

Paralel Proqramlaşdırma Texnologiyalarının Tətbiqi Məsələləri

Rəşid Ələkbərov¹, Tural Mustafayev²

^{1,2}İnformasiya Texnologiyaları İnstitutu, Bakı, Azərbaycan

¹rashid@iit.ab.az, ²tural.mustafayev@iit.ab.az

Xülasə— Məqalədə elmin müxtəlif sahələrində meydana çıxan böyük hesablama və yaddaş resursları tələb edən mürəkkəb məsələlərin həllində paralel proqramlaşdırma texnologiyasından istifadə məsələləri analiz olunmuşdur. Paralel proqramlaşdırma zamanı istifadə olunan alqoritmlər və platformalar haqqında geniş məlumat verilmişdir.

Açar sözlər— *alqoritm, paralel proqramlaşdırma, MPI, OpenMPI, Amdal, Qustafson, PVM, hesablama klasterləri*

I. GİRİŞ

Məlum olduğu kimi mürəkkəb məsələlərin həllində, informasiya bazalarının yaradılmasında, paralel proqramlaşdırma texnologiyalarının rolu əvəz edilməzdir. Qeyd etmək lazımdır ki, elmin müxtəlif sahələrində meydana çıxan böyük hesablama və yaddaş resursları tələb edən mürəkkəb məsələlərin həllində fərdi kompüterlərin hesablama gücü kifayət etmir. Böyük və sürətli hesablama zamanı imkanları daha böyük olan çoxprosessorlu hesablama maşınlarının istifadəsi daha məqsəduyğundur. Hal-hazırda dünyada böyük və sürətli hesablama üçün hesablama klasterlərindən istifadə edirlər. Klaster çoxsaylı hesablama maşınlarını özündə birləşdirərək, onların resurslarını bir məsələnin həllinə yönəltməyə imkan verir. Bu zaman məsələnin bir neçə hissəyə bölünüb hesablanması və alınan nəticənin birləşdirilməsi tələb olunur. Bütün bu əməliyyatlar paralel proqramlaşdırma texnologiyasının tətbiqi ilə mümkün olur. Big Data analizi zamanı böyük həcmdə yarı-strukturlaşdırılmış və struktursuz məlumatların (vəb server loqları, müştəri elektron məktubları və sorğu cavabları, internet istifadəçi loqları, sosial şəbəkə kontentləri, mobil telefon danışıqları, əşyaların interneti zamanı istifadə olunan sensorlardan gələn məlumatlar və s.) emalı tələb olunur. Belə məlumatların tez bir zamanda analiz və emal olunması üçün paralel proqramlaşdırmanın tətbiqi vacibdir.

Paralelləşdirmə artıq uzun illərdir ki, çoxprosessorlu hesablama sistemlərində geniş istifadə olunur. Lakin son illər paralel hesablama sistemlərinə maraq daha da artmışdır. Bunun səbəbi prosessorun işləmə tezliyinin artırılması imkanlarının fiziki olaraq məhdud olmasıdır. Prosessorun işləmə tezliyi yüksək olduqda məsələnin həllinə sərf olunan zaman azalır. Ona görə də məsələnin həllinə sərf olunan zamanı azaltmaq üçün paralel proqramlaşdırmadan istifadə olunur. Son illər kompüterlərin enerji istehlakının azaldılması tez-tez gündəmə gəlir. Məhz bu səbəbdən də çoxnüvəli prosessor arxitekturalarında paralel hesablama texnologiyalarının tətbiqi çox vacibdir [1].

Paralel hesablama sistemlərini paralelləşmənin aparıldığı səviyyəyə görə klassifikasiya etmək mümkündür. Paralelləşmənin çoxnüvəli və ya çoxprosessorlu kompüterlərdə aparılması mümkündür. Həmçinin çoxsayda kompüterlərin birləşdirildiyi qrid və klaster arxitekturalı hesablama sistemlərində paralel hesablama texnologiyalarından geniş istifadə olunur. Bəzi xüsusi məsələlərin həllinin sürətləndirilməsi üçün bəzən ənənəvi paralel arxitekturalarla yanaşı, spesifik paralel arxitekturalar da istifadə olunur [2].

Hər hansı bir məsələnin həlli üçün yazılmış proqramın paralelləşdirməsi zamanı həll vaxtının sürətlənməsi Amdal qanununu ilətin olunur. Bu qanuna görə məsələ bir neçə hissəyə bölünüb paralelləşdirilsə də, nəticəni almağa sərf olunan zaman, paralelləşdirilmiş məsələnin ən uzun fraqmentindən qısa ola bilməz. Qustafson-Barsis qanununa görə isə, məsələni maksimal dərəcədə paralelləşdirərək və istənilən sayda prosessor istifadə edərək nəticəni daha tez almaq olar [3].

II. PARALEL HESABLAMA SİSTEMLƏRİ VƏ ARXİTEKTURASI

Paralel hesablama sistemlərinin təsnifatı ilk dəfə amerikalı alim Q.Flinntərəfindən 1964-cü ildə təklif edilmişdir. Onun təsnifatına görə, hesablama sistemlərinin arxitekturası əmrlər axını və verilənlər axınının qarşılıqlı əlaqələri əsasında yaradılır. Hesablama maşınlarının klassik arxitekturasında əmrlər və verilənlər ardıcıl prosessorayonəldilərək, ardıcılıqla yerinə yetirilir. Lakin hesablama sistemlərində işlənmə prinsipi fərqlidir. Q.Flinnin təklif etdiyi təsnifat aşağıdakı kimidir [4]:

- bir əmr axını və bir verilənlər axını – BƏBV (SISD, Single Instruction Single Data);
- bir əmr axını və çox verilənlər axını – BƏÇV (SIMD, Single Instruction Multiple Data);
- çox əmr axını və bir verilənlər axını – ÇƏBV (MISD, Multiple Instruction Single Data);
- çox əmr axını və çox verilənlər axını – ÇƏÇV (MIMD, Multiple Instruction Multiple Data).

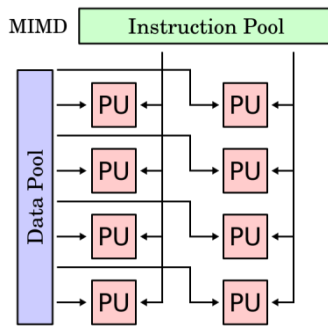
BƏBV arxitektura klassik hesablama maşınlarının (Fon-Neymanın irəli sürdüyü) arxitekturası uyğun gəlir. Belə sistemlər bir mərkəzi prosessordan ibarətdir. Prosessor verilənlər üzərində əməliyyatları ardıcıl olaraq yerinə yetirir. Belə hesablama sistemlərində hər hansı məsələnin həlli üçün yazılmış proqramdakı əmrlər və verilənlər ardıcıl olaraq

yaddaşdan tək-tək çağırılır, yerinə yetirilir və nəticələr yaddaşa yazılır.

İkinci tip hesablama sistemlərində bir əmr ilə çoxsaylı verilənlər üzərində əməliyyatlar aparılır. Bu cür sistemlərdə minlərlə prosessor olur. Bir əmrin köməyi ilə müxtəlif verilənlər üzərində eyni əməliyyatlar aparılır. Məsələn, vektor və matrislər üzərində aparılan əməliyyatları göstərmək olar. Bir əmr vasitəsilə vektorun və ya matrisin bütün elementləri üzərində eyni əməliyyatı həyata keçirmək olar.

Üçüncü tip hesablama sistemlərində bir verilənlər üzərində çoxsaylı əməllər axını yerinə yetirilir. Hələlik bu prinsip əsasında işləyən superkompüterlərdən praktikada istifadə edilmir.

Paralel hesablama sistemlərində ÇƏÇV (Şəkil 1) prinsipi ilə işləyən arxitekturalardan daha geniş istifadə olunur. Belə sistemlər müxtəlif verilənlər üzərində bir neçə əməliyyatın paralel aparılmasını təmin edir. ÇƏÇV prinsipi ilə işləyən çoxsaylı arxitekturalara malik hesablama sistemləri – superkompüterlər yaradılmışdır.



Şəkil 1. ÇƏÇV prinsipi arxitektura

Belə sistemlərdə məlumatlar bir ortaq mənbədən (Data Pool) hesablama qovşaqlarına (PU) ötürülür. Məlumatın emalı təlimatlar (instruction pool) vasitəsilə tənzimlənir. Superkompüterlərin yaradılmasında ÇƏÇV prinsipinə əsaslanan üç tip arxitekturalardan geniş istifadə olunur [4]:

- SMP;
- MPP;
- Klaster.

SMP (symmetric multiprocessing) – simmetrik çoxprosessorlu arxitekturalara malik hesablama sistemlərində prosessorlar arasında əlaqəni təmin etmək üçün ümumi fiziki yaddaşdan istifadə olunur. Prosessorlar ümumi yaddaş vasitəsi ilə bir-biri ilə qarşılıqlı əlaqədə olurlar. Prosessorların hər biri yaddaşın istənilən ünvanına müraciətdə edə bilər. Ona görə də SMP arxitekturalı sistemlər simmetrik sistemlər adlanır. SMP – sistemlər yüksək sürətli sistem şini (SGI PowerPath, Sun Gigaplane, Dec Turbolaset və s.) əsasında yaradılır. Bu sistemin aşağıdakı üstünlükləri var:

- ümumi yaddaşdan istifadə prosessorlararası məlumat mübadiləsinin sürətli aparılmasına imkan verir;

- proqramlaşdırma üçün sadə və universaldır;
- istifadəçilər yaddaşdan istənilən həcmdə istifadə etmək imkanına malikdirlər;
- məsələlərin effektiv avtomatik paralelləşdirilməsi üçün çoxsaylı vasitələr mövcuddur;
- bu tip sistemlərdə otaq şəraitində işləyən adi kondensiyonerdən istifadə edilir;
- qiyməti çox baha deyil.

SMP sisteminin çatışmayan cəhəti ümumi şindən istifadə olunmasıdır. Bu şində hər bir zaman bir anında iki qurğu arasında məlumat mübadiləsi ola bilər. Ona görə də yaddaşın eyni hissəsinə bir neçə prosessorun müraciət etməsi münaqişə halının yaranmasına səbəb olur və hesablama qovşaqları bir-birinin işinə mane olur. Bu cür münaqişələrin yaranması ümumi şinin sürətindən və sistemdə iştirak edən prosessorların sayından çox asılıdır. SMP arxitekturalı sistemlərdə prosessorların sayı 32-yə qədər olur. Bu cür sistemlərdən işçi stansiyaların və serverlərin yaradılmasında geniş istifadə olunur [5].

MPP (Massively Parallel Processors) – kütləvi paralelləşdirilmiş prosessorlu arxitekturasına malik hesablama sistemləri SMP arxitekturalına nisbətən daha geniş yayılıb. Bu tip sistemlərdə yaddaş prosessorlar arasında paylanır. MPP sistemləri kommutatorların köməyi ilə prosessor və yaddaş malik olan ayrı-ayrı qovşaqların birləşməsindən yaradılır. Belə sistemlərdə hər bir qovşaq tam funksiyalı kompüter kimidir. Hər bir yaddaş moduluna yalnız ona qoşulan prosessor müraciət edə bilər.

Əməliyyat sistemi bu arxitekturalda iki variantda işləyə bilər. 1-ci variantda bu modulardan biri idarəedici kompüter kimi seçilir və əməliyyat sistemi tam olaraq ona yüklənir, digər modulalara isə əməliyyat sisteminin sadələşdirilmiş versiyası yazılır.

2-ci variantda isə hər bir modula əməliyyat sistemi tam formada yazılır. MPP arxitekturalı hesablama sistemi yaxşı miqyaslaşdırıla bilər. Bu cür sistemlərdə prosessorların işinin sinxronlaşdırılması asandır. Bu arxitekturala əsasən yaradılan superkompüterlər yüksək hesablama məhsuldarlığına malik olur. Bu sistemlər on minlərlə mikroprosessoru birləşdirir.

Bu sistemlərin bir neçə çatışmayan cəhəti vardır:

- ümumi yaddaşdan istifadə olunmadığından prosessorlararası məlumat mübadiləsinin sürəti aşağı olur;
- hər bir prosessor özünə məxsus olan həcmdə kiçik lokal yaddaşdan istifadə edir;
- prosessorlararası məlumat mübadiləsinə təmin edən xüsusi proqramlardan istifadə olunur.

MPP arxitekturalı superkompüterlərdə mürəkkəb məsələlərin altməsələlərə bölünüb, hesablama qovşaqları arasında paylanması üçün MPI, PVM, BSPlib, Corba və s. kimi tətbiqi proqram paketlərindən istifadə olunur. Bəzi superkompüterlərin yaradılmasında hibrid arxitekturalı olan NUMA-dan (nonuniform memory access) istifadə olunur. Bu

arxitekturdə SMP və MPP arxitekturları birləşdirilib. NUMA-da hesablama sistemi lokal yaddaşa malik olan SMP modullarını yüksək sürətli kommunikasiya şəbəkəsinin köməyi ilə ümumi fiziki yaddaş vasitəsilə birləşdirməklə yaradılır [6].

Klaster-şəbəkə texnologiyasının (ümumi şin kommutatorlarının vasitəsilə və s.) köməyi ilə bir çox kompüter (hesablama qovşağını) özündə birləşdirən hesablama sistemidir. Hesablama qovşağı kimi server, işçi stansiya, fərdi kompüter və ya bleyd-serverlərdən istifadə olunur. Klaster sisteminin digər sistemlərə nəzərən üstünlüyü ondadır ki, hesablama qovşaqlarından hər hansı biri sıradan çıxarsa, onun işini digər hesablama qovşaqları öz üzərinə götürür. Klaster sistemi ayrı-ayrı modullardan təşkil olunur. Hər bir modul özündə prosessoru, əməli yaddaşı və disk yaddaş qurğusunu birləşdirir. Bu arxitektura əsasında yığılmış superkompüterlərin qiymətləri çox ucuz olur. Bu superkompüterlər satışda olan mövcud standart komplektləşdirici elementlər: prosessorlar, kommutatorlar, əməli yaddaş, disk yaddaş və xarici qurğular üzərində yığılır.

Göstərilən modulların bir-biri ilə əlaqələndirilməsi üçün şəbəkə texnologiyasından (Fast/Gigabit Ethernet, Infiniband Myrinet və s.) istifadə olunur. Klaster tipli superkompüterin yığılması, sazlanması, idarə edilməsi çox sadədir. Klaster sistemlərinin qiyməti ucuz, quraşdırılması asan və idarə edilməsi sadə olur. Ona görə də dünyada istehsal olunan superkompüterlərin yaradılmasında bu arxitekturdən geniş istifadə edilir. Superkompüterlərdə Linux və Unix əməliyyat sistemlərindən geniş istifadə olunur.

Klaster sistemlərinə misal olaraq qrid hesablama sistemlərini də (Grid Computing) göstərmək olar. Qrid - regional və milli şəbəkələrin birləşdirilməsini nəzərdə tutan və beləliklə, geniş istifadəçi dairəsi üçün açıq olan universal hesablama sistemidir. Hal-hazırda bir çox Qrid layihəsi var. Bunlara misal olaraq, DataGrid, TeraGrid, NorduGrid və s göstərmək olar [7].

III. PARALEL PROQRAMLADIRMANIN TƏSNİFATI

Paralel proqramlaşdırma hesablamaların və proseslərin bir çoxunun eyni zamanda aparıldığı hesablama tipidir. Böyük məsələləri bir çox hallarda daha kiçik hissələrə bölmək mümkündür. Daha sonra bu kiçik məsələ hissələrini eyni zamanda həll etmək mümkündür.

Paralel hesablamaların bir neçə təsnifatı vardır :

- Bit-level (Bit səviyyəli paralellik);
- Instruction (Təlimatların paralelliyi);
- Data parallelism (Verilənlərin paralelliyi);
- Task parallelism (Tapşırıqların paralelliyi).

Bit səviyyəli paralellik paralel hesablamaların bir forması olub, prosessorun bir dəfəyə qəbul edə bildiyi sözverilənlərin ölçüsünün artırılmasına əsaslanır. Əgər dəyişənin ölçüsü prosessorun bir dəfəyə qəbul edə biləcəyi verilənlərin ölçüsündən böyükdürsə, dövrlərin sayı artacaqdır. Bu da hesablama müddətini artırır. Verilən ölçüsünün artırılması prosessorla ötürülən təlimat sayını azaldır. Məsələn, 8-bitlik prosessor 16-bitlik verilənlər üzərində əməliyyat aparmaq üçün əvvəlcə ilk 8 bit, sonra isə son 8-bit üzərində əməliyyat

aparmalıdır. Halbuki, 16-bitlik prosessor üçün bu cəmi bir əməliyyat deməkdir[8].

Bu forma inteqral sxem texnologiyasının (VLSI, ing. *Very-Large-Scale Integration*) tətbiqindən sonra ortaya çıxmışdır. İnteqral sxemlərin üstünlüyü bir çipdə milyardlarla tranzistorun yerləşdirilməsidir. Kompüter dizaynı və arxitekturası bit səviyyəsində paralellik tətbiq olunaraq yaxşılaşdırılmışdır. Hal-hazırda 64 bitlik prosessorlardan daha geniş istifadə olunur.

Təlimat səviyyəli paralellik bir kompüter proqramında olan təlimatların bir neçəsinin eyni zamanda icra edilməsidir. Təlimat səviyyəli paralelləməyə iki cür yanaşma vardır:

- Hardware level (qurğu səviyyəsi);
- Software level (proqram səviyyəsi).

Qurğu səviyyəli paralelləşmə dinamik paralellikdir. Bu zaman hansı təlimatları paralel icra etmək qərarını prosessor özü verir. Proqram səviyyəsində isə statik paralellik istifadə olunur. Bu zaman kompilyator qərar verir ki, hansı təlimatlar paralel icra olunsun[9]. Pentium prosessorları dinamik səviyyəli paralellik üzərində, Itanium prosessorlar isə statik səviyyəli paralellik üzərində işləyir.

Aşağıdakı üç bənddən ibarət bir məsələ göstərilmişdir:

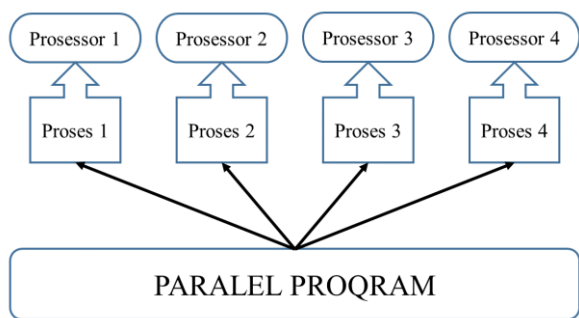
1. $c = a + b$
2. $f = d * e$
3. $x = f / c$

3-cü bənddə göstərilən əməliyyat, 1 və 2-ci bəndlərdəki əməliyyatların nəticəsindən asılı olduğundan bu əməliyyatları hesablamaqdan, 3-cü bənddəki əməliyyata keçmək mümkün deyil. Bununla belə, 1 və 2 bənddəki əməliyyatlar bir-birindən asılı olmadığından onları paralel aparmaq mümkündür. Hər əməliyyatın bir zaman vahidində həll olunduğunu hesab etsək, bu 3 əməliyyat üçün cəmi 2 zaman vahidi tələb olunacaqdır. Ənənəvi proqramlar tipik olaraq ardıcıl icra modeli altında yazıldığından burada təlimatlar bir-birinin ardınca və proqramçı tərəfindən müəyyən edilmiş qaydada icra olunur. Kompilyator və prosessor dizaynerlərinin əsas məqsədi təlimat səviyyəli paralelliyi mümkün qədər daha çox istifadə edərək onun üstünlüklərindən tam yararlanma bilməkdir.

Verilənlər səviyyəsində paralelləşdirmə paralel sistem mühitlərində bir neçə prosessor arasında paralelləşdirmə formasıdır. Verilənləri paralel olaraq işləyən qovşaqlar arasında paylamağa fokuslanır. Müntəzəm verilənlər strukturuna malik olan massivlər və matrislər üzərində aparılan əməliyyatlara tətbiq oluna bilər.

N elementdən ibarət olan massiv üzərində aparılan paralel əməliyyat bütün prosessorlar (və ya nüvələr) arasında bərabər bölünə bilər. Təsəvvür edək ki, biz massivin bütün elementlərini toplamaq istəyirik və hər əməliyyat üçün tələb olunan vaxt T zaman vahididir. Bu zaman əməliyyatlar ardıcıl aparılırsa (məsələn bir prosessorla həll edilərsə), sərf olunan vaxt $T * N$ olacaqdır. Bu əməliyyatlar paralel olaraq 4 prosessorla aparılırsa sərf olunan vaxt $(N / 4) * T$ və alınan nəticələri birləşdirməyə sərf olunan zamanın cəmindən ibarət olacaqdır (Şəkil 2). Təbii ki burada prosessorun keş yaddaşının

həcmi də rol oynayır. Amma ümumi götürdükdə, proses təqribən 4 dəfə sürətlənəcəkdir [9].



Şəkil 2. Verilənlər səviyyəsində paralellik

Tapşırıqların paralelliyifunksiya paralelliyi kimi də tanınır. Çoxprocessorlu sistemdə tapşırıqların paralelliyini o zaman əldə etmək olur ki, hər prosessor eyni və ya müxtəlif verilənlər üzərində əməliyyatlar apararkən hər əməliyyatı müxtəlif axınlar (thread) vasitəsi ilə aparır. Axınlar müxtəlif və ya eyni kodu icra edə bilər. Ümumi halda müxtəlif icraedici axınlar əməliyyatlar apararkən bir-biri ilə əlaqədə olurlar. Əlaqə adətən verilənlərin bir axından digərinə prosesin bir hissəsi kimi ötürülməsi formasında olur. Məsələn, sistem 2 prosessorlu (CPU 1 və CPU 2) bir kompüterdə A və B tapşırıqlarından ibarət kodun bir hissəsini icra edir. Paralel mühitdə A tapşırığını 1 nömrəli, B tapşırığını 2 nömrəli prosessorla yönəldərək onların eyni zamanda həll olunmasına nail olmaq mümkündür:

```
if CPU="a" then do task "A" else if
CPU="b" then do task "B" end if
end program
```

Kanal (axın) səviyyəsində paralellik daha çox tətbiqi məsələlərin həllində (proqramın) özünə xas xüsusiyyətdir. Çünki eyni anda bir neçə kanalın istifadə olunmasını proqram təminatı özü stimullaşdırır. Bu tip paralellik daha çox kommuniya xarakterli məsələlərin həlli üçün istifadə edilərsəverlər üçün nəzərdə tutulmuş verilənlər bazası kimi proqram təminatlarında olur. Belə proqramlarda nəticələr yaddaşa yazıldığı zaman gecikmələr olur. Yaddaş qurğusuna yazılma (və ya yaddaşdan oxunma) zamanı əməliyyatı aparən kanal gözləmə rejimində olur. Məhz buna görə də eyni anda birdən çox kanalı aktivləşdirmək faydalı iş əmsalını artırır, bəzi kanal gözləmə rejimində olduğu zaman digər kanallarla başqa əməliyyatlar aparır [10].

IV. PARALEL PROQRAMLAŞDIRMA ÜÇÜN PROQRAM TƏMİNATI

Paralel proqramlaşdırmada müxtəlif proqram interfeysləri istifadə olunur. Bunlara misal kimi OpenMP, MPI (*Massively Parallel Interface*), PVM (*Parallel Virtual Machine*), CUDA və s. göstərmək olar.

Open MP (Open Multi Processing) 1997-ci ildə yaradılmışdır. Ümumi yaddaşlı çoxnövəli və çoxprosessorlu hesablama sistemləri (SMP arxitekturalı) altproqramlara bölünmüş proqramın paralel yerinə yetirilməsini təmin edir.

OpenMP bir növ Tətbiqi Proqramlaşdırma İnterfeysidir (API, Application Programming Interface). O özündə kitabxanaları (tətbiqi proqram əlavələrini), kompilyator direktivlərini və əməliyyat sistemi sazlamalarını saxlayır. Yalnız paylaşılan (*shared*) yaddaş sistemlərini hədəf alır.

OpenMP interfeysi bir çox kompilyatorlara əlavə olunmuşdur. Aşağıda OpenMP interfeysini dəstəkləyən kompilyatorlar göstərilmişdir :

- Visual C++ (2005-ci ildən etibarən);
- İntel parallel Studio;
- Oracle Solaris Studio;
- C;
- C++;
- Fortran;
- GCC.

MPI (Message Passing Interface – məlumatın ötürülməsi interfeysi) 1993-cü ildə yaradılmışdır. Paylanmış yaddaşlı çoxnövəli və çoxprosessorlu hesablama sistemlərində (MPP arxitekturalı) alt proqramlara bölünmüş proqramın paralel yerinə yetirilməsini təmin edir. MPI da Tətbiqi Proqramlaşdırma İnterfeysidir. O da özündə kitabxanaları (tətbiqi proqram əlavələrini), kompilyator direktivlərini və əməliyyat sistemi sazlamalarını saxlayır. Open MP-dən fərqli olaraqMPI həm paylaşılan (*shared*) həm də paylanmış (*distributed*) yaddaş sistemlərini hədəf alır[11].

MPI interfeysi daha çox iki proqram təminatında tətbiq olunub:

- MPICH;
- Open MPI.

Bu proqramlar yükləndikdən sonra aşağıdakı kompilyatorlarda MPI interfeysi aktiv olacaqdır:

- C;
- C++;
- Fortran;
- R;
- Python;
- Matlab;
- və s

PVM sistemi (Parallel Virtual Machine) şəbəkəyə qoşulmuş bir neçə kompüter və ya serveri bir virtual paralel hesablama maşınında birləşdirmək üçün yaradılıb. Sistem UNIX əməliyyat sisteminə sazlanır və MPP sistemləri də daxil olmaqla müxtəlif platformalarda işləyə bilər.

PVM proqramının əsas prinsipləri:

- Hesablama sisteminin bircinsli olması. Sistem həm təkprosessorlu, həm də çoxprosessorlu maşınları özündə birləşdirə bilər. Bu çoxprosessorlu maşınlar həm paylanan, həm də paylaşılan yaddaşlı ola bilər;

• Hesablama maşınlarının sayı istifadəçi tərəfindən konfigurasiya oluna bilər. Hər hansı birməsələnin həll olunması üçün lazım olan maşın, sistemdə olan istənilən maşınlardan istənilən sayda seçilərək formalaşdırılır. Maşınların sayı proses zamanı istifadəçi tərəfindən azaldıla və ya artırıla bilər;

• Məsələnin həlli üçün tətbiq edilən proqramlar maşınların resurslarını bütöv bir hesablama elementləri kimi görür;

• Proseslər arasında və maşınlar arasında müxtəlif tip verilənlərin mübadiləsi, mesaj mübadiləsi qaydasına uyğun aparılır [12].

PVM-in strukturu iki hissədən ibarətdir:

• **Demon-proqram** – Virtual maşının yaradılması üçün əlavə olunan bütün kompüterlərə əlavə olunur;

• **PVM alt proqramlar kitabxanası** – özündə istifadəçi tərəfindən çağırılaraq məlumat mübadiləsi, proseslərin yaradılması, məsələlərin birləşdirilməsi və virtual maşının modifikasiya olunması üçün istifadə olunan altproqramları saxlayır.

PVM C, C++ və Fortran77 proqramlaşdırma dillərini dəstəkləyir.

Klaster sistemi qurmaq üçün müxtəlif proqramlardan və əməliyyat sistemlərindən istifadə olunur. Populyar əməliyyat sistemlərindən olan Microsoft Windows Server əməliyyat sistemində klaster qurmaq üçün bir neçə servisi aktivləşdirmək lazımdır: Windows Server Failover Clustering və Network Load Balancing. Bundan əlavə 2006-cı ilin MPI iyun ayında, Microsoft şirkəti Windows Compute Cluster Server 2003 əməliyyat sistemini təqdim etmişdi. Bu Microsoft şirkətinin təqdim etdiyi ilk yüksək məhsuldarlıqlı hesablama (HPC, High Performance Computing) klasteri texnologiyası idi.

Adi fərdi kompüterlərdən klaster sistemi qurmaq üçün Beowulf Cluster proqramı istifadə olunur. Bunun üçün minimal üç Linux əməliyyat sistemi yüklənmiş fərdi kompüter olmalıdır. Maddi-texniki bazası kiçik olan institutlar və ya test etmək üçün bahalı avadanlığa sahib olmayan insanlar adətən bu üsuldən istifadə edirlər.

Daha bir klaster sistemi Red Hat Linux tərəfindən istifadəyə verilib və Red Hat Cluster Suite adlanır. Bu sistem 128 hesablama qovşağını özündə birləşdirə bilər. Klaster sistemlərinə misal olaraq Veritas şirkətinin Veritas Cluster System, Sun Microsystems şirkətinin Oracle Solaris Cluster (bəzən Sun Cluster də adlandırılır) və San Diego Superkompüter Mərkəzi (SDSC, San Diego Supercomputer Center) tərəfindən yaradılan Rocks Cluster Distribution-u göstərmək olar.

AzScienceNet elm-kompüter şəbəkəsinin Data Mərkəzində Rocks Cluster proqram təminatından istifadə olunur. Bu proqram təminatı CentOS Linux əməliyyat sisteminin üzərinə yüklənir [13]. Bu sistemin test olunması üçün klasterə Linpack proqram təminatı yüklənmişdir. Linpack proqramı superkompüterlərin və hesablama klasterlərinin məhsuldarlığını yoxlamaq üçün alətdir. Onun içində sürüşkən vergüllü ədədlər, matrislər və s. üzərində müxtəlif

əməliyyatlar apara bilən xüsusi proqram kodları var. Bu proqram təminatı vasitəsilə ölçüsü 20000 olan 20000 ədəd matrisin hasil hesablanmışdır. Nəticədə proqram 1 hesablama qovşağında (CPU işləmə tezliyi 2.7 GHz, əməli yaddaş 80 GB, hər mikroprosessorada olan nüvələrin sayı 32) hesablama apararkən nəticəni 121.9 saniyəyə, 10 hesablama qovşağında paralel hesablama apararkən isə 13.11 saniyəyə almışdır. Alınan nəticələrin bir fiziki maşında alınan nəticələrlə müqayisəsi hesablamağa sarf olunan zamanın təqribən 9.3 dəfə az olduğunu göstərir. Beləliklə, paralel hesablama texnologiyalarının tətbiqi faydalı iş əmsalının dəfələrlə artırılmasına gətirib çıxarır.

NƏTİCƏ

Məqalədə paralel emal sistemlərinin təsnifatı və arxitekturası analiz edilmişdir. Müxtəlif səviyyələrdə paralelliyin əldə olunması formaları göstərilmişdir. Məsələlərin paralelləşdirilərək həll olunması üçün istifadə olunan proqram təminatları haqqında geniş məlumat verilmişdir. Data Mərkəzdə Rock Cluster əməliyyat sistemindən yüklənmiş və 10 qovşaqdan ibarət klaster sistemi üzərində paralel hesablama aparılmışdır.

ƏDƏBİYYAT

- [1] К.Е.Афанасьев, «Многопроцессорные вычислительные системы и параллельное программирование», Кемеровский государственный университет, 2003, 233 с.
- [2] V.Rajaraman, T.Radhakrishnan, “Computer organization and architecture”, New Delhi : Prentice-Hall of India, 2007, 493 p.
- [3] https://en.wikipedia.org/wiki/Gustafson's_law
- [4] R.Q. Ələkbərov, “Superkompüter texnologiyaları: Mövcud vəziyyəti və inkişaf perspektivləri”, İnformasiya cəmiyyəti problemləri, 2016, №2, səh. 26–36.
- [5] В. Л.Бройдо, О. П. Ильина, «Архитектура ЭВМ и систем», Учебник для вузов. 2-изд., 2009, 720 с.
- [6] D. Sima, “Terence Fountain and Peter Kascuk”, Advanced Computer Architectures a design space approach, Addison-Wesley, 1997, 766 p.
- [7] R.Q. Ələkbərov, S.M. Dursunov. Elektron tibdə qrid texnologiyalarının tətbiqi. “Elektron tibbin multidissiplinar problemləri” I respublika elmi-praktiki konfransı. - Bakı, 2016., səh. 161-163.
- [8] А.В.Старченко, В.Н.Берцун, «Методы параллельных вычислений» 2013, 223 с.
- [9] H.L. John, P.A. David, “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann; 5 edition, 2011, 856 s.
- [10] P. Pacheco, “An Introduction to Parallel Programming,” Morgan Kaufmann; 1 edition, 2011, 392 s.
- [11] W. Gropp, E. Lusk, A. Skjellum, “Using MPI: Portable Parallel Programming with the Message Passing Interface”, The MIT Press: second edition edition, 1999, 350 s.
- [12] “Recent Advances in PVM and MPI”, 10th European PVM/MPI Users' Group Meeting, Venice, Italy. Proceedings, Vol. 10, 2003, 693 s.
- [13] “Rocks Cluster Distribution: Users' Guide”, <http://www.Rocksclusters.Org/Rocks-Documentation/4.1/>

Bu iş Azərbaycan Respublikasının Prezidenti yanında Elmin İnkişafı Fondunun maliyyə yardımı ilə yerinə yetirilmişdir – Qrant № EİF-2014-9(24)-KETPL-14/02/1