КРАТКИЕ НАУЧНЫЕ СООБЩЕНИЯ
SHORT SCIENTIFIC REPORT

УДК 004.421, 621.321

ПАРАЛЛЕЛ ҚАЙТА ИШЛАШ ТЕХНОЛОГИЯЛАРИ ВА ЗАМОНАВИЙ ИНСТРУМЕНТАЛ ВОСИТАЛАРИ

Рахимов М.Ф., Намазов А.О.

Ушбу мақолада тезкор ҳисоблашнинг самарали технологиялари, оқимли қайта ишлаш усуллари, параллел тизимларнинг таснифи, бир ядроли процессорларда оқимли қайта ишлаш технологиялари таҳлили кўриб чиқилди. Шунингдек, кўпядроли процессорларда параллел (оқимли) қайта ишлаш жараёнларини янги технология сифатида унумдорликни оширишдаги аҳамияти, кўпядроли процессорларда оқимли қайта ишлаш ва замонавий параллелаштиришнинг инструментал воситалари таҳлил қилинди.

Таянч иборалар: параллел ҳисоблаш, кўпядроли процессорлар, оқимли қайта ишлаш, унумдорлик, жараён, кэш ва оператив хотира, OpenMP пакети, TBB инструментал воситаси, PPL, ICP, MKL, CUDA технологиялари.

В данной работе рассматриваются реализации параллельных алгоритмов для высокопроизводительных вычислительных систем. Даются понятия программного потока, поточной вычислительной модели, рассмотрены технологии многозадачности и гиперпоточности. Исследованы возможности многоядерных процессоров и инструментальные средства поддержки параллельного программирования.

Ключевые слова: параллельная обработка, многоядерные процессоры, кэш память, пакет OpenMP, инструментальные пакеты, ускорение.

In this article researches the implementation of parallel algorithms, modern high-performance computing systems, program flow, Flynn classification, SISD, SIMD, MISD, API calls, flow computational model, multitasking, hyperthreading technology. Also, the problems of stream processing in tasks on multi-core processors are considered as an effective technology to speed up computations. After the emergence of the concept of parallelism at the command level, the constant progress in the development of microprocessors led to the release of processors with multiple cores. To effectively use the capabilities of multi-core processors, it is necessary to understand the features of the flow programming model, as well as the capabilities of the platform hardware. To enable par-

allel processing of programs, the hardware platform must support the ability to simultaneously execute several program threads. With proper implementation, threading can improve performance through more efficient use of hardware resources. There is a direct correspondence between processes and program threads. There may be multiple threads in the process. All program threads of the same process share the same address space and can thus interact with each other. A program has one or more processes, each process contains one or more threads, and each thread is assigned to be executed by the processor by the operating system scheduler.

Keywords: parallel processing, multi-core processors, API (Application Programming Interface), HT Technology (Hyper-Threading Technology), CMP (Chip Multiprocessing), cache memory, OpenMP package, TBB tool package acceleration, PPL, ICP, MKL, CUDA.

I. КИРИШ

Телекоммуникация соҳасига тегишли хизмат кўрсатиш кўламининг кенгайиб бориши, аудио, овоз ва тасвир сигналларини узатиш / қабул қилиш ва қайта ишлаш каби жараёнларнинг ҳажмининг ошиб бориши ҳисоблаш жараёнлар тезлигини янада оширишни талаб қилади. Бунинг ечими кўп ядроли процессорлардан фойдаланиш бўлиб, бу ёндашув дастур даражасида параллел ишлашни талаб қилади. Алгоритмни мукаммал шакллантириш кўп ядроли процессорлар ишлаш унумдорлигини сезиларли даражада ошириш учун имкон беради. Мукаммал шакллантиришнинг шартлари эса сонли усулларни такомиллаштириш, дастурости қисмларга оптимал тақсимлаш, дастурий тиллардан мос равишда фойдаланиш ва турли хил параллелаштириш технологиялардан самарали фойдаланган ҳолда эришиш мумкин. Кўп ядроли процессорли тизимларда маълумотларни параллел қайта ишлашнинг асосий шартлардан бири бу ишлов бериладиган маълумотларни мақбул оқимларга бўлишдан иборатдир [1].

Буйруқларни параллелаштириш концепсиясининг пайдо бўлиши, микропроцессорлар ривожланиш тенденциясига кўпядроли процессорлар яратилишига туртки бўлди. Кўпядроли процессорларнинг имкониятларидан самарали фойдаланиш учун, оқимли дастурлаш модели хусусиятларини ҳамда платформанинг аппарат қисмининг имкониятларини тўлиқроқ тушуниш керак. Дастурларнинг параллел ишлашини таъминлаш учун аппарат платформаси бир вақтнинг ўзида бир нечта дастур оқимларини бажариш қобилиятига эга бўлиши керак. Кўп оқимли ишлов бериш имкониятларидан унумли фойдаланиш аппарат ресурсларидан янада самарали фойдаланиш орқали иш фаолиятини яхшилашимиз мумкин.

Дастурий оқим – бу бошқа буйруқ қаторларидан мустақил равишда бажариладиган ва ўзаро боғланган буйруқлар кетма-кетлигини ўз ичига олади. Ҳар қандай дастурда камида битта дастурий оқим - асосий оқимга эга бўлади, қайсики дастурни ишга туширади ва биринчи буйруқлар бажарилишини бошлайди.

Аппарат даражасида дастурий оқим дийилганда – бу алгоритмни бажариш учун кўрсатилган йўл бўлиб, бунда бошқа аппарат воситаларнинг дастурий оқимларини бажаришига нисбатан мустақил амалга ошрилади. Ҳар бир дастурий оқим бажарилиши учун операцион тизим томонидан тегишли ресурслар ажратилади. Оқимли қайта ишлаш ўзида дастурнинг тезкор ишлашини ва амалга оширилаётган платформанинг имкониятларини акс эттириши лозим.

II. АСОСИЙ ҚИСМ

Компьютер архитектураси ва уларнинг платформалари икки хил ўлчовда таснифланиши мумкин. Биринчи ўлчов – бу бир вақтда бажарилиши мумкин бўлган буйруқлар оқимлари сони. Иккинчи ўлчов – бу бир вақтда бажарилиши мумкин бўлган маълумотлар оқими сони. Ҳар қандай операцион тизимлар буйруқлар ва маълумотлар оқими бўйича тавсифланиши мумкин [2].

Шундай қилиб, замонавий параллел компьютерлар SIMD ёки MIMD тоифасига талуқли бўлиши мумкин. Дастурлар маълумотлар даражасида ва буйруқлар даражасида параллелизмдан фойдаланишлари мумкин [3, 4]. Бугунги кунда барча асосий дастурлаш тиллари - шартли дастурлаш тиллари (C, Fortran, Pascal, Ada) бўладими, объектга йўналтирилган (C++, Java, C#) дастурлаш тиллари бўладими, функционал (Lisp) ёки мантиқий (Prolog) дастурлаш тиллари бўладими фарқи йўқ, барчаси оқимларни қўллашни қўллаб-қуватлайди.

Дастурий таъминотни ривожланиб бориши иловаларнинг бир вақтнинг ўзида бир неча вазифаларни тенг бажара олиш имкониятини оширди. Сервер дастурий таъминитлари асосан бир ядроли процессорларда бажарилиладиган кўп дастурий оқимлардан ёки жараёнларда ташкиллаштирилган эди. Дастурий таъминот даражасидаги бундай параллелаштириш имкониятларини қўллаб қуватлаш учун дастурий ҳамда аппарат таъминот каби ёндошувлар қабул қилинган.

Ёндошувлардан бири – бу кўп вазифали операцион тизимларда фойдаланиладиган вақтни тақсимлаш режимидир. Вақтни тақсимлаш режимида кўп вазифаликни ташкил қилиш ишлаб чиқарувчиларга бир неча оқимларнинг киритиш – чиқариш амалларидаги кутиш вақтларини яшириш эвазига бажариш имконини беради. Бироқ, бундай маълумотларни қайта ишлаш модели ҳақиқий параллел қайта ишлаш

имконини бермайди, фақатгина ҳар бир сонияда бит оқим қайта ишланиши мумкин.

Функционал жиҳатдан ишлаб чиқарилган бир ядроли процессорларда жорий этилган ушбу ёндашувнинг сўнгги намуналаридан бири – бу мантиқий (виртуал) процессорни тегишли архитектурада маълумотлар эгаллаган майдонларни такрорий ҳосил қилиш, яъни нусхалаш йўли билан яратишдир. Шу билан бирга, ижро этувчи компьютер ресурслари (регистрлар, кеш, шиналар, ижро этувчи ва бошқаратловчи блоклар) операцион тизим томонидан асосий ва мантиқий процессорлар ўртасида тақсимланади. Бу каби бир вақтнинг ўзида кўпоқимли қайта ишлаш усули Intel фирмасининг машҳур HT Technology (Hyper-Threading Technology) номини олган. HT технологиясидан келиб чиқан ҳолда, битта процессор (дастурий нуқтаи назаридан) икки ва ундан ортиқ бўлиб кўриниши мумкин. Бу албатта кўппроцессорли тизимлардаги каби иловалар ва операцион тизимга турли мантиқий процессорларда режалаштириш эвазига бир вақтнинг ўзида бир неча дастурий оқимларни бажарилиш имконини беради. Архитектура нуқтаи назаридан, физик ва мантиқий процессорларга доимий буйруқлар бажарилиш учун келиб туради ва турли ижро этувчи ресурслар томонидан бир вақтнинг ўзида алоҳида бажарилади.

HT технологияси унумдорликни оширади, бироқ кутиш ҳолатларини яшириш орқали. Бир ядрода бир неча дастурий оқимлар биргаликда ижро этилади, аммо бажарилиш жараёни реал ҳолатда параллел бўлмайди. Натижада унумдорлик даражаси илованинг мураккаблигига ва аппарат платформанинг имкониятларига боғлиқ бўлиб қолади. Алоҳида яхши тайёрланган иловалар 25-30%-гача унумдорликни ошишига эришиши мумкин. Бошқача қилиб айтганда, процессор бир оқимли қайта ишлашга нисбатан 1.3 маротаба кўп буйруқларни бажариши мумкин.

Бир ядроли процессорлар фақатгина буюруқлар оқимини саралаши мумкин, лекин бир вақтда бажара олмайди. Шунинг учун бундай архитектурада кўп оқимли иловаларнинг унумдорлигининг ошиши чегараланган.

Кўпядроли процессорлар эса икки ва ундан ортиқ ижро этувчи ядролар билан таминланган ва шу тариқа ҳақиқий аппарат даражасидаги кўпоқимлиликни таъминлаб бера олади.

Гипроқимлилик – бу дастурчига процессорнинг бўш ресурсларида қўшимча иш қилишга рухсат берувчи воситадир. Илгарилари кўп оқимли қайта ишлашни амалга ошириш учун бир нечта ҳисоблаш машиналар (мисол учун кўппроцессорли тизимлар ёки суперкомпьютерлар)ни талаб қилар эди. Ҳозирги пайтда гипроқимлиликни қўллаб қуватловчи функция жихатдан кучли процессорлар битта процессорнинг ўзида кўпоқимлилик қайта ишлашни таъминлаб бера олади. Бу турдаги процессорлар

фақатгина бир ижро этувчи воситага эга бўлади, бироқ у конвейрни ва бошқа аппарат восита ресурсларни бажарилаётган оқимлар ўртасида тақсимлаб беради. Бундай қайта ишлаш параллел қайта ишлаш эмас, балки "рақобат"ли қайта ишлаш ҳисобланади ва CMT (Chip Multiprocessing) технологияси деб қайт этилади.

MMX (Multimedia Extensions – мультимедияли кенгайтма) – бу Intel фирмаси томонидан яратилган, SIMD архитектурасига асосланган технология бўлиб, процессордаги аудио ва видио маълумотлар оқимларини кодлаш/декодлаш жараёнларини тезкорлигини оширувчи буйруқлар мажмуасидир. Бу технология процессорнинг тор иқтисослашган буйруқлар тизимининг кенгайтмаси бўлиб, айнан MMX процессордан фойдаланиш орқали юқори даражадаги самарадорлик эришиб бўлмайди. MMX технологияси билан таниш бўлмаган, умумий хусусиятга эга бўлган иловаларни бажаришда унумдорлик миқдори бир неча фоизларгагина ошиши мумкин. "Ҳақиқий" MMX кодини амалга оширганда тезкорлик 5-6 баробаргача ошиши мумкин, бу албатта маҳаллий қисмлар учун талуқлидир. Шунинг учун MMX технологияси асосан график маълумотлар устида параллел жараёнларни амалга оширишда юқори унумдорликка эришади.

Кўпядроли процессорларда оқимли қайта ишлаш.

Ҳисоблаш машинанинг тан нархини сезиларли даражада оширмаган ҳолда, дастурий оқим даражасида параллелаштиришни жорий этиш ечими бўлиб – кўпядроли процессорларни ишлаб чиқиш ҳисобланди.

Кўпядроли процессорлар бир кристаллнинг ўзида мултипроцессорли қайта ишлаш имконига эга. Бу CMP технологияси деб аталади. Аппарат воситасини лойиҳалаштирувчилар процессор архитектурасида икки ва ундан ортиқ ижрочи ядроларни жойлаштиришни қўллашади. Бу икки ядро мисоли икта процессорнинг бир техник кристаллда жойлаштирилганидир. Ижрочи ядроларнинг ҳар бири ўзнинг архитектуравий ва аппарат ресурсларига эга. Лойиҳалаштирилишига қараб бу процессорлар шахсий кэш хотираларини бўлишган ҳолда бир кристалл ичида фойдаланишлари мумкин. Бундан ташқари ҳар бир ижрочи ядро кўпоқимли SMT (Simultaneous multithreading) технологиясидан фойдаланган ҳолда қўшимча мантиқий ядроларни жорий этиши ва унумдорликни ошириши мумкин.

Натижада, кўпядроли архитектуранинг қўлланилиши кетма – кет буйруқ (дастурий оқим)лар аппарат қисмларни мустақил равишда ишлатиш ҳуқуқига эга бўлишади. Бу эса, ҳар бир ишнинг ҳақиқий параллел равишда амалга оширилишига имкон беради.

Юқорида келтириб ўтилган унумдорликни оширишга бўлган чекловлар кўпядроли архитектураларда диярли йўқ, кўпядроли платформаларда ядролар ресурслар бўшашини кутишмайди ва оқимлар

ҳар хил ядроларда мустақил ижро этилади. Кўпядроли платформалар инжинерларни турли процессор ядролари орасидаги иш юкламаларини бир-бирлари билан оқилона алмашиш орқали иловаларни оптималлаштиришга имкон беради.

Агар процессор икки ёки ундан ортиқ ядрога эга бўлса, у CMP технологиясида ишловчи, бир кристал чип устида кўп ишлов бериш амалларини бажара олади.

CMP тушунчаси телекомуникация ва сигналларни қайта ишлаш соҳасида қўлланиладиган махсус процессорларда кенг учраб туради. CMP технологиясида бир кристаллнинг чипида бир неча тўлиқ процессорлар жойлашган бўлади. Саноат соҳасидаги кенг қамровли масалаларда CMP технологияси қўллаш мақсадида ундан фарқ қилувчи кўпядроли процессорлар жорий қилинди. CMP тизимларида бир неча процессорлар бир кристалл чипида жойлаштирилган бўлса, кўпядроли процессорларда бир неча ядролар бир чип ичида жойлашган бўлади. Ядро ва процессор ўртасида каттагина фарқ мавжуд.

Маълумки, процессор буйруғи дастурий таъминотни амалга оширишнинг асосий қисми ҳисобланади. Буйруқ икта асосий қисмдан иборат: операцион қисми (операция коди, манзиллаш тизими) ва манзил қисми (қайта ишланган операндларнинг хотирасида сақлаш манзили). Айнан шу икки компонента – операцион тугун ва хотира асосий ўринни эгаллайди. Ушбу икта асосий ишлов бериш тугунларининг мавжудлиги ва аппарат архитектураси бошқа элементлари билан алоқа интерфейси мавжудлиги - ядронинг функционал жиҳатдан маълумотларни қайта ишлаш бирлигини аниқлайди.

Процессор – бу кенгроқ тушунчадир, у бир нечта ядроларни ўз ичига олиши мумкин. У ядродан катта миқдордаги маълумотларни қайта ишлаш усули ва ички хотира турлари (кэш хотираси, RAM, ROM) билан кенгроқ фарқ қилади, шунингдек, кенг имкониятли бошқарувчи модуллари ва кенгайтирувчиларга эга эканлиги билан ҳам ажралиб туради. Шунинг учун, ишлов бериш тугунларидаги бундай иерархияга кўп ядроли процессор ёки кўплаб ядроларни бирлаштирувчи компьютер чипи ҳам деб аталади.

Агар процессорнинг ҳар бир ядроси ўзининг оқимларни мустақил қайта ишлай олиш даражасига эга эканлигини ҳисобга олсак, унда кўпядроли архитектура ҳисоблаш жараёнларни ташкил этишнинг (аппарат ечимларининг нисбатан соддалиги билан) энг самарали усулларидан бири деб ҳисоблаш мумкин.

Параллел ҳисоблаш тизимларини ташкил қилишнинг турли хил усуллари мавжуд. Бу борада вектор-конвейрли компьютерлар, массивли-параллел ва матрицали тизимлар, кенг буйруқли компьютерлар, махсус процессорлар, кластерлар, кўп тармоқли архитектурали компютерларни

айтиб ўтиш мумкин. Худди шу рўйхатга, мисол сифатида систолик массивлар ёки датафлов компютерлар каби архитектураларни киритиш мумкин. Ҳар бир архитектурани айнан қайси асосий омиллар ифодалашини аниқлаш ҳисоблаш тизимларини классификациялаш заруратини келтириб чиқаради. Юқорида келтирилган Флинн классификациясини таҳлил қилиш натижасида шундай хулосага келиндики, кўпядроли процессорларни таққослаш учун икта архитектурани танлаш лозим: SIMD ва MIMD. Ушбу иккита архитектура бир вақтнинг ўзида ишловчи ҳисоблаш тугунларини (процессорлар, компьютерлар) ишлатган ҳолда параллел оқимли қайта ишлаш имкониятидан максимал фойдаланади. Ушбу икки архитектуранинг замонавий микропроцессор технологиялари ютуқлари билан биргаликда ривожланиши кўпядроли процессорларни яратишга олиб келди. Параллелаштиришнинг замонавий инструментал воситалари.

Параллел қайта ишлашда аппарат воситалардан ташқари махсус кутубхоналари ва параллелаштиришнинг замонавий инструментал воситалардан ҳам унумли фойдаланиш орқали самарадорликка ижобий тасир кўрсатиш мумкин. Кўпядроли процессорларнинг аппарат таъминоти имкониятларини ошириш йўли билан самарадорликни ҳар доим ошириш имконияти йўқлигини кўпгина инжинерлар ўзларининг илмий тадқиқот ишларида амин бўлишди. Бу эса кўпядроли процессорларни самарадорлигини оширишда бошқа томондан ёндошиш кераклигини кўрсатди.

Шундан келиб чиқиб, иловалардаги мустақил қайта ишлаш имкони мавжуд бўлган дастурий оқимларни кўпядроли процессорларда параллел қайта ишлаш имконини берадиган махсус пакетлар: PPL (Parallel Patterns Library), ICP (Intel Cilk Plus), OpenMP (Open Multi-Processing), TBB (Intel Threading Building Blocks), MKL (Math Kernel Library) ва CUDA (Compute Unified Device Architecture) кабилар машҳур фирмалар томонидан тақдим этила бошланди. Қуйида буларга қисқа қилиб изоҳ берамиз.

PPL – бу параллеллаштириш кутубхонаси бир вақтда дастурларни ишлаб чиқиш учун кенг кўламли ва қулай фойдаланишни таъминлайдиган дастурий моделини тақдим этади. PPL кутубхонасининг Concurrency Runtime (оқимларнинг биргаликда қайта ишлаш даври) нинг режалаштириш ва ресурсларни бошқариш қисмларига асосланади. Бу параллел равишда маълумотлар устида параллел ишлайдиган алгоритм ва махсус контейнерларни тақдим этиш орқали дастур бажалишини параллел бажарилишини таъминлаб берадиган C++ дастурлаш тили тадқим қилган кутубхона ҳисобланади.

PPL қуйидаги хусусиятларни тақдим этади:

- вазифа параллелизми: бир нечта иш элементини (вазифаларини) параллел равишда бажаришни Windows ThreadPool (Windows операцион

тизимининг оқимлар билан ишлаш технологияси) устида ишлатадиган механизмни яратиш;

- параллел алгоритмлар: параллел алгоринтмларни генерация қилиш, параллел равишда ишлаш учун ўзаро бир вақтда ишлайдиган умумий алгоритмларни яратиш;
- параллел контейнерлар ва объектлар: махсус контейнерлар ва махсус объектлар устида параллел алгоритмлар асосида ишлов бериш кабилардир.

PPL бир вақтнинг ўзида маълумотларни тўплаш бўйича ишларни амалга оширадиган алгоритмларни тақдим этади. Ушбу алгоритмлар C ++ стандарт кутубхонаси томонидан тақдим этилганларга ўхшайди.

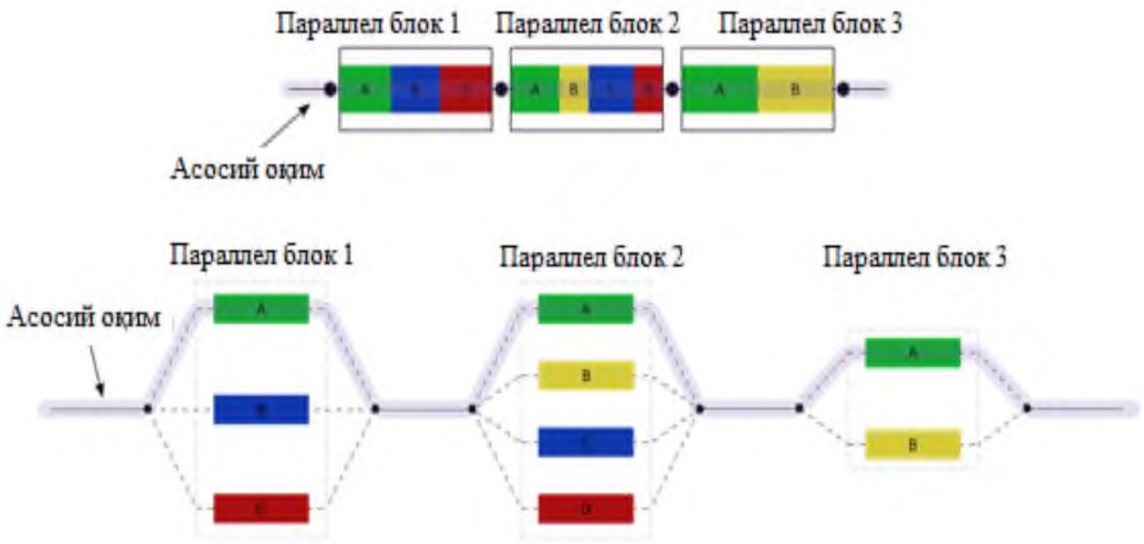
Параллел алгоритмлар Concurrency Runtime да мавжуд функциялардан иборат. Буларга мисол қилиб parallel_for ва parallel_for_each ларни келтириш мумкин.

ICP – бу C ва C++ дастурлаш тилининг кенгайтмаси бўлиб, умумий хотирали тизимларда параллелаштиришни вазифалар (task) ва маълумотлар (data) бўйича эффектив ва хавфсиз, ҳамда векторли параллелизмни таъминлашни дастурчи учун енгиллаштиради, дастурнинг параллелаштирилиши лозим бўлган фрагментлари қисм вазифаларга ажралган ҳолда («ота»-«фарзанд») муносабатда боғланиш тартибини белгилайди. Бундай параллелизмни ташкил этиш, одатда «fork-join» деб аталади.

Intel Cilk Plus дастурнинг кетма-кет семантикасини сақлаган ҳолда ҳам кетма-кет ҳам параллел режимларда бажарилади. Параллел бажарилиш жараёни, қачонки тизим платформасида етарлича ядролар сонига қараб амалга оширилаверади.

OpenMP - бу API, умумий хотирали кўпядроли тизимлар учун мўлжалланган технология ҳисобланади. OpenMP спецификациясини ишлаб чиқиш бир неча йирик компютер ва дастурий таъминот ишлаб чиқарувчилари томонидан амалга оширилди. OpenMP асосий компиляторлар томонидан қўллаб-қувватланади.

OpenMP пакетида коддаги оқимлар кўринмайди. Бунинг ўрнига, компиляторга код блокларини параллел ҳолга келтириши мумкин бўлган #pragma кўрсатмалари билан буйруқ бериш мумкин. Ушбу маълумотни билиш учун компилятор параллел блок коди учун кўплаб бошқа оқимларни яратадиган бир асосий оқимдан иборат бўлган дастурни ишлаб чиқиши мумкин. Бу оқимлар параллел блоклар охирида синхронлаштирилади ва биз яна битта асосий оқимга бирлаштириб қайтамиз (1 - расм).



1 – расм. OpenMP стандартини оқимлар билан ишлаш чизмаси

OpenMP директивлари #pragma omp билан бошланади. Parallel - ушбу кўрсатма оқимлардан N гуруҳини яратади. N қиймати асосан амаллар бажарилиш вақтида процессор ядролари сони билан белгиланади, аммо N қийматини дастурчи томонидан ҳам ўрнатилиши мумкин. Амалга оширилгандан кейин, оқимлар асосий оқимга "бирлаштирилади" [7].

TBB кутубхонаси параллел оқимли дастурлашда C++ дастурлаш тили томонидан фойдаланувчиларга тақдим этилган қўшимча имконият ҳисобланади. TBB пакети C++ нинг махсус шаблонларини амалга оширишга ва кўп оқимли дастурлашнинг юқори бўлмаган қисимларини параллеллаштириш мақсадида таклиф қилинган. TBB турли операцион тизимлар ва турли архитектуралар билан ҳам ҳисоблашларни амалга ошира олади [8].

TBB кутубхонаси параллел дастурлашнинг умумий муаммоларни ҳал этиш учун турли функция ва синфлардан иборат:

- parallel_for функцияси – чегараси аниқланган параллел цикллар учун;
- parallel_reduce функцияси – параллел цикллар учун (комбинация қилиш йўл билан);
- parallel_while синфи – итерациялар сони номаълум бўлган параллел циклларни яратиш;
- pipeline синфи - конвейр ҳисобларни амалга ошириш учун кенг фойдаланилади.

Intel Parallel Studio таркибига кирувчи яна бир кутубхоналардан бири бу MKL ҳисобланади. MKL – кутубхона ўз ичига математик операцияларни ва алгоритмларни оптималлаштирилган ҳолда амалга ошириш имконини яратади.

MKL кутубхонаси Intel процессорлари учун ҳисоблашларни оптималлаштириш ва Intel, бошқа (gcc, Microsoft) компляторлари кабилар сифатида ишлатиш имконини яратади. MKL математик ҳисоблаш ишларини оптималлашган ҳолда амалга оширади мисол учун векторлар ва матрицалар устида бажариладиган барча амалларни компьютернинг процессос ядроларига тақсимлаган ҳолда параллел ҳисоблайди ва эффектлилиги бошқа праллеллаштириш муҳитларидан бутунлай фарқ қилади

CUDA технологияси – бу ҳозирда замонавий юқори унумдорли ҳисоблаш тизимлари гетероген (гибридли) тизимлар ҳисобланади. Бундай тизимлар икта асосий типга эга компоненталардан ташкил топган: кўп ядроли марказий процессор ва массивли параллел тезлатгич ҳисобланган график процессорлардир. Бу компоненталарнинг биргаликда ишлаши учун махсус технологиялар зарур ҳисобланади.

III. ХУЛОСА

Юқорида келтирилган параллел қайта ишлаш технологиялари ва замонавий инструментал воситалари таҳлили натижасида қуйидаги хулосалар чиқарилди.

1. Параллел қайта ишлаш, ҳақиқатан ҳам бир нечта дастур оқимларини бир вақтнинг ўзида бажарилишини англатади. Кўпгина замонавий кўп ядроли компьютерлар – бу бир нечта буйруқлар оқимлари ва бир нечта маълумотлар оқимларга (MIMD) эга бўлган процессорлар ҳисобланади.
2. Иловаларда оқимларни қайта ишлаш API ёки кўп тармоқли кутубхоналар ёрдамида амалга оширилиши мумкин. Маълумотларни қайта ишлаш самарадорлигини оширишда аппарат воситаларини HT ва MMX технологиялари билан таъминлаш ижобий натижалар беради.
3. Кўпядроли процессорларнинг аппарат таъминоти имкониятларини ошириш йўли билан юқори самарадорликга ҳар доим ҳам эришиб бўлмайди. Шунинг учун маълумотларни параллел қайта ишлашда махсус кутубхоналар ва параллелаштиришнинг замонавий инструментал воситалардан ҳам унумли фойдаланиш орқали самарадорликка ижобий тасир кўрсатиш мумкин.

АДАБИЁТЛАР

[1] M.M.Musaev, U.A.Berdanov. "The Technology of Parallel Processing on Multicore Processors," International Journal of Signal Processing Systems, Vol. 4, No. 3, pp. 252-257, June 2016. doi: 10.18178/ijsps.4.3.252-257.
[2] Флинн М. Сверхбыстродействующие вычислительные системы. Пер.с англ. – Труды ТИИЭР, 1966, т.54, вып.12, с.311-320.
[3] Головкин Б.А. Параллельные вычислительные системы – М, Наука, 1980, 520с.
[4] http://parallel.ru/computers/taxonomy/flynn.html (Лаборатория параллельных информационных технологий НИВЦ МГУ).
[5] Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования. Пер с англ. – М.: «Вильямс», 2003 – 512с.
[6] Хокни Р., Джессхоуп К. Параллельные ЭВМ. Архитектура программирование и алгоритмы - М, Радиj и связь, 1986,- 420с.
[7] Guralnik, David B. Neufeldt. Webster's New World; 3rd edition. 1988, the University of Michigan.
[8] Reinders, James. Intel Threading Building Blocks: Outfitting C++ for Multi-core Processor Parallelism (Paperback) Sebastopol: O'Reilly Media, 2007. ISBN 978-0-596-51480-8.