



ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ТҰҢҒЫШ ПРЕЗИДЕНТІ - ЕЛБАСЫНЫҢ ҚОРЫ

«ҒЫЛЫМ ЖӘНЕ БІЛІМ – 2017»

студенттер мен жас ғалымдардың
XII Халықаралық ғылыми конференциясының
БАЯНДАМАЛАР ЖИНАҒЫ

СБОРНИК МАТЕРИАЛОВ

XII Международной научной конференции
студентов и молодых ученых
«НАУКА И ОБРАЗОВАНИЕ – 2017»

PROCEEDINGS

of the XII International Scientific Conference
for students and young scholars
«SCIENCE AND EDUCATION - 2017»



14th April 2017, Astana



**ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Л.Н. ГУМИЛЕВ АТЫНДАҒЫ ЕУРАЗИЯ ҰЛТТЫҚ УНИВЕРСИТЕТІ**

**«Ғылым және білім - 2017»
студенттер мен жас ғалымдардың
XII Халықаралық ғылыми конференциясының
БАЯНДАМАЛАР ЖИНАҒЫ**

**СБОРНИК МАТЕРИАЛОВ
XII Международной научной конференции
студентов и молодых ученых
«Наука и образование - 2017»**

**PROCEEDINGS
of the XII International Scientific Conference
for students and young scholars
«Science and education - 2017»**

2017 жыл 14 сәуір

Астана

УДК 378

ББК 74.58

Ғ 96

Ғ 96

«Ғылым және білім – 2017» студенттер мен жас ғалымдардың XII Халықаралық ғылыми конференциясы = The XII International Scientific Conference for students and young scholars «Science and education - 2017» = XII Международная научная конференция студентов и молодых ученых «Наука и образование - 2017». – Астана: <http://www.eni.kz/ru/nauka/nauka-i-obrazovanie/>, 2017. – 7466 стр. (қазақша, орысша, ағылшынша).

ISBN 978-9965-31-827-6

Жинаққа студенттердің, магистранттардың, докторанттардың және жас ғалымдардың жаратылыстану-техникалық және гуманитарлық ғылымдардың өзекті мәселелері бойынша баяндамалары енгізілген.

The proceedings are the papers of students, undergraduates, doctoral students and young researchers on topical issues of natural and technical sciences and humanities.

В сборник вошли доклады студентов, магистрантов, докторантов и молодых ученых по актуальным вопросам естественно-технических и гуманитарных наук.

УДК 378

ББК 74.58

ISBN 978-9965-31-827-6

©Л.Н. Гумилев атындағы Еуразия
ұлттық университеті, 2017

9. <http://rushkolnik.ru/docs/10/index-13765.html?page=3>
10. <http://www.volpi.ru/umkd/zki/index.php?man=1&page=12>
11. <http://mirznanii.com/a/110761/kriptografiya-blochnyy-shifr>
12. http://otherreferats.allbest.ru/programming/00197560_0.html

ӘОЖ 004.042

КӨПАҒЫНДЫ БАҒДАРЛАМАЛАУДЫ ЗЕРТТЕУ ЖӘНЕ ҚОЛДАНУ МҮМКІНДІКТЕРІНЕ ШОЛУ

Ерболова Нұрән Арайқызы

nuran95@mail.ru

Л.Н.Гумилев атындағы ЕҰУ Ақпараттық технологиялар факультетінің «Есептеу техникасы»
кафедрасы, Астана, Қазақстан
Ғылыми жетекшісі –Ташенова Ж.М.

Бұл мақалада көпағынды бағдарламалық зерттеудің мүмкіндіктері қарастырылады. Көпағынды бағдарламалауда көпағындылықтың мағынасын ашамыз. Көпағындық қасиеті - операциялық жүйеде туындайтын процесс уақытта белгіленген ретсіз параллельді орындалатын бірнеше ағыннан тұратын платформа (мысалы, операциялық жүйе, виртуалды машина және т.с.с). Яғни, бірнеше есепті орындаған кезде мұндай бөлу есептеу құрылғысының ресурстарын тиімді қолдануға мүмкіндік береді. Мұндай ағындарды орындау ағындары (thread of execution) деп аталады. Көпағындық маңызы бір орындалатын процесс деңгейінде көпесептік болып табылады, яғни барлық ағындар процесстің адресілік кеңістігінде орындалады. Сонымен қатар, процесстің барлық ағындарында бір адресілік кеңістік, дескрипторлары болады. Орындалатын процесте кем дегенде бір (негізгі) ағыны болады. Көпағындықты көпесептік және көп процессорлықпен шатастырмау керек, бірақ көпесептілікті іске асыратын операциялық жүйелер көпағындықты да іске асырады.

Программалаудағы көпағындықтың артықшылықтары:

- Кейбір жағдайларда жалпы адресілік кеңістікті босату арқылы программаны жеңілдету;
- Ағынды құруға кететін уақыттың аз шығыны;
- Процессорлық есептеулер және енгізу-шығару операцияларын параллельдеу арқылы процесс өнімділігін арттыру [1].

Қарапайым процессорда ағындарды басқару операциялық жүйемен іске асырылады. Аппараттық үзу, жүйелік шақыру немесе сол ағынға бөлінген уақыт аяқталғанға дейін ағын орындалады. Осыдан кейін процессор ағын жағдайын сақтайтын операциялық жүйе кодына ауысады немесе, басқа ағын күйіне ауысады. Мұндай көпағындық жағдайында процессордың көп такттер саны контекстерді (бір есептен екінші есепке ауыстыру) ауыстыратын операциялық жүйелер кодына жұмсалады. Егер ағындарды қолдауды аппаратты түрде іске асырса, процессор өзі ағындар арасында ауысу жасап отыра алады, кейбір жағдайларда бірнеше ағындарды әрбір тактте біруақытты орындай алады.

Процессорларда аппаратты іске асырылатын 2 көпағындық түрі болады:

- уақытша көпағындық;
- біруақытты көпағындық.

Ағындарды іске асыру типтері:

- Қолдану кеңістігіндегі ағын. Әрбір процессте ағындар кестесі болады, ядро процестері кестесіне ұқсас. Кемшіліктері:
 1. Бір процесс ішінде таймер бойынша үзудің болмауы.
 2. Процесте бұғаттау жүйелік сұраныс қолданғанда оның барлық ағындары бұғаттанады.
 3. Іске асыру қиындығы.

- Ядро кеңістігіндегі ағын. Процестер кестесімен қатар ядро кеңістігінде ағындар кестесі болады.

- «Талшықтар» (fibers). Ядроның бір режимінде орындалатын қолданушы режимінің бернеше ағындары. Ядро кеңістігінің ағыны көп ресурстар қолданады, ең алдымен физикалық жады және ядро режимінің адрестерінің диапазоны. Сондықтан «талшық» түсінігі еңгізілген - тек қолданушы режимінде орындалатын жеңілдетілген ағын. Әрбір ағында бірнеше талшық болуы мүмкін [2].

Параллельдік есептеулер – өзара байланысатын есептеу процестерінің жиыны ретінде программалары құрылатын компьютерлік есептеулерді ұйымдастыру әдісі. Термин программалаудағы параллельдік және тиімді аппараттық іске асыруды құруды қамтиды. Параллельдік есептеулер теориясы алгоритмдердің қолданбалы теориясының бөлімі.

Параллельді есептеулерді іске асырудың әртүрлі әдістері болады. Мысалы, әрбір есептеу процесі операциялық жүйе процесі ретінде іске асырылуы мүмкін, немесе есептеу процесі ОЖ-ның бір процесс ішінде орындалу ағындар жиыны ретінде болуы мүмкін. Параллельдік программалар тізбекті бір процессорда орындалуы мүмкін, немесе параллельді – әрбір есептеу процесіне бір немесе бірнеше процессорлар беру арқылы іске асырылады [3].

Параллельдік программаларды жобалау кезіндегі негізгі қиындық – әртүрлі есептеу процестері арасындағы байланысты қамтамасыз ету, процестер арасында бөлінетін ресурстар координациясын қамтамасыз ету [5].

Кейбір параллельді программалау жүйелерінде компоненттер арасында мәліметтерді алмастыру программистқа көрсетілмейді, басқаларда ол нақты көрсетілу керек. Нақты өзара қатынастар екі типқа бөлінеді:

- Бөлінетін жады арқылы қатынас: мультипроцессорлық жүйенің әрбір процессорында бір процестің орындалу ағыны іске қосылады. Ағындар берілген процеске жалпы жады бөлігі арқылы мәліметтермен алмасады. Ағындар саны процессорлар санына сәйкес болады. Ағындар тіл құралдарымен құрылады (Java немесе C#, C++ (C++11 бастап), C (C11 бастап)), библиотекалар көмегімен (мысалы, C/C++ -та PThreads көмегімен), декларативті түрде (мысалы, OpenMP библиотекасы көмегімен), немесе компилятордың кірістірілген құралдарымен автоматты түрде (мысалы, High Performance Fortran). Бұл программалау түрі ағындарды өзара координациялау үшін басқаруды қармау формасын талап етеді.

- Хабарламалар жіберу арқылы өзара байланыс: көппроцессорлық жүйенің әрбір процессорында бір ағынды процесс іске қосылады, ол басқа процессорларда жұмыс істеп тұрған мәліметтермен хабарламалар арқылы алмасады. Процестер операциялық жүйенің сәйкес функциясын шақыру арқылы айқын құрылады, ал хабарламалармен алмасу – библиотека арқылы (мысалы, MPI хаттаманы іске асыру), немесе тіл құралдары арқылы (мысалы, High Performance Fortran, Erlang немес occam). Хабарламалармен алмасу асинхронды немесе «рандеву» әдісін (жіберуші оның хабарламасы жеткізілмегенше бұғатталады) қолдану арқылы іске асырылуы мүмкін. Хабарламаларды асинхронды жіберу сенімді (жеткізу кепілдігімен) немесе сенімсіз болуы мүмкін [6].

Хабарламалармен алмасуға негізделген параллельдік жүйелер бөлінетін жадылы жүйелерден қарағанда қарапайым болады.

Хабарламалармен алмасуға негізделген параллельдік жүйелер және бөлінетін жадылы жүйелерде өнімділік сипаттамалары әртүрлі.

- Гибридты әдіс: жүйенің әрбір түйіні жалпы жадылы мультипроцессор болатын үлестірілген жадылы көппроцессорлы жүйелерде программалаудың гибридты әдісін қолдануға болады. Жүйенің әрбір түйінінде осы түйіннің процессорлары арасында ағындарды үлестіретін көп ағынды процесс іске қосылады. Түйіндегі ағындар арасында мәліметтер алмасу жалпы жады арқылы іске асырылады, ал түйіндер арасында мәліметтер алмасу – хабарламалар беру арқылы іске асырылады. Программалаудың гибридті әдісі күрделірек, бірақ көп процессорлы жүйенің әрбір түйін ресурстарын қолдануда тиімді [7].

Көпағанды бағдарламалау мүмкіндіктері:

- көпесептікті қамтамасыз ететін аппараттық құралдар арқылы өнімділікті

арттыру. Әдетте, көпағындықты іске асыратын қосымшалар бар процессорлар санын ескермеу керек. Қосымша өнімділігі қосымша процессорлар болғанда бірқалыпты артады. Параллельдіктің жоғары деңгейлі қосымшалары және сандық алгоритмдер жылдам орындалады;

- қосымшаның өткізу қабілеттілігінің артуы. Бірнеше ағындар санымен оңтайландырылған қосымша тізбекті қосымшадан қарағанда берілген класстың көп есебін орындай алады;

- қосымша жауап беру жылдамдығының артуы. Бір біріне тәуелсіз көп әрекеттері бар кез-келген қосымша әрбір әрекет бөлек ағында орындалатындай етіп қайта жобалануы мүмкін. Мысалы, көпағынды интерфейс қолданушысы басқа әрекет бастау үшін алдыңғы әрекеттің аяқталуын күтуге міндетті емес;

- жүйелік ресурстарды тиімдірек қолдану. Ағындардан қарағанда бірнеше процесс орындау шығынды болады;

- кодтың оптималды құрылымы. Кейбір қосымшалар бір қосымшадан қарағанда бірнеше өзара тәуелсіз бөліктер ретінде тиімді болады. Көпағынды қосымшаларды қолданушы талаптарының өзгеруіне бейімдендіру жеңіл болады [8].

Көпағынды бағдарламалауды қолдану жобалушыға келесі әрекеттерді орындауды талап етеді:

1. жүйенің объекттік декомпозициясы;
2. параллельдеу әдісін таңдау;
3. функционалдық және құрылымдық декомпозиция;
4. ақпараттық өзара қатынасуларды анықтау;
5. есептеу құралын таңдау;
6. программалық қамтаманы құру және оңтайландыру [9].

Көпағынды бағдарламалауда бірнеше нұсқаулар бір уақыт моментінде орындалуы мүмкін. Бағдарлама бірнеше параллель есептерге жіктеледі. Бағдарлама бір уақытта орындалатын бірнеше ішкі бағдарламалардан тұруы мүмкін. Көпағынды бағдарламалауда ішкі бағдарламалар тізбектілігі әрдайым болжана бермейді. Бірнеше есеп кез-келген процессорда орындала басталуы мүмкін, бір есеп бірінші немесе барлық есеп бір уақытта аяқталуына кепіл болмайды.

Компьютерде бірнеше процессор болуына қарамастан бірнеше есеп бір процессорда орындалуы мүмкін. Мұндай жағдай жоспарлаудың жүйелік стратегияларымен жасалады. Есептердің ұзақтығына жоспарлау стратегиялары, процесстер, ағындар басымдылығы, еңгізу- шығару құралдарының жылдамдығына байланысты болады.

Көпағынды бағдарламалауды қолдану жүйелердегі процесстерді іске асыру үшін қажет және де олардың тиімділігін, өнімділігін арттыруға, жоғары сапа көрсеткіштерге жетуге мүмкіндік береді.

Қолданылған әдебиеттер тізімі

1. А.С.Антонов Введение в параллельное программирование, методическое пособие, М, НИВЦ МГУ, 2002-69с
2. В.Э.Малышкин, В.Д.Корнеев Параллельное програмирование мультикомпьютеров, Новосибирск, НГТУ, 2006-296с
3. А.В.Старченко Высокопроизводительные вычисления на кластерах, Томск, 2008-198с
4. Букатов А.А. Программирование многопроцессорных вычислительных систем, ООО ЦВВР, 2003-208с
5. Бабаян Б.А., Бочаров А.В., Волин В.С. и др, Многопроцессорные ЭВМ и методы их проектирования, М.: Высшая школа, 1990
6. Бабенко Л.К., Кравченко П.П., Макаревич О.Б., Чефранов А.Г., Архитектура и математическое обеспечение многопроцессорных супер ЭВМ. Учебное пособие, Таганрог: Таганрогский радиотехнический институт, 1992
7. Барский А.Б, Параллельные процессы в вычислительных системах: планирование и

организация, М.: Радио и связь, 1990

8. Барский А.Б, Параллельные технологии решения оптимизационных задач, Информационные технологии. Приложение, №2, 2001

9. Барский А.Б., Шилов В.В, Вычислительная система, управляемая потоком данных, Информационные технологии, Приложение, №8, 2000

ӘОЖ 004.032.26.681.3

НЕЙРОНДЫҚ ЖҮЙЕЛЕР КӨМЕГІМЕН БЕТ БЕЙНЕНІ ТАҢУ

Есенбаева Гүлшат Бийболатқызы

gulshat-95-95@mail.ru

Л.Н.Гумилев атындағы ЕҰУ «Есептеу техникасы және бағдарламалық қамтамасыз ету»

мамандығының студенті, Астана, Қазақстан

Ғылыми жетекшісі – А.Адамова

Соңғы жылдары кибернетиканың жаңа облысы - нейронды желілер өте жақсы дамып келеді. Нейронды жүйелер бейнелерді тану процестерін автоматизациялау мәселелерін тиімді шешуге мүмкіндік береді [1]. Басты міндеті тиімділігі адам мүмкіндіктерінен кем болмайтын, нейронды желілер көмегімен бағдарламалық қамтаманы жасақтау болып табылады. Нейрондық жүйелердің басқа әдістерден, мысалы, эксперттік жүйелерден басты айырмашылығы: нейрожүйелер негізінен алдын ала белгілі модельдерге мұқтаж емес, ұсынылатын ақпарат негізінде оларды өздері құра алатынында. Сондықтан, нейрондық жүйелер және генетикалық алгоритмдер болжау, жіктеу, тану, басқару міндеттерді шешу керек жерлердің барлығында қолданысқа кеңінен енді [2]. Адамды бет бейнесі бойынша танудың негізгі ерекшеліктері:

1. Арнайы немесе қымбат жабдықтар талап етілмейді. Көптеген қосымшалар үшін дербес компьютер мен қарапайым бейнекамера жеткілікті.

2. Құрылғылармен физикалық байланыс қажет емес. Жүйесінің жұмыс істеуін күту үшін арнайы тоқтауға немесе бір нәрсеге жанасу қажет етілмейді. Көп жағдайда камера алдынан өтуге немесе аздаған уақытқа кідіріс жасаса жеткілікті.

Басты мақсат видео ағындағы бет бейнені адамның қатысуынсыз тануға арналған нейрондық жүйе көмегімен жасақталған бағдарлама құру болып табылады. Видео ағынында бет бейнені автоматты түрде тану көптеген басымдықтар береді. Бұл міндеттерде нейрондық жүйелерді пайдалану адамның қатысуынсыз, видео ағының нақты уақытында адамдардың беттерін табуға және деректер қорына беттерді енгізу үшін ең жақсы жағдайын таңдауға мүмкіндік береді. Әзірленіп жатқан бағдарламалық қамтаманың жазбаларды салыстыру, тану және нақты уақытта сәйкестік туралы сигнал беру мүмкіндіктері болады. Сондай-ақ, бағдарламалық қамтама адамдарды беттері бойынша мәліметтер қорынан іздей алады, жынысын және кадрда адамдардың кездесу жиілігін санай алады.

Бет бейнені тану үшін келесі тапсырмаларды орындау керек [3]:

1. Face Detection (бет бейнені табу). Кадрдан бір немесе бірнеше бет бейнесін табу және кескіндемеде беттің орналасқан жерінде тік бұрышты облысты салу.

2. Face Verification (бет бейнені тексеру). Екі бет бейненің бір адамға тиесілі екендігінің мүмкіндігін тексеру.

3. Face Identification (бет бейнені сәйкестендіру). Бет бейнені тану және сәйкестендіру.

4. Similar Face Searching (ұқсас бет бейнелерді іздеу). Мәліметтер қорында іздеу.

5. Face Grouping (бет бейнелерді топтастыру). Бет бейнелерді визуалды ұқсастық негізінде бірге топтастыру.

Бағдарламада Python бағдарламалау тілі және OpenFace пен dlib кітапханалары қолданылды. Бет бейнені тану үшін терең үйірткілді (свёрточный) нейрондық жүйені оқыту