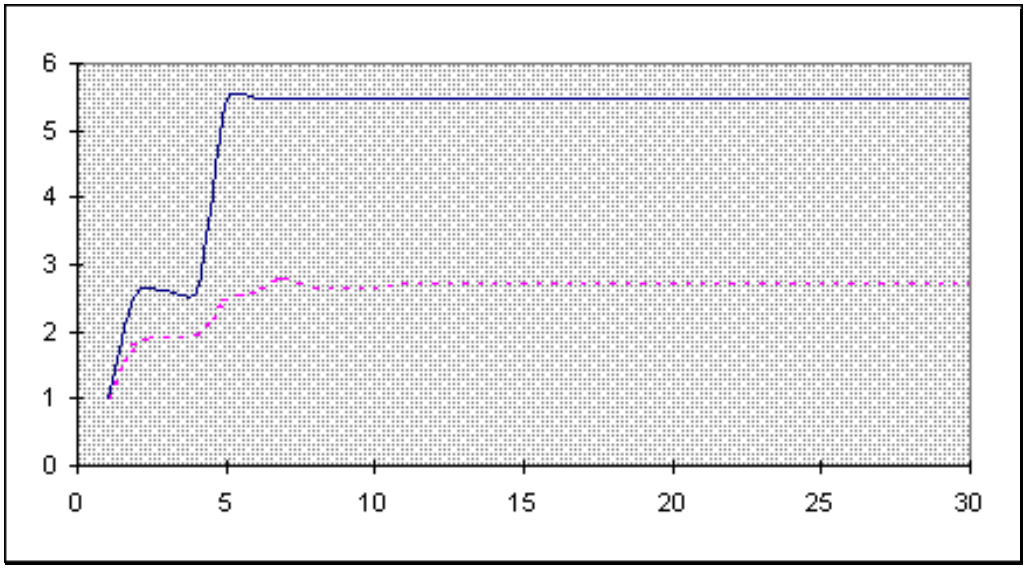
۵ ارزیابی تجربی

اگر چه در تحلیل الگوریتم‌های تقریبی، می‌توان حداکثر ضریب تقریب الگوریتم را به صورت نظری تحلیل کرد، ولی برای تحلیل ضریب تقریب متوسط الگوریتم در اغلب موارد به ارزیابی‌های عملی نیاز داریم[۸]. ارزیابی‌های انجام گرفته در این بخش با استفاده از روش شبیه‌سازی ارائه شده توسط حسن در [۸]، انجام گرفته است. هر نمودار ارائه شده در این بخش، میانگین نتایج اِعمال الگوریتم‌ها بر ۲۵۰۰ درخت عملگر یکنوا می‌باشد. کلیهٔ ارزیابی‌های نمایش داده شده در این بخش بر روی درخت‌های عملگر یکنوا با حداکثر ۲۵ گره، محدودهٔ وزن گره‌ها و یال‌های ۱۰۰ ... ۱ و با حداکثر سه فرزند برای هر گره، انجام شده است. ارزیابی‌های انجام شده با سایر مقادیر برای اندازه و ساختار درخت ورودی تفاوت قابل ملاحظه‌ای را در نمودارهای حاصل نشان ندادند. ازآنجا که محاسبهٔ یک زمان‌بندی بهینه برای این تعداد از درخت‌های عملگر ورودی نیاز به زمانی بسیار طولانی دارد، که در مواردی این کار را غیر ممکن می‌سازد، کلیهٔ ضرایب تقریب به دست آمده در این ارزیابی به توجه با حدود پایین پاسخ بهینه، محاسبه شده‌اند.

نمودار ۳ ضریب تقریب متوسط الگوریتم‌های برش‌های محلی / اِل‌پی‌تی و برش‌های محلی گسترش‌یافته / اِل‌پی‌تی را برای زمان‌بندی بر روی یک تا سی پردازنده، نشان می‌دهد. چنان که ملاحظه می‌شود، نسخهٔ گسترش‌یافتهٔ الگوریتم دارای



شکل ۳: مقایسهٔ ضریب تقریب متوسط الگوریتم‌های برش‌های محلی (خط پر) و برش‌های محلی گسترش‌یافته (خط چین) در سرعت‌های برابر



شکل ۴: مقایسهٔ ضریب تقریب متوسط الگوریتم‌های برش‌های محلی (خط پر) و برش‌های محلی گسترش‌یافته (خط چین) در سرعت‌های متفاوت با مقدار تفاوت ۲

ضریب تقریب متوسط نسبتاً کمتری از آن به ویژه هنگامی که تعداد پردازنده‌ها کمتر است، می‌باشند.

برای مقایسهٔ الگوریتم‌ها در هنگامی که پردازنده‌ها دارای سرعت‌های متفاوتی هستند، حالتی در نظر گرفته شده است که در آن تفاوت میان سریع‌ترین و کندترین پردازنده ۲ واحد است. همان‌گونه که در شکل ۴ مشاهده می‌شود، الگوریتم برش‌های محلی بدون در نظر گرفتن سرعت پردازنده‌ها در زمان‌بندی، ضریب تقریب متوسط فوق العاده بالایی نسبت به الگوریتم برش‌های محلی گسترش‌یافته دارد.

٦ نتیجه‌گیری و سوی کارهای آینده

در این مقاله تعمیمی از مسألهٔ زمان‌بندی درخت عملگر لوله‌ای برای محیط‌های ناهمگون ارائه شد. در این حالت تعمیمی از روش بخش‌بندی/ زمان‌بندی [۳]؛ و بر مبنای این روش یک الگوریتم برای حل مسألهٔ جدید ارائه شده‌اند. حداکثر ضریب تقریب الگوریتم جدید همواره کمتر از حداکثر ضریب تقریب الگوریتم قبلی مشابه با آن حتی در حالتی که پردازنده‌ها دارای سرعت‌های برابر هستند؛ می‌باشد. الگوریتم‌های ارائه شده را می‌توان در کاربردهای دیگری از زمان‌بندی عملگرها، غیر از بهینه‌سازی پرس‌وجو، نیز استفاده کرد. گرچه الگوریتم ارائه شده ضریب تقریب قابل قبولی برای حل مسأله دارد، ولی یافتن الگوریتم‌هایی با ضریب تقریب پایین‌ها و یا حتی اثبات وجود یک حد پایین برای حل تقریبی مسأله؛ کماکان یک مسألهٔ باز است.

مراجع

[1] *Bamford, R., Bulter, D., and Klots, B., "Architechture of Oracle Parallel Server,"* In Proceedings of 24th VLDB Conference , pp. 659-670, New York City, NY, USA, August 1998

[2] *Brunie, L., Exbrayat, M., and Flory, A., "Parallel Evaluation of Relational Queries on a Network of Workstations,"* Research Report N RR1999-22, University of Lyon, Department of Computer Science. March 1999.

[3] *Chekuri, C., Hasan, W., and Motwani, R., "Scheduling Problems in Parallel Query Optimization,"* In Proceedings of Principle of Database Systems, August 1996.

[4] *Chen, M. S., Wolf, J., L., and Yu, P. S, "A Hierachical Approach to Parallel Multiquery Schduling,"* IEEE Transaction on Parallel and Distributed Systems. No. 6, June 1995.

[5] *Florescu, D., Hasan, W., and Valdurize, P., "Open Issues in Parallel Query Optimization"*. Sigmod Record, Vol 28, pp.123-131, September 1997.

[6] *Garofalakis, M., Ioannidis, Y. E., and Ozdan, B., "Resource Scheduling for Composit Multimedia Objects,"* In Proceedings of 24th International Conference on Very Large Databases, August 1998.

[7] *Gonzalez, T., Ibarra, O., and Sahni, S., "Bounds for LPT Schedule on Uniform Processors.* SIAM Journal on Computing Vol 6, No 1, pp.150-164, March 1977.

[8] *Hasan, W., "Optimization of SQL Queries for Parallel Mechines,"* PhD. Thesis, Stanford University, Department of Computer Science, December 1996.

[9] *Hasan, W., Personal Communication*, Hasan@dbwizard.com, September 1999.

[10] *Hasan, W., and Motwani, R., "Optimization Algorithms for Exploiting the Parallelism, Comunication Tradeoff in Pipelined Parallelism,"* In Proceedings of International Conference on Very Large Database, pp.36-47, Santigo, September 1995.