

پردازش موازی الگوریتم ضرب ماتریس ها با استفاده از واسط تبادل پیام^۱ در شبکه ad hoc

سولماز فرشیدپور

دانشجوی کارشناسی ارشد مهندسی نرم افزار، دانشگاه علوم و تحقیقات واحد کرمان
farshidpour_s@yahoo.com

چکیده

تبادل پیام، یک الگویی مورد استفاده بسیار گسترده روی برخی ساختارهای ماشین های موازی است. تبادل پیام در واقع یک مدل محاسباتی است که پروسه ها توانایی ارتباط با دیگر پروسه ها را بوسیله ارسال و دریافت پیغام ها دارند. برای الگویی تبادل پیام استاندارد طراحی و پیاده سازی شد. این استاندارد به نام واسط تبادل پیام معرفی شد.

در کارهای انجام شده گذشته، پردازش موازی الگوریتم ضرب ماتریس بر روی شبکه سیمی اجرا شده است اما در این مقاله به منظور سهولت کار و عدم نیاز به شبکه سیمی، ابتدا شبکه بی سیم ad hoc متشکل از چند کامپیوتر راه اندازی شده و سپس الگوریتم ضرب ماتریس ها توسط واسط تبادل پیام و استفاده از استاندارد MPICH^۱، در این شبکه به صورت موازی پردازش می شود. در پایان مقاله نتیجه حاصل از پردازش موازی الگوریتم ضرب ماتریس ها در این شبکه بی سیم ad hoc بررسی شده است.

کلمات کلیدی

پردازش موازی، واسط تبادل پیام، شبکه ad hoc

¹ Message passing interface

۱- مقدمه

تبادل پیام ، يك الگوي مورد استفاده بسيار گسترده روي برخي ساختارهاي ماشين هاي موازي است . تبادل پیام در واقع يك مدل محاسباتي است كه پروسه ها توانايي ارتباط با ديگر پروسه ها را بوسيله ارسال و دريافت پيغام ها دارند كه موارد استفاده آن در سيستم هاي حافظه توزييعي مانند كلاستر ها و ماشين هاي موازي حجيم و در سيستم هاي حافظه اشتراكي مانند ابر رايانه ها

مي باشد. در دهه ۱۹۸۰ محيط هاي تبادل پیام متنوعي ايجاد شد . بعضي از آنها براي سيستم هاي خاص منظوره اي مانند مكعب N بعدي كلنگ ايجاد شده بودند و بعضي ديگر براي شبكه هايي از ايستگاه يونيكس طراحي شده بودند مثل كتابخانه هاي $P4$ آراگون و $PICL$ و PVM . تا سال ۱۹۹۲ همه طراحان اين محيط ها و كتابخانه ها به نوعي تلاش همديگر را تكرر مي کردند. تا بالاخره در كنفرانسي در سال ۱۹۹۲ تصميم گرفته شد تا براي الگوي تبادل پیام يك استاندارد كه شامل همه تجربيات و ايده شركت هاي مختلف بود، طراحي و پياده سازي شود . اين استاندارد به نام واسط تبادل پیام معرفي شد.

در ابتدا يك طرح پيشنهادي اوليه به اسم $MPI-1$ در ن و امبر ۱۹۹۲ ارائه شد كه بعدا نوع تجديد نظر شده آن در سال ۱۹۹۳ عرضه گرديد . واسط تبادل پیام در سال هاي بعد به نسخه هاي ديگري مثل ۱.۰ و ۲ گسترش يافت كه نسخه ۲، نقص هاي نسخه ۱ را برطرف كرد و شامل اضافاتي مثل توليد پروسه ، ارتباطات يك طرفه مديریتی ، عمليات گروهی گسترش يافته ، رابط هاي خارجي ، ورودی/خروجی و ... مي باشد. مزايای اصلی بنا نهادن يك استاندارد تبادل پیام ، ايجاد قابليت انتقال و راحتي استفاده از آن است . علاوه بر اين توصيف يك استاندارد تبادل پیام ، يك مجموعه از روتين هايي كه مي توانند بصورت كارا اجرا شوند را در اختيار توليد كنندگان و شركت ها قرار مي دهد كه توسط آن ها براي توليد سخت افزارهايشان

و همچنين توسعه مقياس پذيري طرح هايشان مورد استفاده قرار مي گيرد .

استاندارد تبادل پیام داراي پياده سازي هاي مختلفي مي باشد كه ليست آن ها در زير آمده است :

- MPICH
- MP-MPICH
- MPICH-G و MPICH-G2
- LAM-MPI
- CHIMP
- WMPI
- SCAMPI
- MPI/PRO

همچنين بر روي سيستم هاي حافظه توزييعي نيز نسخه اي با نام $openMP$ بوجود آمد. واسط تبادل پیام يك توصيف براي كتابخانه هاي تبادل پیام است و به اين دليل طراحي شده كه يك استاندارد براي حافظه توزييعي ، تبادل پیام و محاسبه موازي باشد. محيط كاري ما در اينجا يك سيستم حافظه توزييعي است كه شامل شبكه هاي ناهمگون است. برنامه نويس مسئول تشخيص موازي سازي درست و اجراي الگوريتم با استفاده از ساختار MPI مي باشد. تعداد وظايف اختصاص داده شده براي اجراي يك برنامه موازي اي ستا مي باشد . وظايف جديد نمی توانند بطور پويا و در طول زمان اجرا توليد شوند.

در اين مقاله شبكه بي سيم $ad hoc$ متشكل از چند كامپيوتر راه اندازي مي شود. سپس با استفاده از استاندارد $MPICH$ و واسط تبادل پیام ، الگوريتم ضرب ماتريس ها بصورت موازي پردازش مي گردد و در آخر نتايج حاصل از پردازش بيان مي شود .

۲- كارهاي انجام شده پيشين

در مقايسه با پردازش سريال ، پردازش الگوريتم ها بصورت موازي و توسط تعداد مناسبي از پردازنده ها در شبكه ، باعث

افزایش ضریب تسریع و کارایی می‌شود.

در مقالات گذشته، الگوریتم ضرب ماتریس‌ها توسط واسط تبادل پیام و بر روی شبکه سیمی متشکل از تعدادی کامپیوتر، بصورت موازی پردازش شده است.

در این مقاله ایده جدیدی پیشنهاد داده ام. ایده پیشنهادی من، استفاده از شبکه بی سیم *ad hoc* برای پردازش الگوریتم ضرب ماتریس‌ها می‌باشد. در این شبکه بی سیم مانند قبل، برای پردازش موازی الگوریتم از واسط تبادل پیام استفاده کرده ام. در ادامه مقاله مراحل انجام کار و نتایج بدست آمده را شرح می‌دهم.

۳-مراحل انجام کار

در ابتدا، شبکه بی سیم *ad hoc* متشکل از ۵ کامپیوتر راه اندازی می‌کنیم. راه اندازی این شبکه آسان می‌باشد. تمام کامپیوترهای موجود در شبکه می‌بایست مجهز به کارت شبکه بی سیم بوده سپس با انجام تنظیمات ساده و مراحل نرم افزاری، شبکه راه اندازی می‌گردد.

همانطور که اشاره شده *MPI* یک زبان نیست بلکه یک کتابخانه از توابع به منظور نوشتن برنامه های موازی است و می‌بایست هدر فایل های آن را در سورس برنامه معرفی نمود. فرمت توابع *MPI* که بهنگام برنامه نویسی بکار گرفته می‌شود بصورت زیر است:

MPI_Xxxx(parameter,...);

در این جا عملیات ضرب ماتریس انجام می‌شود. به منظور تمایز پروسس های مختلف اولین روتین که بکار می‌رود *MPI_Init* است. این روتین یک گروه یا *Communicator* ایجاد می‌کند و به تمام پردازنده هایی که در یک گروه هستند شماره ای اختصاص می‌دهد. به این شماره اصطلاحاً "*Rank*" گفته می‌شود. بصورت پیش فرض *Master* شماره *Rank* صفر را می‌گیرد و بقیه پردازنده ها به ترتیب شماره گذاری می‌شوند. در واقع *Rank* به پروسه *ID* است.

الگوریتم کار بدین صورت است که بعد از تعریف متغیرها ی مورد نیاز، زمان شروع برنامه را ثبت می‌کنیم سپس برای نمایش شماره و نام ماشین پردازش گر از فرمان *Fprintf* استفاده کرده ایم که در واقع *MyID* و *processor name* را بر می‌گرداند. سپس تعداد سطر و ستون ماتریس مورد نظر را تعریف می‌کنیم. این مقادیر ثابت می‌باشند و انجام تغییرات آن منوط به دستکاری در سورس برنامه است. در واقع دو ماتریس ۱۰۰۰ در ۱۰۰۰ در این برنامه در هم ضرب شده اند. که بصورت ریاضی بصورت $1000 \times 1000 \times 1000$ در هم ضرب می‌شود. در ادامه بعد از تخصیص حافظه مورد نیاز ماتریسها را بصورت تصادفی مقدار دهی می‌کنیم و در بخش آخر عملیات ضرب ماتریسها انجام می‌شود. برای این منظور ابتدا توسط تابع *MPI_Bcast* اندازه هریک از ماتریسها برای بقیه گره ها معرفی می‌کنیم. در واقع در شرط اول فقط اطلاعات ماتریس ها برای *ID* صفر و رد شرط دوم اطلاعات مورد نیاز سایر گره ها مشخص می‌شود. در بخش عمل ضرب ماتریسها بدین صورت انجام می‌شود هر پروسس یک سطر از ماتریس را در کل ماتریس بعد ضرب می‌کند.

این برنامه در یک محیط شبکه *Workgroup* با ۵ سیستم اجرا شده است. مشخصات این سیستم ها طبق جدول (۱) است.

جدول (۱) : مشخصات کامپیوتر های شبکه

نام ماشین	نوع پردازنده	میزان حافظه اصلی
Laptop1	1.6 centrino	512MB DDR
Laptop2	1.83 dual core napa	512 MB DDR2/533
Laptop3	3000 full cache LGA	512 MB DDR2/677
Laptop4	3000 full cache LGA	512 MB DDR2/677
Server HP) Laptop5 (ML370	Dual Xeon 3.6 2MB L2	2GB DDR2/677 ECC

ارتباطات زمان انجام پردازش سیر صعودی به خود می گیرد. همچنین همیشه با افزایش تعداد پردازنده ها زمان پردازش افزایش نمی یابد و به علت وجود سر بار سیستم بصورت None dedicate عمل خواهد کرد.

مراجع

- [1] V. Bala and S. Kipnis. Process groups: a mechanism for the coordination of and communication among processes in the Venus collective communication library. Technical report, IBM T. J. Watson Research Center, October 1992. Preprint.
 - [2] V. Bala, S. Kipnis, L. Rudolph, and Marc Snir. Designing efficient, scalable, and portable collective communication libraries. In SIAM 1993 Conference on Parallel Processing for Scientific Computing, pages 862-872, March 1993.
 - [3] Luc Bomans and Rolf Hempel. The Argonne/GMD macros in FORTRAN for portable parallel programming and their implementation on the Intel iPSC/2. Parallel Computing, 15:119-132, 1990.
 - [4] J. Bruck, R. Cypher, P. Elustond, A. Ho, C-T. Ho, V. Bala, S. Kipnis, , and M. Snir. Ccl: A portable and tunable collective communication library for scalable parallel computers. IEEE Trans. on Parallel and Distributed Systems, 6(2):154-164, 1995.
 - [5] R. Butler and E. Lusk. User's guide to the p4 programming system. Technical Report TM-ANL-92/17, Argonne National Laboratory, 1992.
 - [6] Ralph Butler and Ewing Lusk. Monitors, messages, and clusters: the p4 parallel programming system. Journal of Parallel Computing, 20(4):547-564, April 1994.
 - [7] Robin Calkin, Rolf Hempel, Hans-Christian Hoppe, and Peter Wypior. Portable programming with the parmacs message-passing library. Parallel Computing, 20(4):615-632, April 1994.
 - [8] S. Chittor and R. J. Enbody. Performance evaluation of mesh-connected wormhole-routed networks for interprocessor communication in multicomputers. In Proceedings of the 1990 Supercomputing Conference, pages 647-656, 1990.
 - [9] S. Chittor and R. J. Enbody. Predicting the effect of mapping on the communication performance of large multicomputers. In Proceedings of the 1991 International
- همانطور که از جدول (۱) پیداست این ۵ سیستم برای اجرای ماتریس 1000×1000 در نظر گرفته شده اند. با اجرای مرحله به مرحله برنامه زمان اجرای پردازش ثبت شده است. برای اینکه بتوانیم برنامه ها را بصورت موازی اجرا نماییم می بایست در ابتدا MPICH را در تمامی ماشینها نصب کنیم. بعد از نصب برای اینکه تمامی سیستم ها با هم کار کنند اولاً یک کاربر مشترک با رمز عبور یکسان باید تعریف گردد و ثانیاً در صورتیکه Firewall روی کارت شبکه سیستم فعال است باید غیر فعال گردد. در مرحله بعد با کاربر و رمزی که برای آن تعریف کرده ایم MPI register را اجرا می کنیم تا رجیستر گردد. سپس فایل کمپایل شده (فایل اجرایی برنامه) در تمامی کامپیوترهای شبکه در یک پوشه که نام آن در تمامی سیستم ها یکسان است و به اشتراک گذاشته شده کپی می کنیم سپس از طریق MPIch Configure tool لیست ماشینهای شبکه را اضافه می کنیم و برنامه را در ماشین Master اجرا می کنیم.

۴- نتیجه

در اولین مرحله با یک کامپیوتر بیشترین زمان انجام محاسبه ثبت شده است. با افزایش تعداد پردازنده ها تا ۵ عدد کاملاً زمان پردازش بهینه شده و در کمترین مقدار خود قرار گرفته است. اما با زیاد کردن تعداد کامپیوترها چون هر پردازنده ممکن بوده است بیشتر از یک عمل را انجام دهد به علت وجود پروسه های مختلف برای تقسیم کار و همچنین به علت وجود سر بار در

- In 15th International Conference on Distributed Computing Systems, pages 127-135, June 1995.
- [17]A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. PVM: A Users' Guide and Tutorial for Networked Parallel Computing. MIT Press, 1994. The book is available electronically, the url is <ftp://www.netlib.org/pvm3/book/pvm-book.ps>.
 - [18]G. A. Geist, M. T. Heath, B. W. Peyton, and P. H. Worley. A user's guide to PICL: a portable instrumented communication library. Technical Report TM-11616, Oak Ridge National Laboratory, October 1990.
 - [19]William D. Gropp and Barry Smith. Chameleon parallel programming tools users manual. Technical Report ANL-93/23, Argonne National Laboratory, March 1993.
 - [20]V. Karamcheti and A.A. Chien. Software overheads in messaging layers: Where does the time go? In 6th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS VI), pages 51-60, October 1994.
 - [21]O. Krämer and H. Mühlenbein. Mapping strategies in message-based multiprocessor systems. *Parallel Computing*, 9:213-225, 1989.
 - [22]nCUBE Corporation. nCUBE 2 Programmers Guide, r2.0, December 1990.
 - [23]Paraso Corporation, Pasadena, CA. Express User's Guide, version 3.2.5 edition, 1992.
 - Conference on Parallel Processing, vol. II (Software), pages II-1 - II-4, 1991.
 - [10]R. Cypher and E. Leu. The semantics of blocking and nonblocking send and receive primitives. In 8th International Parallel Processing Symposium, pages 729-735, April 1994.
 - [11]J. J. Dongarra, R. Hempel, A. J. G. Hey, and D. W. Walker. A proposal for a user-level, message passing interface in a distributed memory environment. Technical Report TM-12231, Oak Ridge National Laboratory, February 1993.
 - [12]Nathan Doss, William Gropp, Ewing Lusk, and Anthony Skjellum. A model implementation of MPI. Technical report, Argonne National Laboratory, 1993.
 - [13]Edinburgh Parallel Computing Centre, University of Edinburgh. CHIMP Concepts, June 1991.
 - [14]Edinburgh Parallel Computing Centre, University of Edinburgh. CHIMP Version 1.0 Interface, May 1992.
 - [15]Message Passing Interface Forum. MPI: A message-passing interface standard. *International Journal of Supercomputer Applications*, 8(3/4), 1994. Special issue on MPI.
 - [16]H. Franke, H. Wu, C.E. Riviere, P.Pattnaik, and M. Snir. MPI programming environment for IBM SP1/SP2.