MPI paketini ishlashini o‘rganish

Sobirov Muzaffar Mirzaolimovich, Xodjimatov Jaxongir Murodovich
Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti Farg‘ona filiali o‘qituvchilari.

Annotatsiya. *MPI yordamida, bir nechta kompyuterlar orasida ma'lumot almashish va ulashish mumkin bo'ladi. Ushbu MPI paketlarini o'rganish mavzusi esa MPI kutubxonalari va ularni ishlatishni o'z ichiga oladi. MPI kutubxonalari paralel dasturlashning murakkab va ko'p-yordamli xususiyatlarini ta'minlaydi.*

Kalit so‘zlar: *hisob, parallel dasturlash, murakkablik, kutubxona, superkompyuter, MPI.*

Ko'pincha, eng murakkab algoritmlar real vazifalarni bajarishda katta miqdordagi hisoblash resurslarini talab qiladi, agar dasturchi protsessual yoki ob'ektga yo'naltirilgan dasturlash (OOP) standart tushunchasida kod yozsa, u holda katta hajmdagi algoritmik vazifalarni bajaradi. ma'lumotlar va vazifani bajarish vaqtini kamaytirishni talab qiladi, optimallashtirishni amalga oshirish kerak.

Hisob - kitoblar protsessorda amalga oshiriladi, protsessor registrlar deb nomlangan maxsus ma'lumotlarni "saqlaydi". Protsessor registrlari mantiqiy elementlarga to'g'ridan to'g'ri bog'langan va operatsiyalarni bajarish uchun operativ xotiradagi ma'lumotlarga qaraganda ancha kam vaqt talab etiladi, hatto qattiq diskda ham, chunki ikkinchisi uchun ma'lumotlarni uzatish vaqtning katta qismini oladi. Shuningdek, protsessorlarda Cache deb nomlangan xotira maydoni mavjud bo'lib , u hozirda hisob - kitoblarda qatnashadigan yoki yaqin kelajakda ishtirok etadigan qiymatlarni, ya'ni eng muhim ma'lumotlarni saqlaydi.

Algoritmni optimallashtirish vazifasi operatsiyalar ketma - ketligini to'g'ri tuzish va ma'lumotlarni Keshda maqbul joylashtirish, xotiradan mumkin bo'lgan ma'lumotlarni uzatish sonini kamaytirishdir.

Xo'sh, vektorlashtirish va parallel hisoblash o'rtasidagi farq nima?

Vektorlashtirish - berilgan butun sonli vektorlar bo'yicha amallarni bajarishga imkon beradi. Misol uchun, uchun C++: Biz foydalanish, agar AVX ko'rsatmalarga, keyin biz float32 elementlar qo'yish mumkin 256-bit vektorlar bor. Ma'lum bo'lishicha, biz (256/32) = 8 float32 qiymatining 2 vektorini yig'ishimiz mumkin. Shundan so'ng, har bir tsikl uchun bitta operatsiyani bajaring, garchi biz buni qilmagan bo'lsak ham, agar biz aniq elementlar indekslarini hisobga olsak, shunga o'xshash hisob - kitoblar 8 ta operatsiyani bajarishi mumkin. Bunday optimallashtirish bilan juda yaxshi kompilyatorlar endi juda aqlli va o'zlari, shuning uchun kuch eng ketadi: Ammo, bir muammo bor aslida to'g'ri ma'lumotlarni tashkil qilish, deb.

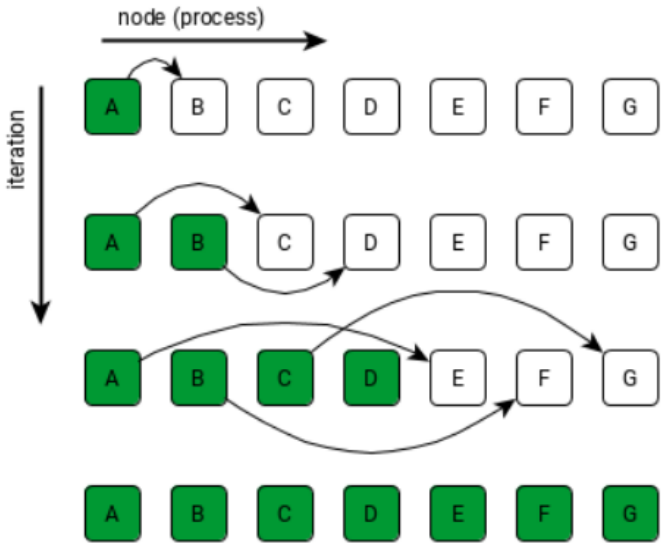
Boshqa tomondan, parallelizatsiya sizga bitta protsessorning hisoblash imkoniyatlaridan emas, balki ko'plab protsessorlardan, shu jumladan xotira taqsimlangan tizimlarda va boshqalardan foydalanishga imkon beradi. Bunday tizimlarda, qoida tariqasida, hisoblash kuchi ko'proq bo'ladi, lekin bularning barchasi ma'lum bir chiziqlar bo'yicha parallel ishlarni boshqarishi va to'g'ri tashkil etishi kerak.

Hozirgi vaqtda parallel hisoblashni tashkil etishni osonlashtiradigan ko'plab texnologiyalar mavjud, hozirgi vaqtda eng ommaboplaridan biri bu MPI . MPI - xabarlarni uzatish interfeysi . Bu ism semantik xarakterga ega bo'lib, keyingi maqolalarda aniq ifodalanadi, lekin hozircha siz bilishingiz kerak bo'lgan narsa - bunday tizimdagi parallel jarayonlarning o'zaro ta'sirining asosiy usuli - ular orasidagi xabarlarni uzatish. Bu shuni anglatadiki, bizning tarmoqlarimiz ma'lumotlarni qayta ishlashda bir -biri bilan aloqa o'rnatadilar, lekin bu qanday amalga oshirish masalasidir.

Parallel dasturlarni yaratishda ikkita asosiy uslub mavjud: MIMD (Multiple Instruction Multiple Data) va SPMD (Single Program Multiple Data).

Juda soddalashtirish uchun, birinchi model har xil dastur manba kodlarini oladi va ularni har xil ish zarrachalarida bajaradi, ikkinchi model bitta manba kodini oladi va uni barcha tanlangan iplar ustida ishlaydi. MIMD uslubida yozilgan dasturlarni arxitekturasi tufayli disk raskadrovka qilish juda qiyin,

shuning uchun ko'pincha SPMD modeli ishlatiladi . Yilda MPI, u har ikkala foydalanish mumkin, lekin standart (albatta, amalga oshirish qarab) foydalanadi SPMD modelini.



1-rasm. MPI_Reduce va MPI_Bcast ning samarali amalga oshirilishi

Har bir muhitda u kompilyatorga bog'liq va u tomonidan amalga oshiriladi . Shaxsan men Ubuntu Budgie 20.04 LTS -da ishlaganman va sizga uni o'rnatish bo'yicha ko'rsatmalarni aytib beraman.

Buyruq satriga quyidagi buyruqlarni kiriting:

[user - name] $ sudo apt- get update

[user - name] $ sudo apt- get gcc ni o'rnating

[user - name] $ sudo apt- get mpich

Birinchi buyruq paket menejerini yangilaydi, ikkinchisi GCC kompilyatorini o'rnatadi , agar u tizimda bo'lmasa, uchinchi dastur kompilyatorni o'rnatadi, u orqali biz aslida C \ C ++ va MPI kodlari bilan ishlaymiz .

MPI dasturi - bu o'zaro ta'sir qiluvchi jarayonlar majmui bo'lib, ularning har biri o'z ajratilgan xotira maydonida ishlaydi. Aslida, bu ish jarayonida bir -biri bilan muloqot qiladigan N mustaqil dasturlar. Yilda MPI, eng ma'lumotlar turlari allaqachon qayta tavsiflaydi va qisqartmasi bilan boshlanadi MPI_ [nomi] yanada aniq bo'ladi.

Tushunish uchun bir nima sodir bo'ladi, keyin bir necha shartlarni aniqlash kerak:

Communicator - bir ob'ekt deb bola jarayonlar ma'lum bir guruh orqali muloqot qilib. C ++ / C da bu MPI_Comm ma'lumotlar turi . Kommunikator bir nechta jarayonlarni bir - biriga yuborish orqali birlashtirishi mumkin, bir nechta kommunikatorlar bo'lishi mumkin, ular tuzgan guruhlar bir - birining ustiga chiqa olmaydi yoki qisman bir - birining ustiga chiqa olmaydi. Dastur ishga tushganda, barcha jarayonlar MPI_COMM_WORLD nomli bitta kommunikator ostida ishlaydi.

Bunga qo'shimcha ravishda, MPI_COMM_SELF, MPI_COMM_NULL kommunikatorlari ham bor, ular mos ravishda faqat joriy jarayonni o'z ichiga oladi va jarayonlar yo'q. Xabar - bu jarayonlar o'rtasidagi aloqa paytida uzatiladigan ma'lum turdagi ma'lumotlar to'plami. Har bir xabar bir nechta atributlarga ega, xususan, jo'natuvchi jarayoni, qabul qiluvchi, xabar identifikatori, kommunikator va teg. Xabar yorlig'i 0 dan 32767 gacha bo'lgan manfiy bo'lmagan tamsayıdir (Amalga bog'liq. Teg mumkin bo'lgan teg qiymati MPI_TAG_UB doimiyida saqlanadi).

Bu misolda biz ma'lumotlarni uzatishdan foydalanmaymiz, faqat MPI protseduralari asoslari bilan tanishamiz.MPI protseduralarining ko'pini istalgan dasturda ishlatishga imkon berish uchun quyidagi tartiblar bo'lishi kerak:

int MPI_ Init (int * argc, char *** argv) ;

int MPI_Finalize (bekor) ;

Birinchi protsedura dasturning parallel qismini ishga tushirish uchun mo'ljallangan, boshqa maxsus protseduralar bundan mustasno, faqat MPI_Init protsedurasi chaqirilgandan so'ng chaqirilishi mumkin , bu funksiya buyruq qatori argumentlarini qabul qiladi, ular orqali tizim ba'zi ishga tushirish parametrlarini o'tkazishi mumkin. jarayonga. Ikkinchi protsedura dasturning parallel qismini bajarish uchun mo'ljallangan.

Dasturning ishini sinab ko'rish uchun biz MPI yordamida C ++ tilida eng oddiy dasturni amalga oshiramiz .

```
# o'z ichiga oladi <stdio.h>
# "mpi.h" ni o'z ichiga oladi

int main ( int argc, char ** argv)
{
  printf ( "MPI_INIT \ ndan oldin" );
  MPI_ Init ( & argc, & argv);
  printf ( "Parallel sekta \ n" );
  MPI_ yakunlash ( );
  printf ( "MPI_FINALIZE dan keyin \ n" );
  qaytarish 0 ;
}
```

Ushbu kodda, MPI_Init funksiyasi MPI-ni boshlash uchun ishlatiladi. MPI_Comm_rank va MPI_Comm_size funksiyalari orqali, har bir ish stansiya o'zi haqiqiy ish stansiya sifatida qanday tartibda ishlashi kerakligini aniqlaydi. Natijada, har bir ish stansiya o'z tartib raqami (rank) va umumiy ish stansiyalar soni (size)ni chiqaradi. MPI_Finalize funksiyasi MPI-ni yakunlash uchun ishlatiladi.

Ushbu misol C dasturlash tilida yozilgan, lekin boshqa dasturlash tillarida ham MPI-ni ishlatish mumkin (masalan, Python, Fortran va hokazo). Har dasturlash tili uchun o'ziga xos MPI qo'llanmalari mavjud bo'ladi, shuning uchun MPI-ni istalgan dasturlash tilida ishlatish mumkin.

Ushbu dasturni ishga tushirish uchun kirishni * va .cpp nomli faylga saqlang va konsolda quyidagi amallarni bajaring (mening holimda kod main.cpp faylida):

[Foydalanuvchi - nomi] $ mpic ++ main.cpp Optsiya -O Asosiy

[Foydalanuvchi - nomi] $ mpiexec -n2./main

Birinchi buyruq MPI dasturimizni kompilyatsiya qiladi, ikkinchi buyruq esa uni ishga tushirishga imkon beradi. E'tibor bering, biz - n 2 variantini ikkinchi qatorga o'tkazyapmiz, nima uchun bu? Shunday qilib, biz 2 parallel jarayonni

boshlash zarurligini ijrochiga ma'lum qilamiz.

Foydalanigan adabiyotlar.

1. Расулов, А. М., & Ходжиматов, Ж. М. (2021). ОБУЧЕНИЕ ПАРАЛЛЕЛЬНОМУ ПРОГРАММИРОВАНИЮ С ИСПОЛЬЗОВАНИЕМ JAVA.

2. Собиров, М. М., Хамидов, Э. Х., & Ходжиматов, Ж. М. (2022). Виртуальные электростанции—будущее энергетики. Journal of new century innovations, 11(1), 117-126.

3. Хамидов, Э. Х., Собиров, М. М., & Ходжиматов, М. М. (2022). НЕЙРОННЫЕ СЕТИ RNN И LSTM. Journal of new century innovations, 11(1), 127-135.

4. Садирова, Х., & Набижонов, Р. (2023). Методы создания корпоративной системы безопасности для обеспечения информационной безопасности. Journal of technical research and development, 1(2), 170-174.

5. Nabijonov, R. M. o‘g‘li, & Mamayeva, O. I. qizi. (2023). Ta’lim sifatini oshirishda elektron amaliy dasturiy paketlarning ahamiyati. GOLDEN BRAIN, 1(25), 51–55. Retrieved from https://researchedu.org/index.php/goldenbrain/article/view/4782

6. Nabijonov, R. (2022). Theories of fuzzy sets and their application in face recognition. Innovation in the modern education system.

7. Nabijonov, R. (2020). 9x9x9 ko'rinishda joylashtirilgan LED lampalarda svetomuzika dasturini loyixalash. Журнал «Студенческий вестник»№ 24 (122), часть 4, 2020 г.

8. Nabijonov, R. (2019). Network data management of communication systems. SCIENTIFIC RESEARCHES FOR DEVELOPMENT FUTURE.

9. Isaqovich, T. N., & Muxammadjon o‘g‘li, N. R. (2023). To ‘g ‘ri to ‘rtburchakda Laplas tenglamasi uchun shartli Korrekt qo ‘yilgan masala. IMRAS, 6(6), 90-94.

10. Umarovich, I. U. (2023). Ovwerview of the comparations of the main parametters of the modern television standards. PEDAGOG, 6(10), 41-47.

11. Обухов, В. А. (2023). Цифровая безопасность данных в блокчейн-сетях. PEDAGOG, 6(10), 304-308.

12. Muxammadjon o‘g‘li, N. R., Alisher o‘g‘li, A. S., & Nodirovich, N. Q. (2022, November). Qidiruv algoritmlari va ularni optimallashtirish. In Proceedings of International Conference on Educational Discoveries and Humanities (Vol. 1, No. 2, pp. 209-217).

13. Сотволдиева, Д. Б., & Хусанова, М. К. (2020). Сравнение фильтров с конечной импульсной характеристикой и бесконечной импульсной характеристикой в программе matlab. in цифровой регион: опыт, компетенции, проекты (pp. 840-845).

14. Khonturaev , S. I., Fazlitdinov , M. X. ugli, & Mamayeva , O. I. kizi. (2023). EMPOWERING EDUCATION: THE IMPACT OF AI IN LEARNING MANAGEMENT SYSTEMS. Educational Research in Universal Sciences, 2(11), 348–350. Retrieved from http://erus.uz/index.php/er/article/view/3985

15. Khonturaev , S. I., & Kodirov , A. A. ugli. (2023). REVOLUTIONIZING COTTON PICKING: THE ROLE OF AI IN AGRICULTURE. Educational Research in Universal Sciences, 2(11), 354–356. Retrieved from http://erus.uz/index.php/er/article/view/3987

16. Khonturaev , S. I., & Fazlitdinov, M. X. ugli. (2023). AI IN UZBEKISTAN: PIONEERING A TECHNOLOGICAL TRANSFORMATION. Educational Research in Universal Sciences, 2(11), 351–353. Retrieved from http://erus.uz/index.php/er/article/view/3986

17. Тошматов, Ш., Исаков, А., & Махмудов, Ш. (2023). Модульные измерительные датчики. Journal of technical research and development, 1(2), 210-218.

18. Тошматов, Ш. (2023). Работа нейронной сети. Формирования Graphic detection detection проекта в языке программирование Python определение используемых библиотек. Journal of technical research and development, 1(2), 297-305.

19. Расулов, А. М., & Тошматов, Ш. М. (2023). Создание базы данных и классификация для Graphic Detection проекта искусственного нейронного сети. CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES, 4(4), 15-21.

20. Murotzhonovich, T. S. (2023). Introduction to Artificial Neural Networks. Web of Synergy: International Interdisciplinary Research Journal, 2(4), 198-202.

21. Tolipov, N., Xudoynazarov, Q., & Munavarjonov, S. (2023). Об одной некорректной задаче для бигармонического уравнения в полушаре. Research and implementation.

22. Tolipov, N., Isaxonov, X., & Zunnunov, M. (2023). Shar tashqarisidagi soha uchun garmonik davom ettirish masalasi. Research and implementation.

23. Isaqovich, T. N. (2023). Chorak doira tashqarisida bigarmonik tenglama uchun nokorrekt qo ‘yilgan masala. Talqin va tadqiqotlar ilmiy-uslubiy jurnali, 1(18), 73-83.

24. Искандаров, У. У., & Эгамбердиев, М. М. (2018). АСПЕКТЫ И ПРОБЛЕМЫ СОЗДАНИЯ И СОДЕРЖАНИИ «УМНОГО ДОМА».

25. Абдукадиров, А. Г. (2022). ЧИСЛЕННЫЙ РАСЧЕТ РАСПРЕДЕЛЕНИЯ ФОТОНОСИТЕЛЕЙ В АМОРФНЫХ МАТЕРИАЛАХ. Евразийский журнал академических исследований, 2(6), 805-809.

26. Akbarov, D., & Abdukadirov, A. (2022, June). Research of general mathematical characteristics of logical operations and table replacements in cryptographic transformations. In AIP Conference Proceedings (Vol. 2432, No. 1). AIP Publishing.

27. Gapirovich, A. A., Mirzapo‘Latovich, E. O., & O‘G‘Li, O. S. O. (2022). KLINIK LABORATORIYALAR UCHUN MO’LJALLANGAN AVTOMATLASHTIRILGAN TIZIMLARNING TAHLILI. Central Asian Research Journal for Interdisciplinary Studies (CARJIS), 2(Special Issue 2), 163-167.