

K.A.SALMANOVA

**Paralel kompüter sistemlərinin arxitekturası və
proqramlaşdırma
(*dərs vəsaiti*)**

Gəncə Dövlət Universitetinin 17.02.2023-cü il tarixli 5/36 sayılı əmri və Riyaziyyat-İnformatika fakültə Elmi Şurasının 31.01.2023-cü il tarixli 07 sayılı protokoluna əsasən təsdiq edilmişdir.

Gəncə-2023

Elmi redaktor: fizika-riyaziyyat elmlər namizədi, dosent **Əkrəm Əhməd oğlu Aslanov**

Rəy verənlər: texnika elimləri doktoru, professor **Verdiyev Sakit Qambay oğlu**, texnika fəlsəfə doktoru, dosenti **Məmmədov Mahmud Neyman oğlu**, fizika-riyaziyyat elmlər namizədi, dosenti **Hüseynov Sahib Tofiq oğlu**

Salmanova K.A. Paralel kompüter sistemlərinin arxitekturası və proqramlaşdırma. Dərs vəsaiti/-Gəncə: GDU, 2023-118s.

Kitabda elmin bir çox sahələrində meydana çıxan çox böyük hesablama və yaddaş resursları tələb edən qarışıq məsələlərin həllində paralel proqramlaşdırma texnologiyasından istifadə məsələləri qeyd edilmişdir. Burada paralel proqramlaşdırma zamanı istifadə olunan alqoritmlər və platformalar haqqında geniş izahatlar verilmişdir.

Bu vəsaitdə müasir hesablama sistemlərinin arxitekturası haqqında müasir məlumatlar var. Vəsaitdə paralel hesablama sistemləri və proqramlaşdırma haqqında əsas məsələlərin öyrənilməsi məqsədilə 8 praktik mövzu verilmişdir.

Kitabda paralel hesablama sistemləri haqqında ümumi məlumat verilmiş, bu çür sistemlərin qurulma prinsipi, iş prinsipləri təhlil edilmişdir. Bu məqsədlə bu bölmədə paralel hesablama sistemlərinin təsnifatı, proqramlaşdırma modelləri: SİSD, SİMD, MİMD arxitekturaları və paralellik məsələləri və s. izah edilmişdir.

Tərtib olunmuş dərs vəsaiti informatika və digər istiqamətli ixtisasların bakalavr və magistr pilləsində təhsil alan tələbələr üçün nəzərdə tutulmuşdur. Dərs vəsaitindən təhsil alan tələbələr, eləcə də elmi işçilər istifadə edə bilərlər.

GİRİŞ

İnsan fəaliyyətinin bütün sahələrinin müasir rəqəmsallaşmasının əsasını təşkil edən kompüter sistemləri intensiv elmi, mühəndis-texniki inkişafdan keçir. Kompüterləşdirmə imkanları nə qədər yüksək olarsa, kompüter sistemlərinə bir o qədər mürəkkəb tələblər qoyulur. Bu vəsaitdə müasir hesablama sistemlərinin arxitekturası haqqında müasir məlumatlar var. Vəsaitdə paralel hesablama sistemləri və proqramlaşdırma haqqında əsas məsələlərin öyrənilməsi məqsədilə 8 praktik mövzu verilmişdir.

Birinci praktik məşğələdə paralel hesablama sistemləri haqqında ümumi məlumat verilmiş, bu çür sistemlərin qurulma prinsipi, iş prinsipləri təhlil edilmişdir. bu məqsədlə bu bölmədə paralel hesablama sistemlərinin təsnifatı, proqramlaşdırma modelləri: SİSD, SİMD, MİMD arxitekturaları və paralellik məsələləri izah edilmişdir.

Praktiki məşğələ 2-də müasir super kompüterlər və onların növləri, modellərini, qurulma prinsiplərini, super kompüter anlayışının tərifi, paralel vektor sistemləri, ümumi yaddaş olan simmetrik multi-processor sistemləri, bölüşdürülmüş yaddaşa malik massiv-paralel sistemlər, yaddaşa müraciəti bərabər olmayan sistemlər, super kompüterlərin məhsuldarlığı və proqram təminatı araşdırılır.

Praktiki məşğələ 3-də müasir kompüterlərin yaddaşının öyrənilməsi, onların növləri, yaddaşın təşkilinin təsnifatları təhlil edilir. Bu məqsədlə paylaşılan və ayrılmış yaddaşa

malik multiprocessorlar və multikompyuterlər, informasiya mübadiləsi interfeysi texnologiyası və s. məsələlər öyrənilir.

Praktiki məşğələ 4-də paralel kompüterlərin strukturunu öyrənilir, praktiki klassifikasiya təhlil edilir, paralel massiv sistemləri araşdırılır.

Praktiki məşğələ 5-də kommunikasiya, şəbəkə, şəbəkələrin qurulma üsulları, şəbəkə qurğuları, şəbəkə topologiyaları: şin, ulduz və halqa topologiyaları təhlil edilir.

Praktiki məşğələ 6-da paralel proqramlaşdırmanın əsas anlayışları və tətbiq sahələri şərh edilir.

Praktiki məşğələ 7-də OpenMP mühiti, əsas pəncərə, komanda sətiri ilə tanış olmaq, interfeysin öyrənilməsi məqsədilə zəruri məlumatlar verilir.

Praktiki məşğələ 8-də paralel proqramlaşdırma texnologiyalarının əsas məsələləri haqqında anlayış verilir.

Praktiki məşğələ №1

Mövzu: Paralel hesablama sistemləri

İşin məqsədi: *Paralel hesablama sistemləri haqqında ümumi məlumat. Onların qurulma prinsipini, iş prinsiplərini öyrənmək.*

Nəzəri hissə.

Paralel hesablamlar böyük məsələlərin eyni vaxtda yerinə yetirilə bilən kiçik məsələlərə bölünməsidir. Tipik olaraq, paralel hesablama müəyyən əlaqələndirilmiş hərəkətlər tələb edir. Paralel hesablama bir neçə formada olur (təlimatlar, bitlər, məlumatlar, tapşırıqlar səviyyəsində). Paralel hesablama illər ərzində tətbiqini əsasən yüksək performanslı hesablamalarda tapdı. Ancaq son vaxtlar vəziyyət dəyişdi. Prosessorun saat tezliyinin artmasının fiziki məhdudiyətləri səbəbindən bu cür hesablamalara tələbat var. Paralel hesablama kompüter arxitekturasında dominant fikir oldu. Çox nüvəli prosessorlar formasını aldı¹.

Paralel hesablama sistemləri fiziki kompüter sistemləri, eləcə də bir çox hesablama qovşaqlarında bu və ya digər şəkildə paralel verilənlərin emalını həyata keçirən program sistemləridir.

¹https://studme.org/151249/informatika/arhitektura_parallelnyh_vychislitelnyh_sistem

Paralel hesablama proqramların bir-biri ilə paralel (eyni vaxtda) icra edildiyi (yerinə yetirildiyi) hesablama proseslərinin prosessorlarının məcmusu kimi kompüter hesablamlarının təşkili üsuludur. Bu ifadə proqramlaşdırmada paralellik problemləri toplusunu, eləcə də səmərəli cihazların icra xüsusiyyətlərini əhatə edir. Paralel hesablamlar nəzəriyyəsi praktiki alqoritmlər nəzəriyyəsinin yaranmasına səbəb oldu.

Paralel hesablamları həyata keçirməyin müxtəlif yolları var. Məsələn, hər bir hesablama sistemi, əməliyyat sistemi-sistem prosesi kimi həyata keçirilə bilər və ya hesablama prosesləri vahid əməliyyat sistemi prosesi daxilində icraçı tellər toplusunu təmsil edə bilər. Paralel proqramlar fiziki baxımdan bir prosessorada yerinə yetirilə bilər.

Paralel proqramların layihələndirilməsində əsas mürəkkəblik ondan ibarətdir ki, müxtəlif hesablama sistemləri ilə qarşılıqlı əlaqəni, eləcə də proseslər arasında paylanmış resursların düzgün ardıcılıqla əlaqələndirilməsini təmin etmək lazımdır.

Eyni vaxtda bir neçə hərəkətin yerinə yetirilməsi ideyasından ibarət olan məlumatların paralel hesablanmasına dair iki fərqli baxış mövcuddur. Bunlar paralel və konveyer qurşaqlarıdır. Əgər cihaz zaman vahidində bir hərəkəti yerinə yetirirsə, o zaman min zaman vahidində min hərəkəti yerinə yetirir. Əgər beş cihaz eyni anda və bir-birindən müstəqil işləmək qabiliyyətinə malik hesab edilərsə, onda onlar yuxarıdakı min hərəkəti min vaxt vahidində deyil, iki yüz dəfə bir vahiddə yerinə yetirəcəklər. N cihazlarından ibarət sistem

1000/N-lıq bir vaxtda 1000 hərəkət yerinə yetirir. Buna bənzər vəziyyətləri həyatdan da gətirmək olar. Məsələn, bir əsgər 10 saata polis məntəqəsini işə sərsəmləsə, 50 əsgərdən ibarət yol eyni vaxtda fəaliyyət göstərəcək və 12 dəqiqəyə polisləri işə siləcək. Paralel hərəkətlər prinsipi belədir².

Paralel hesablama sistemlərinin təsnifatı

Hesablama sisteminin arxitekturası proqramçı nöqtəyindən nəzərdən kompüterin mücərrəd təsviri kimi başa düşülür. Sistem arxitekturasının tam təsvirinə aşağıdakılar daxildir:

- məlumatların əsas təqdimat formatları;
- proqramda verilənlərin ünvanlanması yolları;
- kompüterin texniki vasitələrinin tərkibi, bu vasitələrin xüsusiyyətləri, hesablama prosesinin təşkili prinsipləri.

Hesablama sisteminin strukturu, aralarındakı əsas əlaqələri göstərən kompüter avadanlığı dəsti kimi müəyyən edilə bilər.

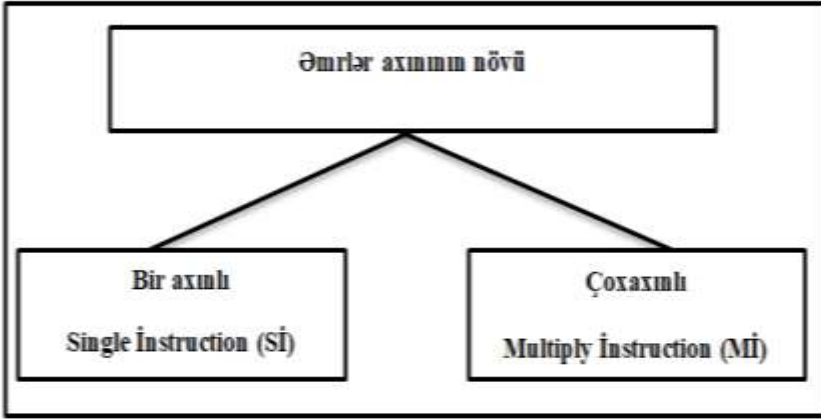
Hesablama sistemlərinin bir çox müxtəlif təsnifatı var. Ən çox istifadə olunan təsnifatları nəzərdən keçirin.

Flinin təsnifatı.

1966-cı ildə Stenford Universitetinin professoru M.D.Flinin (M.J.Flynn) təklif etdiyi hesablama sistemlərinin təsnifatı daha çox yayılmışdır. Bu təsnifat yalnız iki təsnifat

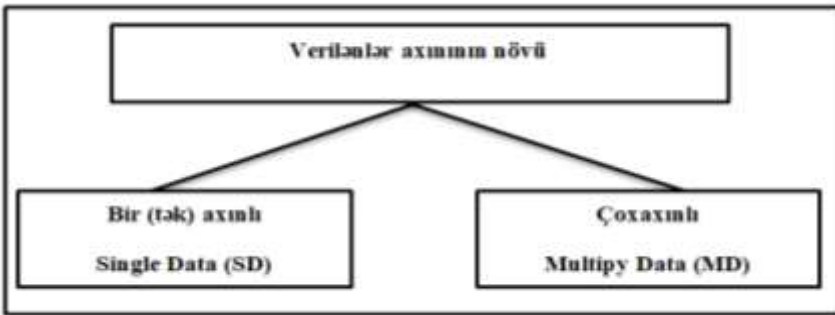
²<http://bigor.bmstu.ru/?cnt/?doc=Parallel/ch010101.mod/?cou=Parallel/base.cou>

əlamətini əhatə edir - əmrlər axını növü və verilənlər axını növü (bax: Şəkil 1.1.).



Şəkil. 1.1. Flinin təsnifatı. Əmrlər axınının növünə görə təsnifat.

Tək bir əmr axınında bir anda yalnız bir əmr yerinə yetirilə bilər. Bu halda, bu tək əmr kompüter sisteminin bütün və ya ən azı bir çox qurğusunun müəyyən bir zamanda işləməsini müəyyən edir.



Şəkil 1.2. Flinin təsnifatı. Verilənlər axınının növünə görə təsnifat.

Çoxlu əmərlər axınında bir çox əmr eyni vaxtda yerinə yetirilə bilər. Bu halda, bu əmərlərin hər biri müəyyən bir zamanda kompüter sisteminin yalnız bir və ya bir neçə (lakin hamısı deyil) qurğusunun işini müəyyən edir.

Tək axında fərdi əmərlər ardıcıl olaraq, çoxlu axında isə əmərlər qrupları yerinə yetirilir. Tək (bir) verilənlər axını mütləq hesablama sistemində yalnız bir operativ yaddaş qurğusunun və bir prosessorun olmasını nəzərdə tutur. Lakin prosessor istənilən qədər mürəkkəb ola bilər, belə ki, axındakı hər bir informasiya vahidinin işlənməsi prosesi bir çox əmərlərin yerinə yetirilməsini tələb edə bilər.

Çoxlu verilənlər axını bir çox asılı və ya müstəqil tək verilənlər axınlarından ibarətdir.

Deyilənlərə əsasən, bütün hesablama sistemləri dörd növə bölünür:

SISD (OKOD);

MISD (MKOD);

SIMD (OKMD);

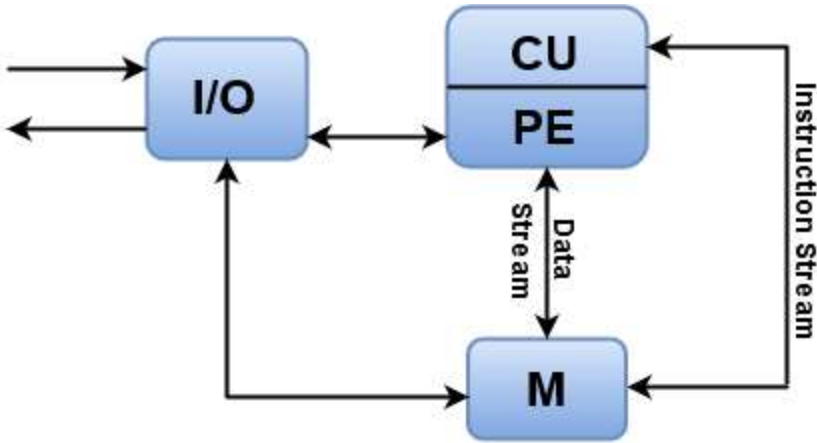
MIMD (MKMD).

Proqramlaşdırma modeli – mücərrəd kompüter arxitekturasına cavab verən və alqoritmlərin müəyyən sinfini həyata keçirmək üçün nəzərdə tutulmuş proqramlaşdırma metodları kompleksidir. Proqramlaşdırma modelləri kompüterin və onun arxitekturasının məntiqi təşkil edilməsi haqqında aydın təsəvvürə malik olmağa xidmət edir.

SISD (Single Instruction Single Data) — "Tək əmr və Tək verilən axını" deməkdir. O, idarəetmə bloku, prosessor bloku və yaddaş blokundan ibarət vahid kompüterin təşkilini təmsil edir.

Əmrlər ardıcıl olaraq yerinə yetirilir və sistem daxili paralel emal imkanlarına malik ola və ya olmaya bilər.

Ənənəvi kompüterlərin əksəriyyəti ənənəvi Con-Neymanın kompüterləri kimi SISD arxitekturasına malikdir. Bu halda paralel emal bir çox funksional bloklar vasitəsilə və ya konveyer emalı vasitəsilə əldə edilə bilər.



CU = Control Unit (İdarəetmə qurğusu(bloku)),

PE = Processing Element (Emal elementi (HMQ)),

M = Memory (Yaddaş qurğusu)

Şəkil 1.3. SISD Arxitekturası³

³ <https://www.javatpoint.com/sisd>

Əmrlər idarəetmə bloku ilə kodlaşdırılır, sonra isə idarəetmə bloku emal bloklarına əmr göndərir.

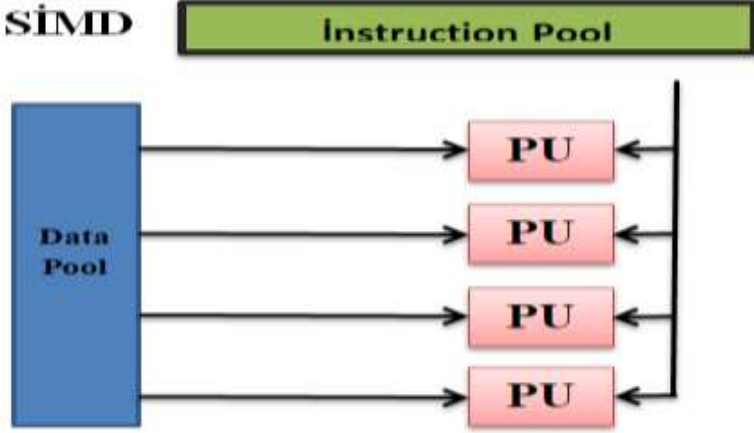
Verilənlər axını prosessorlarla yaddaş arasında ikiqat keçir.

Nümunə:

İlk nəsil kompüterlər, mini-kompüterlər və iş stansiyaları ***SIMD (Single Instruction Multical Data)*** — axın əmri, çoxlu axın məlumatları.

Tək instruksiya eyni çıxışı yerinə yetirmək üçün çoxlu məlumat elementinə tətbiq olunur. Master instruksiya ciddi klidləmə yanaşması ilə eyni təlimatı bir saat dövrü işləyən prosessorlarının operand say vektoru üzərində işləyir. Bu, bir neçə Emal Elementi/Yaddaş Modulları arasında paralel sinxron əlaqə yaratmağa imkan verən təlimat səviyyəli paralellik rabitə şəbəkəsinin növüdür.

Şəkil 4-də SIMD prosessorunun arxitekturası iki SIMD arxitekturasından sonra mesaj ötürmə paradiqmasına əsaslanan məlumatların paralel işlənməsi üçün əsaslı şəkildə fərqli yanaşmaları təsvir edir. Burada yaddaş PE-nin bir hissəsidir və beləliklə, məlumatların ötürülməsi üçün qarşılıqlı əlaqə şəbəkəsi vasitəsilə əlaqə qurur.

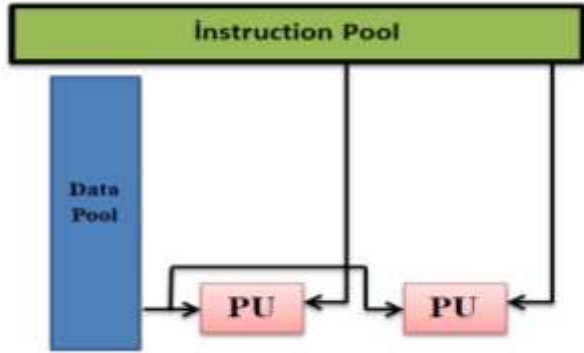


Şəki1.4. SIMD arxitekturası

Bir axın əmri, çoxlu axın məlumatları (SIMD) olan kommutatorlarda eyni əməliyyatı müxtəlif məlumatlarla yerinə yetirə bilən çoxlu prosessorlar var. SIMD - aparatlar veksellər üzərində hərəkətləri yerinə yetirmək üçün çox rahat olduqları üçün bəzən vektor prosessorları da adlandırılır. Bu halda hər bir prosessorla bir vektor həmrəyliyi verilir, hərəkət başa çatdıqdan sonra isə nəticə vektor olur.

MISD (Multical Instruction Single Data) — axın verilənləri, çoxlu axın əmrləri.

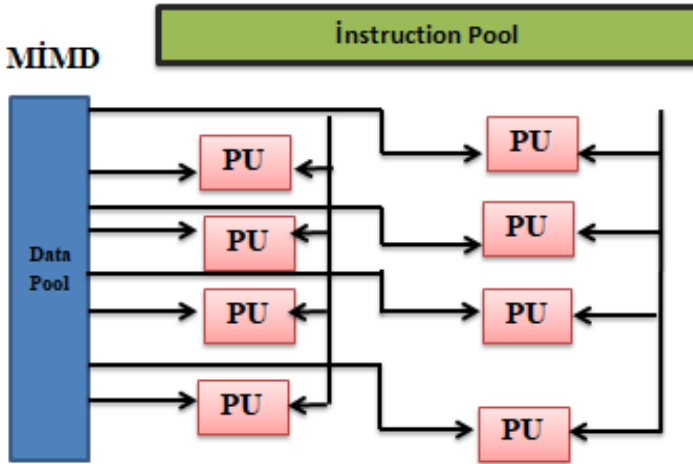
MİSD



Şəkil 1.5. MİSD arxitekturası

Eyni anda yalnız eyni verilənlər üzərində bir hərəkətin həyata keçirilməsi ilk növbədə qəribə görünə bilər. Çünki bir rəqəmi kvadrata qaldırmaq, ikiyə çoxalmaq, onluqlara bölmək kimi tətbiqlər nadir hallarda rast gəlinir. Amma fərqli nöqtəyindən nəzərdən görürük ki, belə maşınlarda ədədin dibinin və ya mürəkkəbliyinin yoxlanılmasını təkmilləşdirmək mümkündür. MİSD – bir əməliyyatda aparat vasitəsi ilə yoxlaya bilərik. Əgər X ədədi mürəkkəbdirsə, onda ona uyğun olmayan bölücü olmalıdır.

MİMD (Multical Instruction Multical Data) çox axın əmrləri, çoxlu axın məlumatları.



Şəkil 1.6. MIMD arxitekturası

Bu kateqoriya kateqoriyalar arasında daha mürəkkəbdir. MIMD sistemləri məsələsində biz öz qaydalarını həyata keçirə bilən bir neçə prosessorla işləyirik. Bir neçə məlumat axını da mövcuddur və istənilən prosessor öz datasetini həll edə bilər.

Paralel arxitekturalar

Paralel hesablama sistemlərinin (PCS) və superkompüterlərin istifadəsi kompüter texnologiyasının inkişafında strateji istiqamətdir [1]. Bu, təkcə adi seriyalı kompüterlərin mümkün olan maksimal sürətinin əsaslı şəkildə məhdudlaşdırılması ilə deyil, həm də mövcud kompüter texnologiyalarının imkanlarının həmişə qeyri-kafi olduğu

hesablama tapşırıqlarının demək olar ki, daimi olması ilə əlaqədardır.

Bu vəziyyətdən çıxış yolu belədir. Əgər bir kompüter problemi lazımi vaxtda həll edə bilmirsə, onda gəlin iki, üç, on kompüter götürüb onları eyni vaxtda ümumi problemin müxtəlif hissələrində işlətməyə çalışaq və müvafiq sürətlənmə əldə edək.

Kompüterlərin vahid sistemdə birləşdirilməsi bir çox nəticələrə səbəb olurdu. Fərdi kompüterləri işlə təmin etmək üçün orijinal tapşırığı bir-birindən asılı olmayaraq yerinə yetirilə bilən fraqmentlərə bölmək lazımdır. Belə bir ayrılmanın mümkünlüyünə imkan verən xüsusi ədədi üsullar belə görünməyə başladı. Tapşırığın müxtəlif hissələrinin müxtəlif kompüterlərdə eyni vaxtda necə yerinə yetirildiyini təsvir etmək üçün xüsusi proqramlaşdırma dilləri, xüsusi əməliyyat sistemləri və s. lazım idi. Tədricən "paralel" sözü –"sinxron", "müstəqil" və buna bənzər sözlər tək sözlə əvəz olunmağa başladı. Bəzi proseslərin, hərəkətlərin, faktların, bir-biri ilə əlaqəsi olmayan halların təsvirini nəzərdə tutsaq, bütün bunlar sinonimdir. Kompüterlə bağlı sahələrdə "paralellik" və "paralel" sözləri başqa heç nə ifadə etmir.

Paralel hesablama sistemlərinin təşkili üçün bir çox üsullar mövcuddur [1, 2]. Hesablamaların paralelliyi, eyni anda bir neçə məlumat emal əməliyyatı yerinə yetirildikdə, əsasən funksional bölmələrin (çoxprosessorlu) artıqlığının tətbiqi hesabına həyata keçirilir [3]. Bu zaman tətbiq olunan alqoritmi informasiya baxımından müstəqil hissələrə bölmək və hesablamaların hər bir hissəsinin müxtəlif funksional

cihazlarda (prosessorlar, nüvələr, hesab məntiqi vahidləri və s.) icrasını təşkil etməklə hesablama məsələsinin həlli prosesini sürətləndirmək mümkündür. Bu yanaşma hesablamların ardıcıl təşkili ilə müqayisədə lazımi hesablamları daha az vaxt sərf etməyə imkan verir. Maksimum sürəti əldə etmək imkanı (ən azı prinsipçə) yalnız mövcud prosessorların sayı və hesablamalarda paralel icra edən hissələrin sayı ilə məhdudlaşır.

Paralel hesablamların təşkili problemlərini nəzərə alaraq, proqramın müstəqil hissələrinin icrasının aşağıdakı tanınmış rejimlərini fərqləndirmək lazımdır:

- bir neçə prosesi yerinə yetirmək üçün tək prosessordan istifadə edilən çoxtapşırıqlı (məsələli) rejim (vaxtın paylaşılması (bölünməsi) rejimi). Bu rejim psevdoparaleldir: yalnız bir proses aktiv ola bilər (icra edilə bilər) və bütün digər proseslər öz növbəsini gözləyir. Vaxt bölgüsü rejiminin istifadəsi hesablamların təşkilinin səmərəliliyini artırır bilər (məsələn, giriş məlumatlarını gözləmək səbəbindən proseslərdən biri yerinə yetirilə bilmirsə, prosessor hazır olan başqa bir prosesi yerinə yetirmək üçün istifadə ediləcəkdir). Bundan əlavə, bu rejimdə paralel hesablamanın bir çox effektləri meydana çıxır (proseslərin bir-birini istisna etməsi və sinkronlaşdırılması ehtiyacı və s.) və nəticədə bu rejim paralel proqramların ilkin hazırlanmasında istifadə edilə bilər;

- Paralel hesablama, eyni zamanda bir neçə verilənlərin emal əmri yerinə yetirilə bildikdə. Belə bir hesablama rejimi yalnız bir neçə prosessorun istifadəsi ilə deyil, həm də boru

kəməri və vektor emal qurğularının köməyi ilə təmin edilə bilər;

- paylanmış hesablama – adətən bir-birindən kifayət qədər uzaqda olan bir neçə emal qurğusu istifadə edildikdə, məlumatların rabitə xətləri ilə ötürülməsi əhəmiyyətli vaxt gecikmələrinə səbəb olan paralel məlumatların işlənməsini ifadə etmək üçün istifadə olunan termindir. Nəticədə, hesablamaların təşkilinin bu üsulu ilə məlumatların səmərəli işlənməsi yalnız prosessorlararası məlumat ötürülməsi axınlarının aşağı intensivliyi ilə paralel alqoritmlər üçün mümkündür. Yuxarıda göstərilən şərtlər, məsələn, yerli və ya qlobal informasiya şəbəkələrinin rabitə kanallarından istifadə edərək bir neçə ayrı-ayrı kompüterləri birləşdirərək formalaşan çoxkompüter komplekslərində hesablamaların təşkili zamanı tipikdir.

Problemlərin həlli üçün paralel alqoritmlər hazırlayarkən əsas məqam paralellikdən [4] istifadənin səmərəliliyinin təhlilidir ki, bu da adətən hesablama prosesinin nəticədə sürətlənməsini qiymətləndirməkdən ibarətdir (məsələnin həlli üçün vaxtın azaldılması). Bu cür sürətləndirmə təxminlərinin formalaşması seçilmiş hesablama alqoritmi ilə əlaqədar həyata keçirilə bilər (məüyyən bir alqoritmin paralelləşdirilməsinin səmərəliliyinin qiymətləndirilməsi). Başqa bir vacib yanaşma, məüyyən bir növ problemin həllinin əldə edilməsi prosesinin mümkün olan maksimum sürətləndirilməsinin təxminlərinin qurulmasından ibarət ola bilər (problemin həlli üçün paralel metodun effektivliyinin qiymətləndirilməsi).

Paralellyin özü [5] verilənlərin emalı üçün bir neçə (n) cihazın və hər birinə hesablamaların müstəqil hissəsini yerinə yetirməyə imkan verən alqoritmin mövcudluğunu nəzərdə tutur. Emalın sonunda, son nəticə əldə etmək üçün qismən məlumatlar birlikdə toplanır. Bu halda (məlumatların əldə edilməsi və saxlanması üçün əlavə məsrəflərə məhəl qoymadan) prosesin n əmsalı ilə sürətlənməsini əldə etmək mümkündür. Hər bir alqoritmi bu şəkildə uğurla paralelləşdirmək mümkün deyil (paralelləşdirmə üçün təbii şərt eyni və ya oxşar prosedurlardan istifadə etməklə çıxış məlumatlarının müstəqil hissələrinin hesablanmasıdır; iterasiya və ya rekursivlik paralelləşdirmədə ən böyük problemlərə səbəb olur).

Bəzi ədəbiyyat mənbə [3] paralellyin dörd növünü ayırır: bitlər, əmrlər, verilənlər və tapşırıqlar səviyyəsində paralellik.

Bit səviyyəli paralellik

Əməliyyatları həyata keçirərkən maksimum paralelləşmə təmin edilir, məsələn, ədədin bitləri əlavə edildikdə, maşın sözünün bütün bitləri eyni vaxtda əlavə olunur və köçürmələrin uçotu üçün xüsusi mexanizmdən də istifadə olunur [4]. Bir təlimatla işlənən nömrələrin diapazonunu artırmaq üçün maşın sözünün ölçüsünü artırmalısınız. Beləliklə, 8-dən 16-ya və daha sonra 32 bitlik maşın sözünə keçid edildi. Hazırda prosessorlarda 64 bitlik maşın sözü var, bəzi təlimatlar 128 bitlik verilənlərlə işləyə bilər (Larrabee prosessoru 512 bit məlumatlarla işləyir). Daha böyük söz uzunluğu olan maşın sözündən istifadə edərkən, böyük

rəqəmlər daha sürətli emal olunur. 16 bitlik maşın üçün 32 bitlik nömrəni emal etmək üçün ən azı 2 təlimat, 32 bitlik bir maşın üçün isə yalnız bir təlimat tələb olunur. Ancaq bu, bütün 64 bitlik prosessorlar üçün doğru deyil, çünki 64 bitlik nömrələr üzərində bütün əməliyyatlar onlarda həyata keçirilmir. 128 bitlik ədədlər və daha böyük bit dərinliyi olan nömrələrlə nəticələnən əməllərin istifadəsi də məhduddur.

Komanda səviyyəsində paralellik. Konveyer

Mürəkkəb əməliyyatların məhsuldarlığını artırmaq üçün boru konveyerlərdən istifadə ideyası çoxdan məlumdur. Əmr icra müddəti təxminən eyni olan müstəqil mikroəmərlərə bölünür. Mikro təlimatların sayı konveyer bloklarının sayına bərabərdir. Konveyer blokları ardıcıl olaraq növbəti təlimatı emal edir, onun mikro göstərişlərini yerinə yetirir. Əmr yalnız boru kəmərinin sonuncu bloku tərəfindən emal başa çatdıqdan sonra tam icra edilmiş hesab olunur.

Komanda icrasının konveyer prinsipi ilk dəfə 1962-ci ildə Mançester Universitetində hazırlanmış ATLAS maşınında [5] istifadə edilmişdir. Təlimatların yerinə yetirilməsinin 4 mərhələsi var:

1. təlimatın alınması,
2. operand ünvanının hesablanması,
3. operandın alınması
4. əməliyyatın icrası.

Pentium üçün nəticənin yazılması üçün daha bir blok boru xəttinə əlavə edilmişdir. Konveyerin köməyi ilə əməllərin

icra müddətini 6 mikrosaniyədən 1,6 mikrosaniyəyə qədər azaltmaq mümkün olub.

Seriallaşdırma əmrləri

Konveyer əmri paralel olaraq yerinə yetirmək yollarından biri hesab olunur. Prosessor təlimatlarında icraya başlamaq üçün bütün başlanmış təlimatların tamamlanmasını gözləyən xüsusi təlimatlar var. Onlar yerinə yetirildikdə, digər əmrlər yerinə yetirilmir, bu əmrlər seriallaşdırma əmrinin adına malikdir. Seriallaşdırma əmrinə misal olaraq, prosessorun xüsusiyyətlərini təyin etmək üçün istifadə edilən cupid əmrini göstərmək olar. Vaxt ölçmələrinin dəqiqliyini artırmaq üçün bu əmr ölçmələrdən əvvəl və sonunda istifadə edilməlidir.

Paralel kompüter sistemlərinin arxitekturasında iki şey əsas rol oynayır:

- Tərədicilər və onların xatirələri bir-biri ilə necə əlaqəlidir;
- Prosessorların qarşılıqlı təsiri.

Paralel alqoritmləri müzakirə edərkən bu aspektlərdən bəhs edirik. Çünki bu və ya digər həllər müxtəlif məsələlər üçün fərqli effektdə malik ola bilər. Paralel arxitektura da prosessorların əlaqələndirilməsi üçün başqa imkanlar vardır. Bunlara ağac şəklində olan şəbəkələr (prosessorlar hər hansı bir ağacı əmələ gətirir) və ikiölçülü birləşdirən hiperkopiya aiddir.

Paralel Hesablama Sistemləri

Paralel hesablama sistemləri fiziki kompüter sistemləri, eləcə də bir çox hesablama qovşaqlarında bu və ya digər şəkildə paralel verilənlərin emalını həyata keçirən program sistemləridir.

Hesablamaların paralelləşdirilməsi ideyası ona əsaslanır ki, əksər tapşırıqlar eyni vaxtda həll edilə bilən daha kiçik tapşırıqlar toplusuna bölünə bilər. Adətən paralel hesablamalar hərəkətlərin əlaqələndirilməsini tələb edir. Paralel hesablama bir neçə formada olur:

- bit səviyyəli paralellik,
- təlimat səviyyəsində paralellik,
- verilənlər paralelliyi
- tapşırıq paralelliyi.

Paralel hesablamalar uzun illərdir ki, əsasən yüksək məhsuldar hesablamalarda istifadə olunur, lakin son zamanlar prosessorların takt tezliyinin artmasına fiziki məhdudiyətlərin olması səbəbindən maraq artıb. Paralel hesablamalar əsasən çoxnüvəli prosessorlar şəklində kompüter arxitekturasında dominant paradigmaya çevrilmişdir.

Paralel hesablama sistemləri aşağıdakı hallar üçün xarakterikdir:

1. Hesablama sisteminin mərkəzi hissəsinə və xarakter axınının növlərinə əsasən;
2. İnformasiya prosessoru metoduna əsasən;
3. Hesablama sistemlərinin komponentlərinin eyni cinsindən və ya qoşulma səviyyəsindən asılı olaraq.

Hesablama sistemlərinin paralel təşkili ilk iki həriflə müəyyən edilir. OKOD bir prosessorla malik tək-tək hazır personajlara malik olan EHM-lər silsiləsindən ibarətdir.

MKMD və ya bir çox axınlar bir çox əmr axınlarında hərəkətləri yerinə yetirir. Belə MBS prosessorlar BK-yə malik olması ilə assinxron rejimdə fəaliyyət göstərir.

OKMD tipli sistemdə ümumi BK-nin olması bütün prosessorlarda bir ümumi komandanın yerinə yetirilməsini təmin edir.

Bu da sistemin indiki vaxtda bir proqramda tətbiq ediləcəyinə dəlalət edir. Nəticədə rüşeymin qidalı maddələrlə inkişaf etməsinə, sonra isə onun bətninə daxil olmasına icazə verildi. (Matta 24:14; 28:19, 20). Yehovanın Şahidləri məmnuniyyətlə sizinlə bu sualları müzakirə edərlər. Bunun üçün xüsusi yenidən ötürmə tipləri tələb olunur.

Paralel tətbiqləri inkişaf etdirərkən developer tələb olunur:

1. Tərcüməçidə paralel və ya vektor optimallaşdırmaları daxil edin;

2. Paralel hesablama üçün xüsusi proqramın növlərini, kitabxana hissə proqramlarını, xüsusi hazırlanmış kompüterləri konkret arxitekturalarda tətbiq etmək və onların bu arxitekturalar üçün optimallaşdırılmasını həyata keçirir;

3. Paralel kompilyasiyaların göstərişinə daxil olmaq kimi.

Paralel sistemin problemlərindən biri məlumatları yaddaşdan oxuyub yaddaşa yazmaqdır. Məsələn, əgər iki

processor paylaşılan yaddaşın tam eyni yerində məlumat yazmağa cəhd etsə, onda necə?

Əvvəllər gördüyümüz alqoritmlərdə bu alqoritmi hazırlayan aparat daha əvvəl verilən yaddaş cache-ə (RAM) birbaşa çıxışa malikdir. İndi baxdığımız alqoritmlər belə maşınların paralel variantı (PRAM) əsasında hazırlanıb. Bizim PRAM maşın processorlarımız bir-biri ilə əlaqəlidir və ümumi yaddaş blokundan istifadə edirlər. Hər bir processorun bir neçə registri var ki, onlar elə də böyük olmayan məlumatları saxlaya bilir, məlumatların böyük hissəsi isə ümumi yaddaşda saxlanılır.

Paralel Processorlar

Paralel processorlarla işləyərkən bizi iki konsepsiya maraqlandırır:

1. Sürət əmsalı
2. Dəyər (qiymət)

Paralel processor sürətləndirmə faktoru optimal serial processorun nə qədər sürətli işlədiyini göstərir. Məlumdur ki, optimal çeşidləmə alqoritmi $O(N \log N)$ əməliyyatlarını tələb edir. $O(N)$ mürəkkəbliyi ilə paralel çeşidləmə alqoritminin sürətlənmə əmsalı $O(\log N)$ -dir. Processorlar arasında sıx əlaqə ilə yanaşı, biz eyni zamanda onların hamısının yaddaşdan verilənləri oxumaq, onlar üzərində əməliyyat aparmaq və nəticəni yaddaşa yazmaq kimi eyni dövrü yerinə yetirdiyini fərz edirik. Bu o deməkdir ki, bütün processorlar eyni vaxtda yaddaşı oxuyur, oxunan məlumatları eyni vaxtda emal edir və eyni zamanda yazır.

Bu işin iki məqsədi alqoritmlərin proqramın effektivliyinə necə təsir etdiyini və müxtəlif alqoritmlərin təhlilini öyrənməkdir. Bəzi müasir proqramlara diqqət yetirdikdə, onların inkişaf etdiricilərindən bəziləri nə proqramın effektivliyinə, nə də yaddaşın səmərəli istifadəsinə diqqət yetirmirlər. Onların fikrincə, proqram çox yer tutarsa, istifadəçi əlavə yaddaş almaq və ya daha sürətli işləyən yeni bir kompüter almaq məcburiyyətində qalacaq. Amma kompüter sürəti sonsuz artan deyil. Bu, naqillı kabeldə elektronların hərəkət sürəti, optik kabellərdə işığın yayılma sürəti və hesablamada iştirak edən kompüterlər arasında rabitə kanallarının kommutasiya sürəti ilə məhdudlaşdırılır. Digər məhdudiyyətlər kompüterin imkanları ilə deyil, qarşıya qoyulmuş problemin mürəkkəblik dərəcəsi ilə bağlıdır. Elə problemlər var ki, hətta onları həll etmək üçün ən sürətli işləyən alqoritmlərdən istifadə edildikdə belə, insanın ömrü bunu etməyə kifayət deyil. Bu problemlər arasında çox vacib məsələlər də var ki, onların həlli üçün, yəni daha yaxın cavab almaq üçün alqoritmlər lazımdır. Bu praktik məşğələdə paralel proqramlaşdırmadan istifadə olunan əsas konsepsiyaları təqdim edəcəyik. Hər şeydən əvvəl, məlumatların emalının kompüter konsepsiyasını gətirmək ilə işə başlayaq. Sonra kompüter sistemlərinin arxitekturasına keçək. Sonda paralel alqoritmlərin analizində istifadə olunan prinsipləri təhlil edəcəyik.

Kompüter sistemlərinin kateqoriyaları. Kompüter sistemlərini 4 əsas kateqoriyaya bölmək olar. Bunun üçün onun necə işlədiyinə dair təlimatı bir az dəyişəcəyik. CPU

nöqtəyi-nəzərindən proqram deşifrə edilməli və yerinə yetirilməli olan qaydalar axınıdır. Məlumat həm də axın şəklində daxil olan hesab edilə bilər. Təhlil etdiyimiz dörd kateqoriya məlumat və qaydaların bir axına daxil olub-olmaması ilə müəyyən edilir. Bir qayda bir məlumat axınıdır. Tək qaydalı vahid məlumat axını (SISD) modeli bir prosessorlu klassik modeli təmsil edir. Bu, köhnə nəsil kompüterlərin, eləcə də bir çox müasir kompüterlərin nümunəsidir. Belə bir kompüter prosessoru istənilən vaxt yalnız bir qaydanı yerinə yetirməyə qadirdir və yalnız bir verilənlər toplusu ilə işləyə bilər. Belə ardıcıl sistemlərdə digər kateqoriyalardan fərqli olaraq paralellik yoxdur. Bir qayda çoxlu məlumat axınıdır. Digər qayda odur ki, çox yivli kompüterlərdə (SIMD) eyni əməliyyatı müxtəlif verilənlərlə idarə edən bir neçə prosessor var. SIMD maşınlarını bəzən vektor prosessorları adlandırırlar, çünki onlar vektorlar üzərində əməliyyatları yerinə yetirmək üçün çox əlverişlidirlər. Bu zaman hər prosessorla bir vektor koordinatı verilir və əməliyyatdan sonra nəticə vektoru alınır. Məsələn, vektorların əlavə edilməsi koordinatlardan istifadə etməklə həyata keçirilən əməliyyatdır. Vektorların cəminin birinci koordinatı birləşdirici vektorların birinci koordinatlarının cəmidir, ikinci koordinat ikinci koordinatların cəmidir və s.

Bizim SIMD maşın hər bir prosessorun daxil edildiyi iki vektor koordinatı haqqında qayda istifadə edir. Bu yeganə qaydanı yerinə yetirdikdən sonra nəticə başa çatmış hesab olunur. Qeyd edək ki, n elementlərindən ibarət olan bir vektorun həlli üçün SISD maşınında n iterasiya dövrü yerinə

yetirilməlidir, bir hərəkət n - dən az olmayan prosessorların sayı SIMD-maşın üçün kifayətdir. Bir neçə ayrı-ayrı məlumat axınlarının eyni vaxtda istifadəsi qaydası yalnız eyni məlumatları yerinə yetirərkən, əvvəlcə data ilə eyni hərəkətləri yerinə yetirərkən qəribə görünə bilər, çünki QN is tətbiqləri, nömrənin kvadrata qoyulması, iki dəfə çarpma, on hissəyə bölünmə kimi nadir hallarda baş verir. Ancaq bu vəziyyətə başqa bir baxımdan baxırıqsa, bu tip maşınlarda bir ədədin alt yoxlanılması və ya mürəkkəbliyini yaxşılaşdırma biləcəyimizi görürük. Əgər prosessorların sayı n -ə bərabədirsə, onda biz $N1$ və $N2$ arasındakı nömrələrin dərinliyi və ya mürəkkəbliyini MİSD – machine ilə bir əməliyyatda yoxlaya bilərik. X sayı kompleks olduqda, uyğun olmayan bir bölən olmalıdır.

Prosessorlar arasında sıx əlaqə ilə yanaşı, biz eyni zamanda onların hamısının yaddaşdan verilənləri oxumaq, onlar üzərində əməliyyat aparmaq və nəticəni yaddaşa yazmaq kimi eyni dövrü yerinə yetirdiyini fərz edirik. Bu o deməkdir ki, bütün prosessorlar eyni vaxtda yaddaşı oxuyur, oxunan məlumatları eyni vaxtda emal edir və məlumatları eyni vaxtda yazır. Yaddaş hüceyrələrinin mübahisəsi yalnız məlumatların oxunması zamanı deyil, həm də nəticənin yazılması zamanı yaranır. Üç addımlı dövrün şərtləri o deməkdir ki, əgər prosessor Y yaddaş hüceyrəsindəki məlumatları dəyişdirsə, X prosessorunun indicə oxuduğu məlumatları emal etməsi nəticəsində nə baş verdiyi bizi maraqlandırmır. Bundan əlavə, bir prosessor yaddaşdan məlumatları

oxuyarkən, digəri ona heç bir məlumat yazmağa çalışmayacaq.

Mübahisələrə yalnız yaddaşa rəqabətli və ya eksklüziv giriş icazəsi verilməklə icazə verilə bilər. Rəqabətli girişdə birdən çox prosessor eyni vaxtda tam bir yaddaş hüceyrəsinə daxil ola bilər. Xüsusi girişdə yalnız bir prosessor verilmiş yaddaş hüceyrəsinə eyni anda daxil ola bilər və eyni anda ikiyə daxil olmaq cəhdi səhv mesajı ilə nəticələnəcəkdir. Müsabiqəli qəbul təhsil zamanı problem yaratmır. Bizə həmçinin xüsusi oxumaq imkanı ilə işləyən alqoritmlər lazımdır. Xüsusi oxumaq imkanı olan bir neçə prosessor eyni vaxtda eyni yaddaş yerinə daxil olduqda xəta baş verir.

Yazı vaxtı xüsusi və rəqabətə davamlı girişlərin seçilməsində də problem yaranır. Xüsusi girişlə hər hansı yaddaş anbarına yazmaq hüququ yalnız bir prosessorla verilir. Bir neçə prosessor yazmağa cəhd etdikdə xəta baş verir. Lakin iki prosessor eyni vaxtda iki yaddaş anbarına yazıla bilər. Rəqabətə davamlı girişdə isə məsələ çox mürəkkəbdir, yəni yaranan münaqişələrə yol verə bilməkdir. Diploma malik modeldə hər bir prosessorla diplom verilir və yazmaq hüququ daha böyük səviyyəli prosessorla verilir.

Modelin sadələşdirilmiş prosessor səviyyəsi onun adına uyğun gəlir, yəni ad nə qədər kiçik olsa, səviyyə də bir o qədər çox olur. Əgər, məsələn, yaddaşa 4 və 7 prosessorları yazmağa cəhd edirsə, bu hüquq 4 adda prosessorla verilir.

Bir opsional olanı seçimli giriş modeli üzərində yarışan prosessorlardan əldə etmək olar. Sadə modeldə yalnız qeyd olunan məlumatların üst-üstə düşdükdən sonra yazılmasına

icazə verilir. Kombinasiya modelində yazılan məlumatlar üzərində sistem bir neçə hərəkəti yerinə yetirir. Məsələn, onların cəmi, sayı, ən böyük və ya ən kiçik elementi, yaxud məntiqi əməliyyatın nəticəsi (və, və ya) yazıla bilər. Müxtəlif vəziyyətlərdə bu imkanların hər biri faydalı ola bilər.

Nəticə olaraq yazı və oxu variantlarının 4 fərqli kombinasiyası var:

1. Rəqabətli oxuma / rəqabətli yazı (CRCW);
2. Rəqabətli oxuma / xüsusi yazı (CREW);
3. Xüsusi Oxu/Rəqabətli Yazı (ERCW);
4. Xüsusi oxu / Xüsusi yazı (EREW);

Sadə paralel əməliyyatlar. İndi isə iki elementar əməliyyatı görürük:

1. Daxil olan məlumatların prosessorlara ayrılması;
2. Siyahının ən böyük və ən kiçik elementinin müəyyən edilməsi;

Aşağıda biz i - prosessor P_i , prosessorların sayı p , N , j -yaddaş cache M_j ilə gələn siyahıda elementlərin sayını qeyd edirik. Yerinə yetirmək üçün paralel əməliyyatlar mötərizədə yazılır. Məsələn, Paralel Start və Paralel End. Əgər bir blok əməliyyatı başlanğıcda, ardınca ikinci paralel həyata keçirilsə, onlar ayrı-ayrı cüt mötərizəyə daxil edilir.

Suallar:

- 1.** Paralel hesablama sistemləri nəyə deyilir?
- 2.** Çox nüvəli prosessor nədir?
- 3.** Sistem arxitekturasına nələri aid etmək olar?
- 4.** M.D.Flinin əmərlər axınının növünə görə təsnifatı nədən ibarətdir?
- 5.** Proqramlaşdırma modeli nədir?
- 6.** SİSD (Single Instruction Single Data) nəyi təmsil edir?
- 7.** SİSD arxitekturası hansı prinsipə əsaslanır?
- 8.** SİMD (Single Instruction Multical Data) arxitekturası hansı prinsipə əsaslanır?
- 9.** MİSD (Multical Instruction Single Data) arxitekturası hansı prinsipə əsaslanır?
- 10.** MİMD (Multical Instruction Multical Data) arxitekturası hansı prinsipə əsaslanır?
- 11.** Paralel hərəkətlər prinsipi nələrdən ibarətdir?
- 12.** Proqramlaşdırma modeli nəyə xidmət edir?

Praktiki məşğələ № 2

Mövzu: Müasir super kompüterlər və onların növləri

İşin məqsəd: Müasir super kompüterlərin növlərini, modellərini, qurulma prinsiplərini öyrənmək.

Nəzəri hissə.

Superkompüter anlayışının tərifi

“Superkompüter” anlayışının tərifi dəfələrlə çoxsaylı mübahisələrin və müzakirələrin mövzusu olmuşdur.

Çox vaxt terminin müəllifi XX əsrin 60-cı illərinin sonlarında Livermor Milli Laboratoriyasında işləmiş Corc Maykl (George Anthony Michael) və Sidney Fernbach (Sidney Fernbach) və CDC şirkətinə (İng.) aid edilir. Lakin məlumdur ki, hələ 1920-ci ildə “New York World” qəzeti (ingiliscə) Kolumbiya Universitetinin sifarişi ilə yığılmış IBM tabulatorundan istifadə etməklə həyata keçirilən “superkompüter”dən bəhs edirdi⁴.

“Superkompüter” termini ümumi leksikona Seymour Cray-in CDC 6600, CDC 7600, Cray-1, Cray-2, Cray-3 və Cray-4 kimi kompüter sistemlərinin yayılması səbəbindən daxil oldu. Seymour Krey 1960-cı illərin ortalarından 1996-cı ilə qədər ABŞ hökuməti, sənaye və akademik elm və

⁴ <https://dic.academic.ru/dic.nsf/ruwiki/2049?ysclid=l8juwwf8re241614510>

texnologiya layihələri üçün faktiki olaraq əsas hesablama alətlərinə çevrilən hesablama maşınlarını inkişaf etdirdi. Təsadüfi deyil ki, o dövrdə superkompüterin məşhur təriflərindən biri belə idi: - “Seymur Kreyin yaratdığı istənilən kompüter”. Cray özü heç vaxt beyin övladlarını superkompüter adlandırmayıb, əvəzinə ümumi adı "kompüter"dən istifadə etməyi üstün tutub.



Şəkil 2.1. Lourens adına Livermor Milli Laboratoriyasında (LLNL), (Kaliforniya, ABŞ) quraşdırılmış Cray 1 superkompüter prosessorunun görünüşü.

Superkompüterlər böyük meynfreymlərdir, hansılar ki, yüksək sürətlə çox mürəkkəb riyazi əməliyyatların böyük miqdarını yerinə yetirməyə bacarırlar. Onlar riyazi

modelləşdirmə, hava proqnozu və görüntü emalında istifadə olunur. LLNL Cray, Octopus şəbəkəsi vasitəsilə Digər Böyük kompüterlər (Digər Cray maşınları və köhnə Control Data 7600 sistemləri) ilə əlaqələndirilir və nüvə texnologiyasında layihə hesablamaları üçün istifadə olunur. Cray-in kompüter sistemləri 1985-ci ildən 1990-cı ilə qədər 5 il ərzində bazarın zirvəsində idi. 1980-ci illər yüksək məhsuldar kompüterlərin yaradılmasında iştirak edən bir çox kiçik rəqabət aparıcı şirkətlərin meydana çıxması ilə səciyyələnirdi, lakin 90-cı illərin ortalarında onların əksəriyyəti bu fəaliyyət sahəsini tərk etdi və bu, hətta müşahidəçiləri "superkompüterin dağılması" haqqında danışmağa vadar etdi. Bu gün superkompüterlər IBM, Hewlett-Packard, NEC və digərləri kimi "ənənəvi" kompüter bazarının oyunçuları tərəfindən qurulmuş unikal sistemlərdir və bir çox ilk şirkətləri öz təcrübələri və texnologiyaları ilə əldə etmişlər. Cray hələ də superkompüter istehsalçıları arasında layiqli yer tutur.

«Superkompüter» anlayışı haqqında kifayət qədər aydın olmayan fikirlər hələ də geniş yayılmışdır. 1989-cu ildə hazırlanmış Qordon Bell və Don Nelsonun zarafatlı təsnifatı bir ton ağırlığında hər hansı bir kompüterin super kompüter hesab edilməsini təklif edirdi. Müasir superkompüterlər həqiqətən 1 tondan çoxdur, lakin hər ağır kompüter superkompüter sayılmağa layiq deyil. Ümumiyyətlə, super kompüter əksər istifadəçilər üçün mövcud olan maşınlardan daha güclü bir kompüterdir. Bununla yanaşı, bu gün texniki irəliləyişin sürəti belədir ki, hazırkı lider asanlıqla sabahki autsayderə çevrilə bilər.

Arxitektura⁵ superkompüterlər sinfinə mənsubiyyət əlaməti hesab edilə bilməz. Əvvəlki CDC kompüterləri sadəcə sürətli skalyar prosessorlarla təchiz olunmuş adi maşınlar idi. Bunlar digər şirkətlərin təklif etdiyi kompüterlərdən bir neçə on dəfə daha sürətli idi.

70-ci illərin superkompüterlərinin əksəriyyəti vektor prosessorları ilə təchiz olunurdu, 80-ci illərin əvvəllərində və ortalarında isə paralel işləyən vektor prosessorlarının sayı (4-dən 16-ya qədər) demək olar ki, standart superkompüter həlli olmuşdur. 80-ci illərin sonu və 90-cı illərin əvvəlləri superkompüterlərin inkişafının magistral istiqamətinin vektor-konveyer emalından paralel olaraq böyük və çox sayda skalyar prosessorlara dəyişməsi ilə xarakterizə olunurdu.

Kütləvi paralel sistemlər yüzlərlə və hətta minlərlə fərdi prosessor elementlərini birləşdirməyə başladı və onlar nəinki xüsusi hazırlanmış, həm də tanınmış və sərbəst mövcud prosessorlar ola bilərdilər. Əksər kütləvi paralel kompüterlər PowerPC və ya PA-RISC kimi güclü RISC prosessorları əsasında qurulmuşdur.

1990-cı illərin sonunda ixtisaslaşmış superkompüter həllərinin yüksək qiyməti və cəmiyyətin müxtəlif təbəqələrinin mövcud hesablama resurslarına artan tələbatı kompüter klasterlərinin geniş istifadəsinə səbəb oldu. Bu sistemlər serverlər və fərdi kompüterlər üçün ucuz və geniş yayılmış

⁵ *Kompüter arxitekturası hesablama sisteminin proqram təminatının və aparat resurslarının məntiqi təşkili və strukturudur. Bu, kompüterin əsas komponentlərinin təblərinin və həyata keçirilməsinin fundamental sxemi və funksional təsviridir. Arxitektura yaddaşın təşkilinə və onun necə ünvanlandığına əsaslanır.*

kompyuter komponentləri əsasında fərdi qovşaqların istifadəsi ilə xarakterizə olunur və güclü rabitə sistemləri və xüsusi aparat və proqram təminatı həlləri ilə birləşir. Görünən sadəliklərinə baxmayaraq, klasterlər tez bir zamanda super kompyuter bazarının kifayət qədər böyük seqmentini tutdular və həllərin ən aşağı qiymətinə ən yüksək performansı təmin etdilər.

Hazırda əsas vurğu mikroprosessorların sayının artırılmasına, paylanmış yaddaşa işləməyə, proqramların paralelizm dərəcəsinin artırılmasına, qrafik sürətləndiricilərin tətbiqinə, bulud xidmətləri ilə superkompyuter platformalarının yaradılmasına, həmçinin əlbəttə ki, yeni sistem-texniki həllərin axtarışına yönəldilir.

Superkompyuter arxitekturaları

Burada yüksək paralel çoxprosessorlu hesablama sistemlərinin müasir arxitekturalarından danışacağıq. Paralel məlumatların işlənməsinin əsas prinsipi bir neçə əməliyyatın eyni vaxtda yerinə yetirilməsini nəzərdə tutur. Bu halda, ənənəvi olaraq iki emal üsulunu ayırmaq adətdir: paralel və konveyer (mərhələ-mərhələ).

Paralel vektor sistemləri (PVP). PVP sistemlərinin əsas əlaməti konveyer funksional cihazlarında effektiv şəkildə yerinə yetirilən müstəqil məlumatların vektorlarının eyni tipli emal komandalarının təmin olunduğu xüsusi vektor-konveyer prosessorlarının olmasıdır. Bir qayda olaraq, bir neçə belə prosessor eyni zamanda çoxprosessorlu konfigurasiyalar

çərçivəsində ümumi yaddaş üzərində işləyir. Ya da belə bir neçə qovşaq komutator vasitəsilə birləşdirilə bilər. Bu cür arxitektura kompüterlərinin klassik nümunəsi Cray maşınlarının əksəriyyətidir.

Ümumi yaddaş olan simmetrik multi-processor sistemləri (SMP). Sistem bir neçə birhüceyrəli prosessordan və ümumi yaddaş massivindən (adətən bir neçə müstəqil blokdan) ibarətdir. Bütün prosessorlar eyni sürətlə yaddaşa bərabər hüquqludur. Prosessorlar ya ümumi şinlə (Front Side Bus) ya da kommunikatorla yaddaşa qoşulur. Qabların koherentliyi aparatla dəstəklənir. SMP sistemləri üçün nisbətən effektiv avtomatik paralelləşdirmə vasitələri mövcuddur.

Üstünlüklər: ümumi yaddaşın universal modelində proqramlaşdırma heç bir əlavə kod və ya əhəmiyyətli kod dəyişikliyi tələb etmir.

Bölüşdürülmüş yaddaşa malik massiv-paralel sistemlər (MPP). Sistem bir və ya bir neçə mərkəzi prosessor; yerli yaddaş (digər düyünlərin yaddaşına birbaşa giriş mümkün deyil); kommunikasiya prosessoru və ya şəbəkə adapteri; bəzən - sabit disklər və/və ya digər giriş/çıxış qurğuları. Əksər massiv-paralel sistemlər RISC arxitekturası (məhdud komanda dəsti ilə) ilə güclü prosessorlar əsasında yaradılır. Düyünlər kommunikasiya mühiti vasitəsilə bağlıdır.

Mesajların ötürülməsi (Message Passing Interface) modeli çərçivəsində proqramlaşdırma.

Üstünlüklər: çevik miqyaslanma, konfigurasiyanın optimallığı.

Yaddaşa müraciəti bərabər olmayan sistemlər (NUMA). Sistem az sayda prosessor və lokal yaddaş blokundan ibarət olan əsas modullardan ibarətdir. Modullar yüksək sürətli kommunikatorla birləşdirilmişdir. Vahid ünvan məkanı dəstəklənir, uzaq yaddaşa, yəni digər modulların yaddaşına daxil olmaq üçün aparat dəstəklənir. Eyni zamanda, yerli yaddaşa giriş uzaqdan olan yaddaşa nisbətən bir neçə dəfə sürətlidir.

Əgər bütün sistemdə koqerentlik aparat səviyyəsində saxlanılırsa, cc-NUMA arxitekturasından danışılır. Belə arxitekturaya malik maşınların nümunəsi superkompüterdir.

Klaster sistemləri. Klaster 2D və ya 3D şəbəkə ilə birləşdirilmiş qovşaqlar dəstidir. Hesablama qovşağı (computational node) ənənəvi və ya vektor-konveyer olan bir neçə prosessordan və onlar üçün ümumi yaddaşdan ibarətdir. Həmçinin düyünlərin daxili quruluşu eyni (homogen klasterlər) və ya fərqli (heterogen klasterlər) ola bilər. Adətən klasterin ən azı bir baş düyünü və fayl sistemi üçün ayrı-ayrı düyünləri vardır. Düyünlərin əlaqələndirilməsi üçün şin memarlığı və ya kommunikator əsasında standart şəbəkə texnologiyalarından (Fast/Gigabit Ethernet, Myrinet) biri istifadə olunur. Klaster sistemlərinin üstünlükləri - artan arızaya dözümlülük və daha səmərəli miqyaslanma. Klaster prinsipinə əsasən IBM-dən bəzi superkompüter modelləri inşa edilmişdir.

Doğrudur, müasir tapşırıqlar fərqli bir arxitektura tələb edir ki, onu da məlumat mərkəzli (data-centric) adlandırmaq

olar, yəni tək cə sayma sürətinə deyil, həm də böyük həcmli verilənlərlə işləməyin effektivliyinə diqqət yetirilir.

Sistemdə keş koherensiyası hardware səviyyəsində saxlanılırsa, cc-NUMA arxitekturasından danışılır. Bu arxitekturala malik maşınlar misal olaraq ASCI superkompüterini göstərmək olar.

Məhsuldarlıq

Superkompüterlərin məhsuldarlığı ən çox saniyədə (FLOPS) sürüşkən nöqtə ilə əməliyyatların sayında qiymətləndirilir və ifadə olunur. Bunun səbəbi superkompüterlərin yaratdıqları sayılı modelləşdirmə vəzifələrinin çox vaxt bütün ədədlərlə deyil, yüksək dəqiqlikli ədədlərlə əlaqəli hesablamalar tələb etməsidir. Buna görə də superkompüterlər üçün adi kompüter sistemlərinin sürət tədbiri tətbiq olunmur - saniyədə milyonlarla əməliyyat (MIPS). Bütün qeyri-müəyyənliyi və uyğunluğu ilə floplarda qiymətləndirmə superkompüter sistemlərini obyektiv kriteriyaya əsaslanaraq asanlıqla bir-biri ilə müqayisə etməyə imkan verir.

İlk superkompüterlər 1 flop sırasının performansına malik idi. Yəni, saniyədə 1000 sürüşkən nöqtə əməliyyatı. 1 milyon flop (1 Mflops) məhsuldarlığı olan CDC 6600 kompüterini 1964-cü ildə yaradılmışdır. 1 milyard flop (1 gigaflops) nişanı 1985-ci ildə Cray-2 superkompüterini tərəfindən böyük fərqlə (1,9 gigaflops) keçdi. 1 trilyon flop (1 Tflops) limitinə 1996-cı ildə ASCI Red superkompüterini çatdı. 2008-ci ildə

IBM Roadrunner superkompüterü tərəfindən 1 kvadrilyon flopun (1 Pflops) əlamətdar nöqtəsi çəkildi. 2016-cı ilə qədər saniyədə 1 kvintilyon sürüşkən nöqtə əməliyyatı apara bilən exascale kompüterlərin yaradılması üzrə işlər aparılır.

Superkompüterin proqram təminatı.

Ən çox yayılmış superkompüter proqram vasitələri, eləcə də paralel və ya paylanmış kompüter sistemləri MPI və PVM-ə əsaslanan tətbiq proqramlaşdırma interfeysləri (API) və Beowulf və openMosix kimi açıq mənbə həlləridir ki, bu da hətta virtual superkompüterlər yaratmağa imkan verir.

Adi iş stansiyaları və fərdi kompüterlər. Yeni hesablama qovşaqlarını yüksək ixtisaslaşmış klasterlərə tez bir zamanda qoşmaq üçün ZeroConf kimi texnologiyalardan istifadə olunur. Buna misal olaraq Apple tərəfindən paylanmış Shake proqramında renderin həyata keçirilməsini göstərmək olar. Shake proqramı ilə işləyən kompüterlərin resurslarını birləşdirmək üçün onları lokal şəbəkənin ümumi seqmentində yerləşdirmək kifayətdir.

Hal-hazırda superkompüterlər və geniş istifadə olunan proqram təminatı arasındakı sərhədlər çox bulanıqdır və paralelləşdirmə və çoxnövəli texnologiyaların fərdi kompüterlərin və iş stansiyalarının prosessor qurğularına daxil olması ilə birlikdə daha da bulanıqlaşır. Yalnız xüsusi növ kompüterlərin idarə edilməsi və monitorinqi üçün ixtisaslaşmış proqram vasitələrini, eləcə də "özünün" kompüter mərkəzlərində yaradılmış unikal proqram

mühtlərini, superkompüter sistemlərinin unikal konfigurasiyalarını bu gün müstəsna olaraq superkompüter proqramı adlandırmaq olar⁶.

Suallar.

1. Müasir super kompüterlər hansılardır?
2. Müasir super kompüterlər hansı prinsiplə qurulur?
3. Superkompüterlərin neçə emal üsulları var?
4. Paralel vektor sistemləri (PVP) nədir?
5. Hansı sistemlərə ümumi yaddaş olan simmetrik multi-processor sistemləri (SMP) adlandırmaq olar?
6. Bölüşdürülmüş yaddaşa malik massiv-paralel sistemlərə (MPP) nələrə aid etmək olar?
7. Hansılar yaddaşa müraciəti bərabər olmayan sistemlərdir?
8. Klaster sistemlər nəyə deyilir?
9. Superkompüterlərin məsuldarlığına nələr daxildir?
10. Hansı proqramları superkompüterlərin proqramı adlandırmaq olar?
11. 70-ci illərin super kompüterləri hansı processorlar ilə təchiz olunurdu?

⁶ <https://dic.academic.ru/dic.nsf/ruwiki/2049?ysclid=l8juwwf8re241614510>

Praktiki məşğələ № 3

Mövzu: Müasir kompüterlərin yaddaşın təşkilinə görə təsnifatı

İşin məqsədi: kompüter yaddaşının öyrənilməsi, onların növləri, yaddaşın təşkilinin təsnifatları.

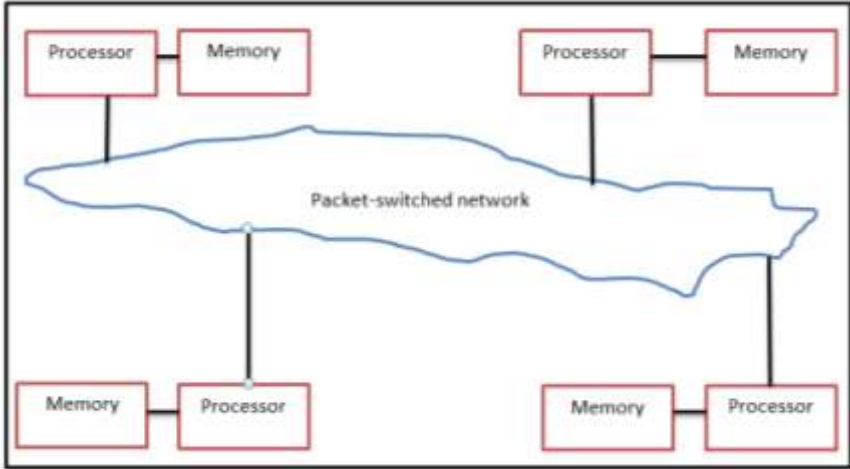
Nəzəri hissə

Paylaşılan və ayrılmış yaddaşa malik multiprocessorlar və multikompyuterlər. Ümumi və ayrılmış yaddaşa malik multisessorlar və multikompyuterlər proqram alqoritmlərində paralelləşmə metodlarının istifadəsini və onların çox nüvəli prosessorlarda praktiki həyata keçirilməsini, C++ -nin proqramlaşdırma dili kimi istifadəsini və kompüter direktivlərinin imkanlarını təsvir edirlər. Bundan əlavə, çoxnüvəli prosessorlarda proqramlarda istifadə olunan paralelləşmə alqoritmlərinin məhsuldarlıq səviyyəsini optimallaşdırmaq üçün, yaddaşdan oxuma və yazma əməliyyatlarını optimallaşdırmaq üçün görüntülərin həllində proqram vasitələrindən və axın bölmə metodlarından istifadə etmək imkanları yaradılır. Kompressiya problemləri.

Prosesorlarda çoxnüvəlilər anlayışının yaranması həm də proqram təminatında “çox oxu” anlayışının daha çox qeyd olunmasına səbəb olmuşdur. Bu iki anlayış indiki informasiya texnologiyalarında ayrılmaz şəkildə əlaqəli “cütdür”.

Multikompüterlər

Multikompüterlər paylanmış yaddaşa malik MIMD arxitekturasıdır. Aşağıdakı diaqram multikompüterin konseptual modelini göstərir.



Şəkil 3.1. Multikompüterin konseptual modeli.

Multikompüterlər məlumat mübadiləsi üçün paketlərin kommutasiya metodunu tətbiq edən mesajların ötürülməsi üçün maşınlardır. Burada hər bir prosessorun Şəxsi yaddaşı var, lakin qlobal ünvan sahəsi yoxdur, çünki prosessor yalnız öz yerli yaddaşına müraciət edə bilər. Beləliklə, kommunikasiya şəffaf deyil: burada proqramçılar kommunikasiya primitivlərini kodlarına açıq şəkildə yerləşdirməlidirlər.

Qlobal yaddaşın olmaması multikompyuterlərin çatışmazlığıdır. Bu, aşağıdakı iki sxem vasitəsilə həll edilə bilər:

Virtual ümumi yaddaş (VSM)

Ümumi virtual yaddaş (SVM)

Bu sxemlərdə tətbiqi programçı qlobal ünvanlanan böyük bir yaddaş nəzərdə tutur. Lazım gələrsə, tətbiqlərin yaratdığı yaddaş əlaqələri mesajların ötürülməsi paradigmasına çevrilir.

Virtual ümumi yaddaş (VSM)

VSM aparat reallaşdırmasıdır. Beləliklə, əməliyyat sisteminin virtual yaddaş sistemi şəffaf şəkildə VSM üstündə həyata keçirilmişdir. Beləliklə, əməliyyat sistemi hesab edir ki, o, ümumi yaddaşı olan maşınla işləyir.

Ümumi virtual yaddaş (SVM)

SVM, prosessorun yaddaş idarəetmə modulunun (MMU) aparat dəstəyi ilə əməliyyat sistemi səviyyəsində program həyata keçirilməsidir. Burada mübadilə vahidi - əməliyyat sisteminin yaddaş səhifələri.

Əgər prosessor müəyyən bir yaddaş xanasına müraciət edərsə, bu, yaddaşa daxil olmaqla bağlı olan yaddaş səhifəsinin yerli yaddaşda olub olmadığını müəyyən edir. Səhifə yaddaşda yoxdursa, adi kompüter sistemində diskdən əməliyyat sistemi yüklənir. Lakin SVM əməliyyat sistemi bu səhifəyə məxsus olan uzaq qovluqdan bir səhifə seçir.

İnformasiya mübadiləsi interfeysi texnologiyası

Paylanmış yaddaş paralel kompüterlərdə ümumi proqramlaşdırma texnologiyası MPI texnologiyasıdır. Belə sistemlərdə paralel proseslərin qarşılıqlı əlaqə üsulu bir-biri ilə mesaj mübadiləsidir. Bu metodun texnologiya adı Message Passing Interface kimi göstərilir. MPI standartı sistemin əməl etməli olduğu və proqram yaratarkən istifadəçilərin əməl etməli olduğu qaydaları ehtiva edir. MPI Fortran və Si dəstəkləyir. İnterfeys tam versiyasında 125-dən çox prosedur və funksiya var. MPI MIMD (Multiple Instruction Multiple Data) üslublu paralel proqramları dəstəkləyir. Müxtəlif mətnlərin verdiyi prosesləri özündə birləşdirir. Lakin bu cür proqramların hazırlanması və həyata keçirilməsi çətindir. Buna görə də, praktikada proqramçılar SPMD (SINGLE PROGRAM MULTIPLE DATA) paralel proqramlaşdırma modelindən istifadə edirlər. Paralel proseslər üçün vahid proqram kodundan istifadə edir. Hazırda MPI daha çox şəbəkə ilə işləməyi dəstəkləyir. Paralel proseslər üçün nəzərdə tutulmuş MPI kitabxanasının funksiyalarını dəstəkləmək üçün proqramı tərtib edərkən zəruri kitabxana modullarını əlaqələndirmək lazımdır. Ya onu əmr kimi əlaqələndirin, ya da əksər sistemlərdə mpicc (C proqramları üçün), mpicc (C++ proqramları üçün) və mpif77/mpif90 (Fortran77/90 proqramları üçün) xüsusi əmr və funksiyalar var. Kompilyatorun "-o adı" bölməsi a.out adlı işlək fayl yaradır, məsələn: mpiff77-o proqramı program.f

İş faylı yaradıldıqdan sonra lazımı sayda proseslər üçün işlədilməlidir. Bunun üçün adətən MPI proqramlarını işə salan mpirun əmri var, məsələn: mpirun - np N< proqram və arqumentlər>

Burada N bir məsələdəki proseslərin sayıdır. Başladıqdan sonra tək bir proqram işə salınan proseslər tərəfindən çağırılır. Zəngin nəticəsi sistemdən asılı olaraq terminala və ya göstərilən fayla yazılır. Bütün digər obyektlər: MPI-də müəyyən edilmiş prosedurlar, sabitlər və məlumat növləri MPI prefiksində malik olacaqdır. Əgər istifadəçi proqramda belə şəkilçili adlardan istifadə etmərsə, MPI obyektləri ilə bağlı heç bir problem olmayacaq. Bundan əlavə, Si dilində funksiya adları böyük hərflərə həssasdır. Adətən, MPI funksiyanın adında MPI şəkilçisindən sonrakı hərf böyük hərflə, sonrakı hərflər isə kiçik hərflərlə yazılır. MPI sabitlərinin adları tamamilə böyük hərflərlə yazılır. MPI interfeysinin təsviri mpif.h (mpi.h) faylında təcəssüm olunub. Bunun üçün MPI proqramına başqa 'mpif.h' direktivi daxil edilməlidir (Si proqramında #include 'mpi.h'). MPI proqramı bir-biri ilə əlaqə saxlayan paralel proqramlar toplusudur. Hər bir proses bir dəfə baş verir və proqramın paralel hissəsini təşkil edir. MPI proqramının icrası zamanı əlavə proseslər yaratmaq və ya mövcud olanları itirmək mümkün deyil. Hər bir proses öz ünvan məkanında yerləşir və MPI-də paylaşılan dəyişənlər yoxdur. Proseslər arasında əlaqə qurmağın yeganə yolu mesaj mübadiləsidir. Paralel proseslərin müstəqil əlaqəsi üçün bir qrup proseslər seçilir və onlar üçün ayrıca mühit kommunikatorları yaradılır. Proseslər yalnız bir

kommunikator daxilində əlaqə qurur və müxtəlif kommunikatorlara göndərilən mesajlar kəsişmir. Fortran dilində kommunikatorlar INTEGER sinfinə malikdirlər. Proqramın işə salınması zamanı baş verən proseslərin MPI_COMM_WORLD adlı tam əhatəli kommunikator daxilində işlədiyi güman edilir. Bu kommunikator həmişə mövcuddur və bütün işləyən MPI proqramları ilə əlaqə saxlamağa xidmət edir. Bundan əlavə, proqram başlayanda tək prosesli MPI_COMM_SELF kommunikatoru və tək prosesi olmayan MPI_COMM_NULL kommunikatoru olacaq. Proseslər arasında əlaqə müəyyən kommunikatorlar daxilində baş verir. Müxtəlif kommunikatorlara göndərilən mesajlar kəsişmir. MPI proqramında hər bir proses aid olduğu qrup daxilində unikal atributuna malikdir - müsbət tam proses nömrəsi. Bu atributun köməyi ilə proseslər arasında əsas əlaqə həyata keçirilir. Bir kommunikatordakı proseslər müxtəlif nömrələrə malikdir, lakin bir proses eyni vaxtda bir neçə kommunikatora aid ola bilər. Bu zaman bir kommunikatordakı prosesin sayı ikinci kommunikatordakı nömrədən fərqli ola bilər. Belə bir prosesin iki əsas atributunu başa düşmək olar: kommunikator və kommunikatordakı nömrə. Qrupda n proses varsa, cari qrupda ixtiyari proses nömrəsi 0-dan $n - 1$ -ə qədərdir. Proseslərin bir-biri ilə əlaqə qurmasının əsas yolu mesaj mübadiləsidir. Mesaj müəyyən bir kateqoriyaya aid məlumat toplusudur. Hər mesajın bir neçə atributu var. Məsələn, göndərən proses nömrəsi, qəbuledici proses nömrəsi, mesaj identifikatoru və s. Mesajın əsas atributlarından biri onun identifikatoru və ya etiketidir. Mesajı

qəbul edən proses eyni prosedəndən alınan iki mesajı mesaj identifikatoru ilə fərqləndirə bilər. Mesaj identifikatoru müsbət tam 0-dır və MPI_TAG_UP diapazonunda yerləşir və MPI_TAG_UP 32767-dən az deyil.

Mesaj atributlarını idarə etmək üçün massiv (Si-də struktur) təqdim olunur. Mesaj dəyərləri massiv elementləri ilə istinad edilir. Əksər MPI prosedurları son arqumentdə uğur haqqında məlumat qaytarır. Uğurla MPI_SUCCESS dəyəri qaytarılır, əks halda xəta kodu qaytarılır. Prosedurun icrası zamanı baş verən səhvi onun təsvirindən öyrənə bilərsiniz. Müxtəlif xəta kodlarına uyğun olan təsvirlər mpif.h faylında yerləşir.

Pentium mikroprosessoru 64 bitlik şin əsasında qurulub və yaddaşdan və registrlərdən məlumat mübadiləsini kəskin sürətdə sürətləndirir. Bir icra cihazı iki U və V cihazı ilə əvəz olunur. Onların hər biri paralel əməliyyatlar yerinə yetirir. O, konfigurasiyasına əsasən bütün əmrləri yerinə yetirir. Cihaz V yalnız ən ümumi əmrləri yerinə yetirən köməkçidir. Daxili registr (kesh) ikiyə bölünür: əmr keşi və məlumat keşi.

Pentium PRP kanalının daxili tezliyində işləyən daxili L2 keşyaddaş, 256, 512 və ya 1024 vahidlik nəzarətçi və yerli 64 bitlik məlumat şini əlavə edilmişdir. Əlavə daxili optimallaşdırma, sürətləndirilmiş boru kəməri və artan paralelləşdirmə səviyyəsi. Nisbətən güclü riyazi prosessor quraşdırılıb.

Multimedia sistemləri şəkillər üzərində müxtəlif hərəkətlərin ardıcılığını həyata keçirməkdə bir neçə mərhələdən keçir. Bunlara multimedia məlumatlarını

(məsələn, görüntünü) yaddaşdan oxumaq, icra səviyyəsindən asılı olaraq prosessoru növbəyə salmaq, operativ yaddaşdan görüntü məlumatlarını müvəqqəti saxlamaq üçün yer ayırmaq, daimi yaddaşa işləyərək əldə edilmiş görüntü dəyərlərini qeyd etmək daxildir.

Lakin bu hərəkətlər hər zaman konkret bir zaman ardıcılığı ilə bitmir və vaxt itkisinə gətirib çıxarır. Məsələn, bunun bir neçə səbəbini görə bilərik. Birincisi, ən böyük problemlərdən biri yaddaş problemləridir. Görüntü ölçüsünün eyni ölçüdə təyin olunmaması görüntünün hər bir bərpası və saxlanması üçün görüntüyə daxil olduqda yeni bir ölçü tələb edir. Bu da görüntü ilə əlaqəli dəyərlərin hesablanması, onun uzadılmasının müəyyən edilməsi, əməliyyat sistemi jurnalında bu dəyişikliklərin geri qaytarılması və vaxt itirməsi kimi çatışmazlıqlara gətirib çıxara bilər. İkincisi, emal edilmiş şəkil prosessoru yükləndikdə, görüntü məlumatları üzərində yerinə yetirilən hərəkətlərin ardıcılığı müəyyən məhdud vaxt ərzində başa çatdırılmalıdır. Əks halda, böyük ölçülü görüntü məlumatları yaddaş qeydi zamanı bəzi gecikmələrə səbəb ola bilər.

Bundan əlavə, gərgin tirlərin vaxtı məhduddur. Kompüter yaddaşında görüntü ölçüsünün artması işləmə sürətinin azalmasına gətirib çıxarır. Multimedia sistemləri mümkün qədər görüntünün işlənməsinin keyfiyyət səviyyəsini azaltmadan kiçik ölçülü saxlama alqoritmlərinin hazırlanmasını tələb edir. Çoxkonfessiyalı prosessorlar üçün nəzərdə tutulmuş paralel işləmə alqoritmlərinin görüntü prosesində istifadəsi kompüterin performansına müsbət təsir göstərir.

Lakin axın prosesində proessora yüklənmiş məlumatların həcmi görüntü məlumatlarının bir işlənməsində mühüm əhəmiyyət kəsb edir. Bu ona görədir ki, yaddaşdan müvəqqəti yer ayrılması məsələləri proqram məhsuldarlığını yaxşılaşdırmaq üçün həmişə həssas bir yerdir.

Yaddaşın bölüşdürülməsi məsələlərini həyata keçirən paralelləşdirmə mexanizmi aşağıdakı məsələləri həll etməlidir:

1. Axının təhlükəsizliyinin təmin edilməsi;
2. Əlavə xərclər;
3. Müsadirənin təşkil edilməsi;
4. X-oturuşlu nasazlıqlar.

İki axın eyni vaxtda yaddaşı məşğul saxlamaq istədikdə və ya boşaltmaq lazım gəldikdə, yaddaş ayırma mexanizminin daxili məlumat strukturunda qarışıqlıq başlayır və tətbiqin nasazlığına səbəb olur.

Problemin sadə həlli kimi yaddaşa girişi məhdudlaşdıran funksiyaların təşkil edilməsi və ondan yerin müvəqqəti ayrılması. Bu halda axınların yaddaşa daxil olması bir hərəkət zamanı yalnız bir dəfə tətbiq etməyə imkan verir. Nəticədə, bir proses icrası zamanı bütün axınlar bir dəfə yaddaşa daxil olur və bütün axın performansını sinxron şəkildə başa çatır.

Ayrılan yaddaşa giriş mexanizminə qadağaların tətbiq edilməsi aşağıdakı iki problemin yaranmasına səbəb ola bilər. Birincisi, yaddaşdan kosmosun ayrılması və ayrılmış yaddaşın buraxılmasına qoyulan qadağalar qurğu nəticəsində əlavə gözləntilərin ləngiməsi ilə nəticələnə bilər. İkincisi, yaddaşa giriş əldə etməkdə axınların hərəkəti qarışıq proseslərə gətirib

çıxara və proqramın işini ləngidə bilər. Şəkil 4-də göstərildiyi kimi axınların hər biri üçün yaddaşa lokal giriş təşkil etməklə hər iki problemi həll edə bilərik.

Hal-hazırda jpeg alqoritmi də daxil olmaqla, multimedia informasiya növü hesab edilən şəklin sıxılması üçün bir neçə alqoritm mövcuddur. Jpeg alqoritmi itkisiz və itkisiz sıxılma metoduna malikdir. İtki itirmə üsulu ilə sıxdıqda, görüntünün ölçüsü əhəmiyyətli dərəcədə azalmır. Görüntünün keyfiyyət səviyyəsinin vacib olmadığı hallarda bu metodun istifadəsi yaxşı nəticə vermir və multimedia sistemlərində istifadə edərkən yaddaş həcmindən daha çox yer alır. Bu problemi həll etmək üçün çoxmərtəli prosessorlar üçün paralelləşdirmə alqoritmlərinin köməyi ilə görüntü itkisi sıxılması üçün daha bir yeni alqoritm hazırlanmışdır.

Bu alqoritmə şəklin ikiölçülü Veyvlet - Dobeşi əvəzliyi işlənmişdir. Veyvlet - Dobeshi discrete modifikasiyası C# proqramlaşdırma dilinin paralelləşdirmə kitabxanaları istifadə edərək hərəkətlərin ardıcılığını görüntü sıxılmaya qədər sürətləndirirdi. Nəticədə rüşeymin qidalı maddələrdə inkişaf etməsinə icazə verildi, sonra isə onun mb-ına daxil edildi və orada implantasiya olundu. Bundan sonra Weivlet-Dobeshi, diskret dəyişiklikdən qaynaqlanan kütlə dəyərləri üzərində sıxılma hərəkətlərini paralelləşdirmək üçün alqoritmlərin köməyi ilə icra edildi.

Intel Paralel Gücləndirici tətbiqin çoxprosessorlu platformadan nə qədər səmərəli istifadə etdiyini aşkar edən və tətbiqin performansına təsir edən maneələri müəyyən edən performans profilidir.

Suallar.

- 1.** Paylaşılan və ayrılmış yaddaşa malik hansı prosessorlar və hansı kompüterlər var?
- 2.** Çoxnüvəli prosessorlar nədir?
- 3.** Multikomputerin konseptual modeli nələrəndən ibarətdir?
- 4.** Virtual ümumi yadda. (VSM) nədir?
- 5.** Ümumi virtual yaddaş (SVM) nədir?
- 6.** Yaddaşın bölüşdürülməsi məsələlərini həyata keçirən paralelləşdirmə mexanizmi hansı məsələləri həll etməlidir?
- 7.** Paylanmış yaddaş paralel kompüterlərdə ümumi proqramlaşdırma texnologiyası hansı texnologiyadır?
- 8.** MPİ (Message Passing Interface) hansı üslublu paralel proqramları dəstəkləyir?
- 9.** SPMD (Single Program Multiple Data) nədir və harda istifadə olunur?
- 10.** Çoxkonfessiyalı proses.sorlar nədir?

Praktiki məşğələ № 4

Mövzu: Müasir paralel kompüterlərin əsas sinifləri

İşin məqsəd: Paralel kompüterlərin strukturunu öyrənmək, praktiki klassifikasiya ilə tanış olmaq, paralel massiv sistemləri ilə tanış olmaq.

Nəzəri hissə

Paralel kompüterlərin təsnifatının əsas parametri ümumi (SMP) və ya paylanmış yaddaşın (MPP) mövcudluğudur. SMP və MPP arasında orta mövqe tutan NUMA arxitekturasıda mövcuddur, burada yaddaş fiziki cəhətdən paylanır, lakin məntiqi cəhətdən istifadəçilərə əlçatandır. Klaster sistemləri daha ucuz variantdır. Vektor məlumatlarının emalı komandalarının dəstəyi ilə vektor-konveyer prosessorlarından danışılır ki, onlar da öz növbəsində ümumi və ya paylanmış yaddaşdan istifadə edərək PVP sistemlərində birləşə bilirlər. Müxtəlif arxitekturanın eyni sistemdə birləşməsi və eynitip (eynincinsli) olmayan sistemlərin qurulması ideyaları getdikcə populyarlaşır.

Qlobal şəbəkələrdə (Internet) bölüşdürülmüş hesablamaların təşkili zamanı meta-kompüterlərdən danışılır ki, bu da ciddi şəkildə paralel arxitektura deyil.

Sadalanan bütün arxitekturaların xüsusiyyətlərinə daha ətraflı bu bölmədə, həmçinin bu siniflərin nümayəndələri olan

konkret kompüterlərin təsvirlərində baxılacaq. Hər sinif üçün aşağıdakı məlumatlar verilir⁷:

- arxitektura xüsusiyyətlərinin qısa təsviri,
- konkret kompüterlərin nümunələri,
- genişlənmə perspektivləri,
- əməliyyat sistemlərinin qurulmasının tipik xüsusiyyətləri,
- ən xarakterik proqramlaşdırma modeli.

Massiv-paralel emal sistemləri (MPP)

Kütləvi paralel emal sistemləri (massively parallel processing, MPP) onilliklər boyu mövcud olmuşdur⁸.

Fərdi satıcı arxitekturaları fərqli ola bilsə də, kütləvi paralel emal böyük miqdarda məlumatların saxlanması və təhlili üçün ən yetkin, sübut edilmiş və geniş istifadə olunan mexanizmdir. Beləliklə, kütləvi paralel arxitektura nədir və onun özəlliyi nədir?

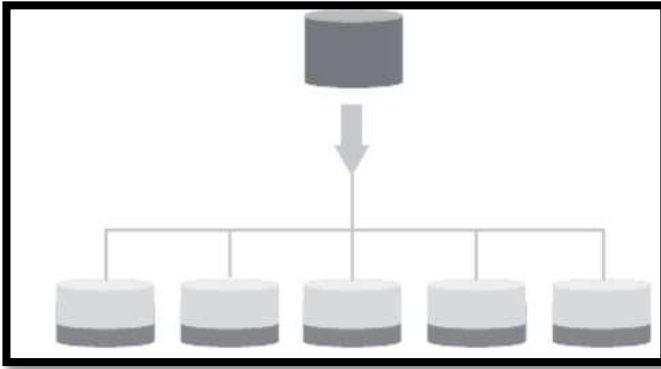
Kütləvi paralel arxitekturadan istifadə edərkən məlumatlar müstəqil mərkəzi emal bölmələri (CPU) tərəfindən işlənmiş və müxtəlif daşıyıcılarda saxlanılan fraqmentlərə bölünür. Bu, bir neçə şəbəkəyə qoşulmuş fərdi kompüterə müxtəlif məlumatların yüklənməsi kimidir. Bu, bir prosessor və bir disk ilə bir mərkəzi serverə malik olmaq məhdudiyyətini aradan qaldırır. Kütləvi paralel sistemdə məlumatlar

⁷ <https://parallel.ru/computers/classes.html>

⁸ <https://www.ckofr.com/rabota-v-internete/939-massivno-parallelnye-sistemy-obrabotki-mpp>

müxtəlif serverlərin prosessorları tərəfindən idarə olunan bir neçə disk üzərində paylanır (şəkil 4.1).

Belə arxitekturanın üstünlüyü nədir? Altı zolaqlı şosse boyunca hərəkəti təsəvvür edin. Əgər bu altı zolaq bir-birinə birləşərsə, hətta yolun qısa hissəsində belə, hərəkət çox çətin olacaq. Əgər altı zolaq başlanğıc nöqtəsindən təyinat yerinə qədər bütün yol boyu açıq qalırsa, səfər daha rahat olacaq. Ən yüksək saatlarda yolda tıxaclar baş verə bilər, lakin onlar daha az olacaq və tezliklə boşalacaq. Verilənlər bazasının ənənəvi arxitektura halında emal prosesində ən azı bir neçə nöqtə mövcuddur ki, zolaqların sayı birə qədər azalır. Yalnız hərəkət həcmi az olduqda bir zolaq kifayət edə bilər. Bu da arxitekturanın böyük həcmli məlumatların təhlili üçün əvəzolunmazdır: bütün zolaqların proses boyu açıq qalmasına imkan verir.



Şəkil 4.3. Massiv-paralel məlumat işləmə sistemi

Belə bir arxitekturanın üstünlüyü nədir? Altı zolaqlı avtomobil yolunda hərəkət etməyi təsəvvür edin. Əgər bu altı

zolaq bir zolaqda birləşərsə, hətta yolun qısa hissəsində olsa belə, hərəkət xeyli çətinləşəcək. Əgər altı zolaq başlanğıc nöqtəsindən təyinat yerinə qədər bütün yol boyunca açıq qalırsa, onda səfər daha rahat olacaq. Pik saatlarında yolda tıxaclar baş verə bilər, lakin onlar daha az olacaq və tezliklə dağılacaqlar. Ənənəvi verilənlər bazası arxitekturası halında emal prosesində ən azı bir neçə nöqtə mövcuddur ki, burada zolaqların sayı bir qədər azalır. Bir zolaq kifayət qədər ola bilər, yalnız əgər hərəkət həcmi kiçik. Bu, MPP arxitekturasını böyük həcmdə məlumatların təhlili üçün əvəzsiz edir: bu, bütün bantların bütün proses boyunca açıq qalmasına imkan verir.

Verilənlər bazası dünyasından bir nümunə nəzərdən keçirək. Ənənəvi verilənlər bazası terabayt cədvəlini bir sıra sorğulayacaq. Bununla belə, 10 prosessorlu kütləvi paralel sistemdən istifadə edərkən məlumatlar hər biri 100 gigabayt olan 10 müstəqil fragmentə bölünür. Bu o deməkdir ki, eyni anda 10 sorğu işləyir. Əgər sizə daha çox emal gücü və daha sürətli sürət lazımdırsa, sadəcə olaraq daha çox emal cihazı əlavə edin.

<p>Arxitektura</p>	<p>Sistem aşağıdakıları əhatə edən bircins hesablama qovşaqlarından ibarətdir:</p> <ul style="list-style-type: none"> • bir və ya bir neçə mərkəzi prosessor (adətən RISC), • Yerli yaddaş (digər qovşaqların yaddaşına birbaşa giriş mümkün deyil), • rabitə prosessoru və ya şəbəkə adapteri • bəzən - sabit (sərt) disklər və ya digər giriş/çıxış qurğuları <p>Sistemə xüsusi giriş-çıxış qovşaqları və nəzarət qovşaqları əlavə edilə bilər. Qovşaqlar bir sıra kommunikasiya mühiti (yüksək sürətli şəbəkə, keçid və s.) vasitəsilə əlaqələndirilir.</p>
<p>Nümunə</p>	<p>IBM RS/6000 <u>SP2</u>, Intel PARAGON/ASCI Red, CRAY <u>T3E</u>, Hitachi <u>SR8000</u>, Parsytec nəqliyyat (<u>transpüter</u>) sistemləri.</p>
<p>Miqyaslaşma</p>	<p>Real sistemlərdə prosessorların ümumi sayı bir neçə minə çatır (ASCI Red, Blue Mountain).</p>
<p>Əməliyyat sistemi</p>	<p>İki əsas variant var:</p> <ol style="list-style-type: none"> 1. Tam hüquqlu ƏS yalnız idarəetmə məşinində (front-end) işləyir, hər bir qovşaq ƏS-nin çox kəsilməmiş versiyasını idarə edir ki, bu da yalnız onun daxilində yerləşən paralel proqram əlavələrinin işini təmin edir. Məsələn: Cray T3E. 2. Hər qovşuqda tam hüquqlu UNIX oxşar OS (klaster yanaşmasına yaxın variant) işləyir. <p>Nümunə: IBM RS/6000 SP + OS AIX.</p>
<p>Proqramlaşdırma modeli</p>	<p>Məsajların ötürülməsi modelində proqramlaşdırma (<u>MPI</u>, <u>PVM</u>, <u>BSPLib</u>)</p>

Simmetrik çoxprocessorlu sistemlər (SMP)

Bu gün çoxaxınlı və paralel proqramlaşdırma təkcə elmi hesablamalar üçün deyil, həm də insan fəaliyyətinin gündəlik sahələrində tətbiq olunur. Bu mikroprocessor istehsalçılarının çoxnövəli arxitekturaya kütləvi keçidi ilə nəticələnir.

Simmetrik multi-processor sistemi (SMP, Symmetric Multiprocessor) aşağıdakı xüsusiyyətlərə malik hesablama sistemidir⁹:

- müqayisə edilə bilən performansla malik iki və ya daha çox processorun olması;

- processorlar əsas yaddaşdan birgə istifadə edirlər və vahid virtual və fiziki ünvan məkanında işləyirlər;

- bütün processorlar şin vasitəsilə və ya başqa bir sxemlə bir-birilə əlaqələndirilir, beləliklə, hər hansı biri üçün yaddaşa giriş vaxtı eynidir;

- bütün processorlar giriş/çıxış qurğularına ya eyni kanallar vasitəsilə, ya da eyni xarici qurğuya çıxışı təmin edən müxtəlif kanallar vasitəsilə girişi bölüşürlər;

- bütün processorlar eyni funksiyaları yerinə yetirməyə qadirdir;

- processorlardan hər hansı biri xarici fasilələrə xidmət edə bilər;

- hesablama sistem processorlar və proqramlar arasında tapşırıqlar, məsələlər, fayllar və verilənlər elementləri səviyyəsində qarşılıqlı əlaqəni təşkil edən və əlaqələndirən

⁹ <https://studfile.net/preview/1511376/page:5/>

inteqrasiya olunmuş əməliyyat sistemi tərəfindən idarə olunur¹⁰.

Mesajın və ya tam faylın informasiya mübadiləsinin fiziki vahidi rolunu oynadığı sərbəst bağlanmış çoxprocessorlu sistemlərdən (klasterlərdən) fərqli olaraq, SMP vahid məlumat elementi səviyyəsində qarşılıqlı əlaqəyə imkan verir. Bu, proseslər arasında yüksək əlaqə səviyyəsinə nail olur.

Texniki cəhətdən SMP sistemləri simmetrik olsa da, onların işində proqram təminatını daxil edən kiçik bir aşınma amili mövcuddur. Sistemin yüklənməsi zamanı processorlardan biri aparıcı (master) statusu alır. Bu o demək deyil ki, sonralar iş zamanı hansısa processorlar məlumatlı olacaq - onların hamısı SMP sistemində bərabərdir. Aparıcı termini hesablama sisteminin ilkin yüklənməsinə rəhbərlik edəcək processora aiddir.

Əməliyyat sistemi SMP arxitekturasının çoxprocessorlu təbiətini istifadəçidən gizlətməklə yanaşı, eyni anda bütün processorlar arasında prosesləri və ya texnoloji prosesləri planlaşdırır.

SMP-sistem yüksək sürətli sistem şinləri (SGI gücü, Sun Gigaplane, DEC TurboLaser) əsasında qurulur, slotlara funksional bloklar qoşulur: processorlar (CPU), giriş/çıxış altsistemi (I/O) və s. Giriş/çıxış (İ/O) modullarına qoşulmaq üçün daha yavaş şinlər (PCI, VME64) istifadə olunur.¹¹.

¹⁰ http://db4.sbras.ru/elbib/data/show_page.phtml?77+854

¹¹ <https://siblec.ru/telekommunikatsii/vychislitelnye-sistemy-seti-i-telekommunikatsii/5-arkhitektury-vysokoproizvoditelnykh-vychislitelnykh-sistem/5-2-simmetrichnaya-mnogoprotsessornaya-arkhitektura-smp?ysclid=I9g29u9ppt104549534>

Ən məşhur SMP sistemləri SMP serverləri və Intel prosessorları əsasında işləyən stansiyalardır (IBM, HP, Compaq, Dell, ALR, Unisys, DG, Fujitsu və s.). Bütün sistem vahid OS altında idarə olunur (adətən UNIX kimi, lakin Intel platformaları üçün Windows NT dəstəklənir). OS prosesləri prosessorlar üzrə avtomatik olaraq bölüşdürür¹².

SMP sistemlərinin əsas üstünlükləri¹³:

- proqramlaşdırma üçün sadə və universallıq. SMP memarlığı tətbiqetmənin yaradılmasında istifadə olunan proqramlaşdırma modelinə heç bir məhdudiyyət qoymur: adətən bütün prosessorlar bir-birindən asılı olmayaraq işlədikdə paralel budaqlar modelindən istifadə olunur. Lakin proseslərarası mübadilədən istifadə edən modelləri də həyata keçirmək olar. Ümumi yaddaşdan istifadə bu cür mübadilənin sürətini artırır, istifadəçi eyni zamanda dərhal bütün yaddaş həcminə daxil olur. SMP sistemləri üçün olduqca effektiv avtomatik paralelləşdirmə vasitələri mövcuddur;

- istismarın asanlığıdır. Bir qayda olaraq, SMP sistemləri hava soyutmasına əsaslanan kondisionerləşmə sistemindən istifadə edir ki, bu da onların texniki xidmətini asanlaşdırır;

- nisbətən aşağı qiymət.

Qüsurlar:

- ümumi yaddaş sistemləri zəif genişləndirilir.

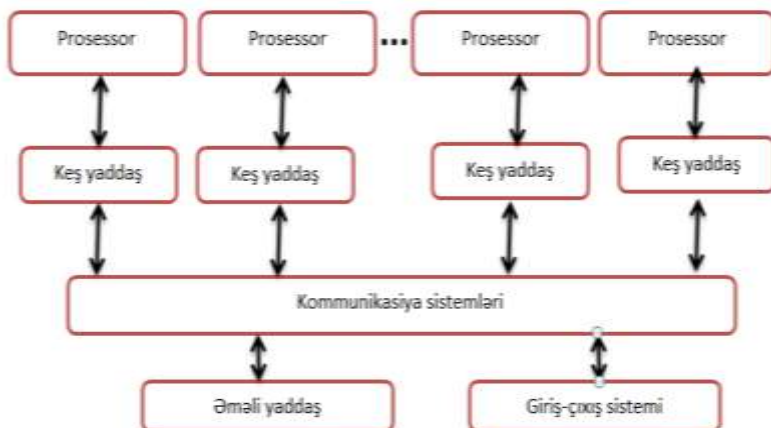
¹² <https://siblec.ru/telekommunikatsii/vychislitelnye-sistemy-seti-i-telekommunikatsii/5-arkhitektury-vysokoproizvoditelnykh-vychislitelnykh-sistem/5-2-simmetrichnaya-mnogoprotsessornaya-arkhitektura-smp?ysclid=l9g29u9ppt104549534>

¹³ <https://textarchive.ru/c-2917188.html>

SMP sistemlərinin bu əhəmiyyətli çatışmazlığı onları həqiqətən perspektivli hesab etməyə imkan vermir. Zəif miqyaslılığın səbəbi odur ki, şinlər hazırda yalnız bir tranzaksiyanı idarə edə bilər və bu, birdən çox prosessor paylaşılan fiziki yaddaşın eyni sahələrinə eyni vaxtda daxil olduqda münaqişənin həllində problemlər yaradır.

Hazırda 8-24 prosessor arasında münaqişələr yarana bilər. Bütün bunlar açıq şəkildə prosessorların və qoşulmuş istifadəçilərin sayının artması ilə performansın artmasına mane olur. Real sistemlərdə 32 prosessordan çox istifadə edə bilməzsiniz. SMP, klaster və ya NUMA arxitekturasına əsaslanan miqyaslına bilən sistemlər qurmaq üçün istifadə olunur. SMP sistemləri ilə işləyərkən ortaq yaddaş proqramlaşdırma paradigmasından istifadə olunur.

SMP-sistemin arxitekturası



Şəkil.1. Simmetrik multiprocessor sisteminin təşkili

Suallar.

- 1.** Müasir paralel kompüterlərin hansı əsas sinifləri vardır?
- 2.** NUMA arxitekturası nədir?
- 3.** Əməliyyat sistemlərinin qurulmasının hansı tipik xüsusiyyətləri vardır?
- 4.** Massiv-paralel emal sistemləri nədir?
- 5.** Kütləvi paralel arxitektura nədir və onun özəlliyi nədir?
- 6.** Sistem nələri əhatə edən bircins hesablama qovşaqlarından ibarətdir?
- 7.** Simmetrik multi-processor sisteminin hansı xüsusiyyətləri vardır?
- 8.** SMP sisteminin əsas üstünlükləri hansılardır?
- 9.** Hansı stansiyalar ən məşhur SMP sistemləri, SMP serverləri və Intel prosessorları əsasında işləyir?
- 10.** SMP sistemlərinin hansı mənfi xüsusiyyətləri vardır?

Praktiki məşğələ № 5

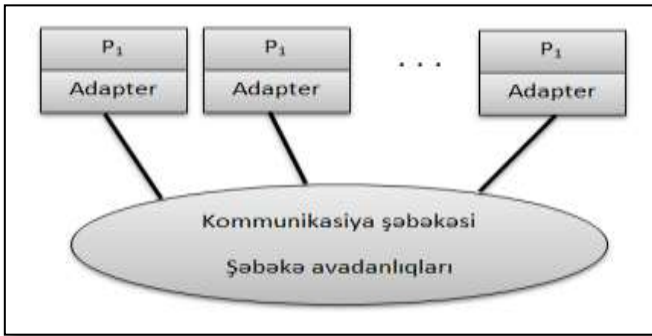
Mövzu: Kommunikasiya şəbəkələrinin topologiyası

İşin məqsəd: kommunikasiya, şəbəkə, şəbəkə üsulu, şəbəkə qurğuları, topologiya: şina, ulduz və halqa topologiyası.

Nəzəri hissə

Struktur olaraq rabitə mühiti aşağıdakı üç komponentdən ibarətdir (şək. 5.1.), burada P_i şəbəkə qovşağıdır (nodu) (prosessor və ya kompüter)¹⁴.

- şəbəkə adapterləri;
- rabitə şəbəkəsi;
- şəbəkə açarları.

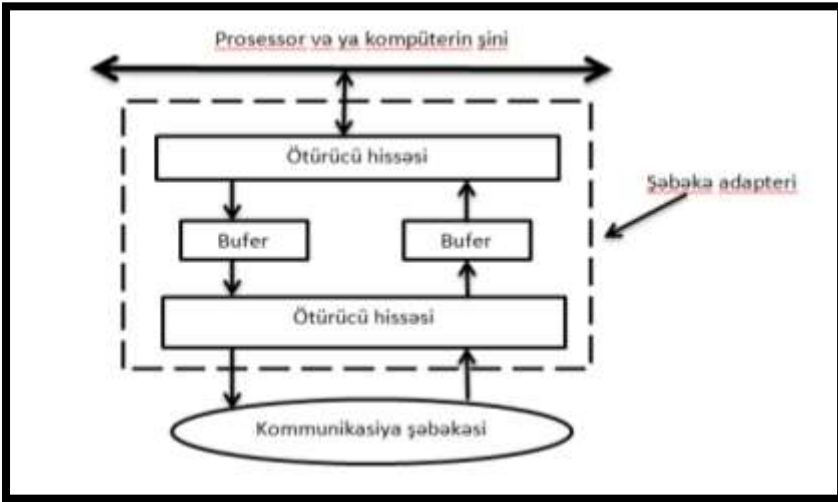


Şəkil. 5.1. Kommunikasiya mühitinin quruluşu.

¹⁴ <https://megaobuchalka.ru/1/74.html>

Şəbəkə adapterləri.

Ən yaxın qonşu olmayan şəbəkə qovşaqları arasında məlumat mübadiləsi zamanı məlumatlar ara qovşaqlar vasitəsilə ötürülür. Aydındır ki, qovşaqlarda qovşaq prosessorlarını məlumat yayımında iştirakdan azad edən bəzi aparat olmalıdır. Şəbəkə adapterlərinin əsas funksiyalarından biri şəbəkədə tərcümə məlumat mübadiləsini təmin etməkdir.



Şəkil 5.2. Şəbəkə adapterinin strukturu.

Qeyd edək ki, NUMA sistemləri keş koherentlik protokollarını dəstəkləyən yüksək sürətli şəbəkə adapterləri əsasında qurulur.

Əgər host prosessorunda eyni vaxtda bir neçə proqram prosesi işləyirsə, hostun şəbəkə adapteri sistem darboğazına

çevrilə bilər, çünki tətbiq prosesləri və əməliyyat sistemi prosesləri ona daxil olmaq üçün rəqabət apara bilər. Buna görə də, bəzən node prosessoru iki şəbəkə adapteri ilə təmin edilir: biri əməliyyat sistemi prosesləri üçün; tətbiq prosesləri üçün digərləri¹⁵.

Kommunikasiya şəbəkələri.

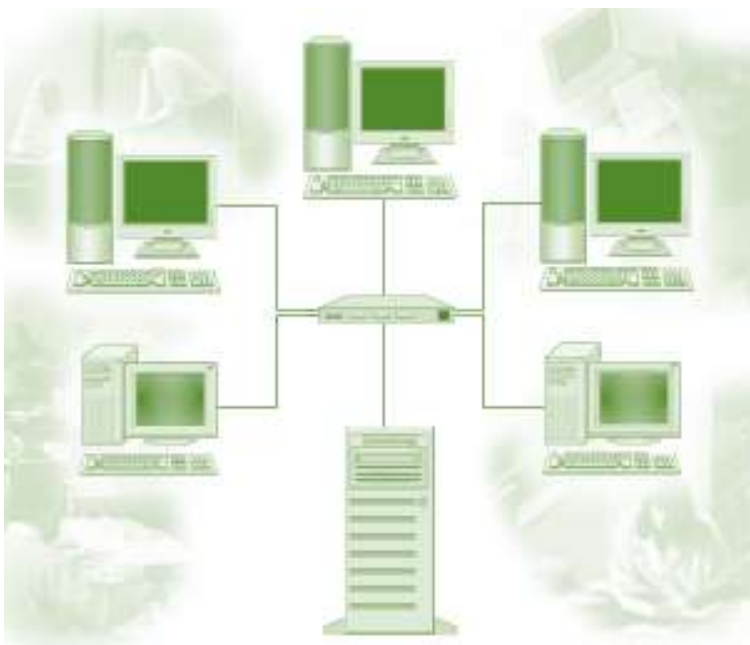
İstənilən kompüter şəbəkəsinin mühüm xüsusiyyəti onun əhatə etdiyi ərazinin genişliyidir. Əhatə dairəsinin genişliyi şəbəkəni təşkil edən kompüterlərin qarşılıqlı uzaqlığı ilə müəyyən edilir və buna görə də şəbəkənin qurulması zamanı seçilmiş texnoloji həllərə təsir göstərir. Klassik olaraq iki növ şəbəkə var: yerli şəbəkələr və qlobal şəbəkələr¹⁶.

Kommunikasiya şəbəkələri geniş miqyaslı kommunikasiya şəbəkələrinə (Wide Area Networks) - WAN şəbəkələrinə və lokal kommunikasiya şəbəkələrinə (Local Area Networks) - LAN şəbəkələrinə bölünür.

Lokal şəbəkələrə (Local Area Network, LAN) adətən kompüterləri nisbətən kiçik ərazilərdə (adətən 1-2 km radiusda) cəmləşən şəbəkələr daxildir.

¹⁵ <http://ks-211.blogspot.com/2015/06/lokal-sbklr.html>

¹⁶ <https://net.e-publish.ru/p216aa1.html>



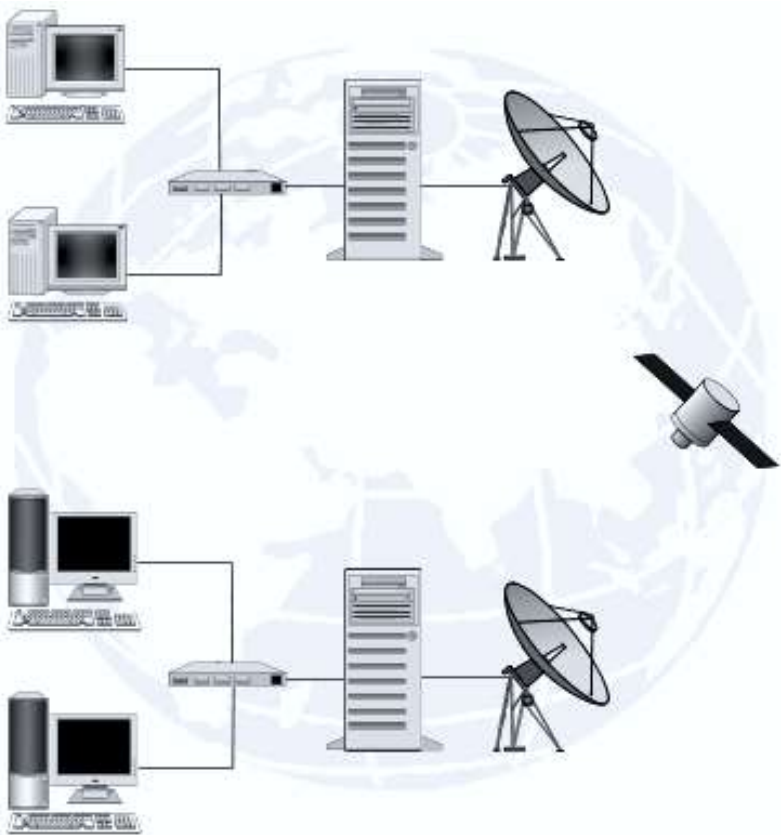
Şəkil 5.3. Lokal şəbəkə¹⁷

Lokal şəbəkələrin klassik nümunəsi bir və ya bir neçə bitişik binada yerləşən bir müəssisənin şəbəkəsidir. Lokal şəbəkələrin kiçik ölçüsü onların qurulması üçün kifayət qədər bahalı və yüksək keyfiyyətli texnologiyalardan istifadə etməyə imkan verir ki, bu da kompüterlər arasında məlumat mübadiləsinin yüksək sürətini təmin edir.

Geniş ərazi şəbəkələri (WAN) fərdi kompüterləri və bir-birindən xeyli məsafədə (yüzlərlə və minlərlə kilometr) yerləşən lokal şəbəkələri birləşdirmək üçün nəzərdə tutulmuş şəbəkələrdir. Xüsusi yüksək keyfiyyətli uzun məsafəli rabitə

¹⁷ <https://net.e-publish.ru/p216aa1.html>

kanallarının təşkili kifayət qədər bahalı olduğundan, qlobal şəbəkələr tez-tez kompüter şəbəkələrinin qurulması üçün nəzərdə tutulmayan mövcud xətlərdən istifadə edirlər (məsələn, telefon və ya teleqraf). Bu baxımdan, belə şəbəkələrdə məlumat ötürmə sürəti yerli şəbəkələrdən xeyli aşağıdır.



Şəkil 5.3. Qlobal şəbəkə¹⁸

¹⁸ <https://net.e-publish.ru/p216aa1.html>

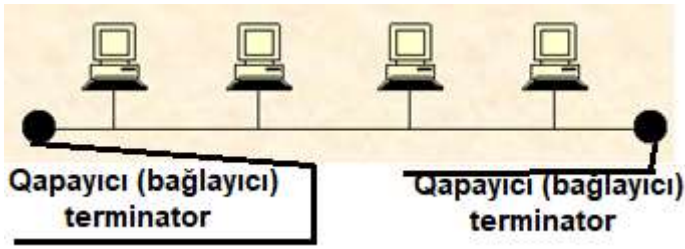
Kompüter şəbəkəsinin topologiyası (birləşmə, vizuallaşma, quruluş) dedikdə şəbəkə kompüterlərinin bir-birinə nisbətən fiziki yerləşməsi və onların rabitə xətləri ilə birləşməsi başa düşülür. Qeyd etmək vacibdir ki, topologiya anlayışı ilk növbədə əlaqələrin formalaşmasını görmək asan olan yerli şəbəkələrə aiddir. Qlobal şəbəkələrdə ünsiyyət strukturu ümumiyyətlə istifadəçilərdən gizlidir və böyük əhəmiyyət kəsb etmir, çünki hər bir rabitə seansı özəl şəkildə həyata keçirilə bilər. Topologiya avadanlıqlara olan tələbləri, məlumat ötürmə mühitinin tipini, mübadilənin idarə edilməsinin mümkün və ən rəşional üsullarını, işin etibarlılığını və şəbəkənin təkmilləşdirilməsi imkanlarını müəyyən edir. Topologiyanın seçimi şəbəkə istifadəçisi üçün çətin olsa da, əsas topologiyaların xüsusiyyətlərini, üstünlüklərini və çatışmazlıqlarını bilməlidir¹⁹.

Şəbəkənin üç əsas topologiyası mövcuddur:

- **Şin (Bus)** ²⁰. Bu halda bütün kompüterlər bir rabitə xəttinə paralel qoşulur və hər bir kompüterdən eyni anda qalan bütün kompüterlərə informasiya ötürülür (Şəkil 5.4.);

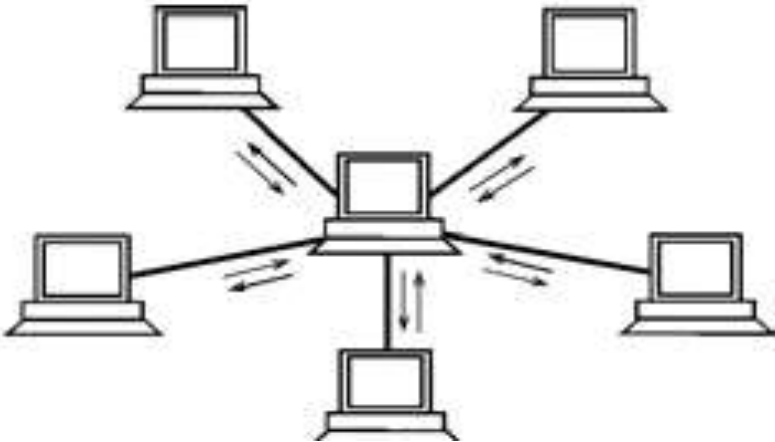
¹⁹ <https://lanfix.ru/clauses/typy-kompjuternyh-setej-i-sposoby-ih-upravlenija/>

²⁰ <https://intuit.ru/studies/courses/57/57/lecture/1672?page=3#image.1.5>



Şəkil 5.4. Şin topologiyası²¹

• **Ulduz** (star)—bir mərkəzi kompüterə digər periferik kompüterlər qoşulur və onların hər biri ayrıca rabitə xəttindən istifadə edir (şəkil 5.5.). Periferik kompüterdən məlumat yalnız mərkəzi kompüterə, Mərkəzi kompüterdən — bir və ya bir neçə periferik kompüterə ötürülür²²;



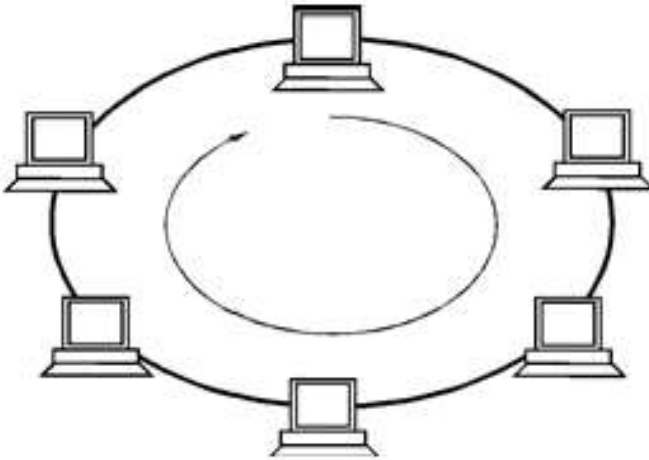
Şəkil 5.5. Ulduz topologiyası²³

²¹ <http://kom-seti.narod.ru/index.files/4.htm>

²² <https://sites.google.com/site/seticomptech/topologia-setej>

²³ <https://intuit.ru/studies/courses/57/57/lecture/1672?page=3#image.1.6>

- **Halqa** (ring) - kompüterlər ardıcıl olaraq halqa (dairəvi) formada birləşdirilir. Halqada informasiyanın ötürülməsi həmişə yalnız bir istiqamətdə aparılır. Kompüterlərin hər biri məlumatı zəncirdən sonrakı yalnız bir kompüterə ötürür və yalnız kompüter zəncirindəki əvvəlki kompüterdən məlumat alır (şəkil 5.6).



Şəkil 5.6. Halqa topologiyası

Praktikada yerli şəbəkələrdə digər topologiyalarından tez-tez istifadə olunur, lakin əksər şəbəkələr məhz üç əsas topologiyaya yönəldilmişdir.

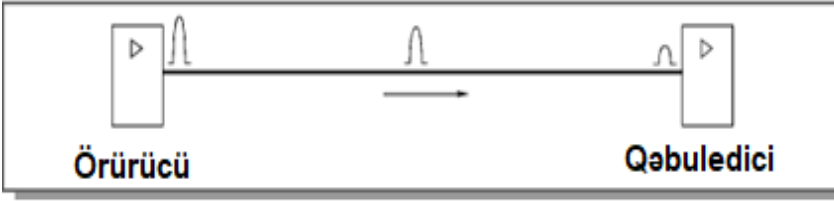
Əsas şəbəkə topologiyalarının xüsusiyyətlərinin təhlilinə keçməzdən əvvəl şəbəkənin fiziki göstəricilərinə təsir edən və topologiya anlayışı ilə bilavasitə əlaqəli olan bəzi mühüm amilləri qeyd etmək lazımdır.

Şəbəkəyə qoşulmuş kompüterlərin (abonentlərin) xidmət qabiliyyəti. Bəzi hallarda abunəçinin uğursuzluğu bütün şəbəkənin işini bloklaya bilər. Bəzən abunəçinin nasazlığı bütövlükdə şəbəkənin işinə təsir göstərmir, digər abunəçilərin məlumat mübadiləsinə mane olmur.

Şəbəkə avadanlıqlarının, yəni şəbəkəyə birbaşa qoşulmuş texniki vasitələrin (adapterlər, ötürücülər, birləşdiricilər və s.) istismara yararlılığı. Abunəçilərdən birinin şəbəkə avadanlığının sıradan çıxması bütün şəbəkəyə təsir göstərə bilər, lakin yalnız bir abunəçi ilə mübadiləni poza bilər.

Şəbəkə kabelinin bütövlüyü. Şəbəkə kabeli qırıldıqda (məsələn, mexaniki təsirlərə görə) bütün şəbəkədə və ya onun hissələrindən birində məlumat mübadiləsi pozula bilər. Elektrik kabelləri üçün kabledə qısaqapanma eyni dərəcədə vacibdir.

Kabel uzunluğunun məhdudlaşdırılması onun vasitəsilə yayılan siqnalın zəifləməsi ilə bağlıdır. Bildiyiniz kimi, hər hansı bir mühitdə yayılma zamanı siqnal zəifləyir (zəiflədir). Və siqnal getdiyi məsafə nə qədər böyükdürsə, bir o qədər zəifləyir (şəkil 1.7). Şəbəkə kabelinin uzunluğunun L_{pr} limit uzunluğundan çox olmamasını təmin etmək lazımdır, ondan yuxarı zəifləmə qəbul edilməz olur (qəbul edən abunəçi zəifləmiş siqnalı tanımır).



Şəkil 5.7. Şəbəkə üzərində yayılma zamanı siqnalın zəifləməsi

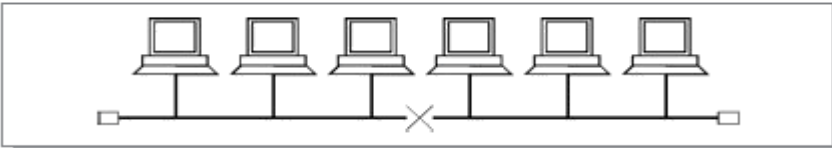
Şin topologiyası

Şin topologiyası (və ya hələ də deyildiyi kimi, ümumi şin) kompüterlərin şəbəkə avadanlıqlarının eyniliyini, habelə bütün abunəçilərin şəbəkəyə daxil olmasının bərabərliyini (eyni hüquqluluğunu) nəzərdə tutur. Şin topologiyasında kompüterlər yalnız növbə ilə məlumat ötürə bilər, çünki bu halda rabitə xətti (kabeli) yeganədir (bütün kompüterlər eyni kabelə qoşulur). Əgər bir neçə kompüter eyni vaxtda informasiya ötürərsə, o, tətbiq (münaqişə, kolliziya) nəticəsində təhrif olunur. Şində həmişə yarımdupleks (half duplex) mübadilə rejimi həyata keçirilir (hər iki istiqamətdə, lakin növbə ilə, eyni zamanda deyil).

Şin topologiyasında bütün məlumatların ötürüldüyü açıq-aşkar ifadə olunan mərkəzi abunəçi yoxdur, bu onun etibarlılığını artırır (çünki mərkəzi imtina etdikdə onun idarə etdiyi bütün sistem fəaliyyətini dayandırır). Şinə yeni abunəçilərin əlavə edilməsi olduqca sadədir və hətta şəbəkə işləyərkən də mümkündür. Əksər hallarda şindən istifadə

edərkən digər topologiyalarla müqayisədə minimum miqdarda birləşdirici kabel tələb olunur.

Mərkəzi abunəçi olmadığından bu halda mümkün münəqişələrin həlli hər bir ayrı abunəçinin şəbəkə avadanlıqlarına düşür. Bu baxımdan şin topologiyasında şəbəkə aparatı digər topologiyalara nisbətən daha mürəkkəbdir. Buna baxmayaraq, şin topologiyası olan şəbəkələrin (ilk növbədə ən populyar Ethernet şəbəkəsi) geniş yayılması səbəbindən şəbəkə avadanlıqlarının qiyməti çox yüksək deyil.



Şəkil 5.8. Şin topologiyası olan şəbəkədə kabel qırılması²⁴

Şinlərin əsas üstünlüyü ondadır ki, şəbəkə kompüterlərindən hər hansı biri imtina etdikdə, digər kompüterlər mübadiləni normal şəkildə davam etdirə biləcəklər.

Belə görünür ki, kabel qırılanda iki tam işlək təkər alınır (şəkil 5.8). Bununla belə, nəzərə almaq lazımdır ki, elektrik siqnallarının uzun rabitə xətləri boyunca yayılmasının xüsusiyyətlərinə görə, Şəkil 5.4.-də göstərilən xüsusi uyğunlaşdırıcı cihazların, terminatorların daxil edilməsini

²⁴ <https://intuit.ru/studies/courses/57/57/lecture/1672?page=4>

təmin etmək lazımdır. 5.4 və 5.8 şəkillərdəki düzbucaqlı kimi. Terminatorlar işə salınmadan siqnal xəttin sonundan əks olunur və təhrif edilir ki, şəbəkə üzərindən rabitə qeyri-mümkün olur. Kabelin qırılması və ya zədələnməsi halında rabitə xəttinin koordinasiyası pozulur və hətta bir-birinə bağlı qalan kompüterlər arasında mübadilə dayanır. Şin kabelinin istənilən nöqtəsində qısaqapanma bütün şəbəkəni sıradan çıxarır.

Hər hansı bir abunəçinin şində şəbəkə avadanlığının iflası bütün şəbəkəni sıradan çıxara bilər. Üstəlik, bütün abunəçilər paralel olaraq işə salındığı üçün belə bir imtinanı lokallaşdırmaq olduqca çətin və onlardan hansının sıradan çıxdığını başa düşmək mümkün deyil²⁵.

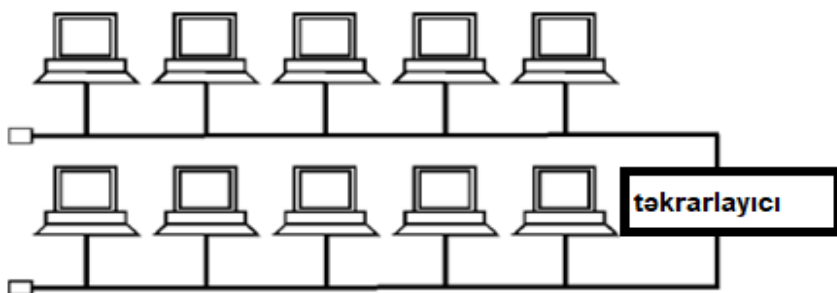
Şin topologiyası ilə rabitə xətti ilə keçərkən informasiya siqnalları zəifləyir və heç bir şəkildə bərpa olunmur ki, bu da rabitə xətlərinin ümumi uzunluğuna sət məhdudiyətlər qoyur. Üstəlik, hər bir abunəçi şəbəkədən ötürücü abunəçiyə qədər olan məsafədən asılı olaraq müxtəlif səviyyəli siqnallar ala bilər. Bu şəbəkə avadanlıqlarının qəbul qovşaqlarına əlavə tələblər qoyur.

Şəbəkə kabelindəki siqnalın L_{pr} uzunluğunda maksimum icazə verilən səviyyəyə qədər zəiflədiyini qəbul etsək, şinin tam uzunluğu L_{pr} ölçüsündən çox ola bilməz. Bu mənada şin digər əsas topologiyalarla müqayisədə ən kiçik uzunluğu təmin edir.

Şin topologiyası ilə şəbəkənin uzunluğunu artırmaq üçün tez-tez bir neçə seqmentdən istifadə olunur (şəbəkənin

²⁵ <http://sebekecomputer.blogspot.com/>

hissələri, hər biri şinidir), xüsusi gücləndiricilər və siqnalları bərpa edicilər - repiterlər və təkrarlayıcılar vasitəsilə bir-birinə birləşdirilir (şəkil 5.9, bu halda şəbəkənin maksimum uzunluğu $2 L_{pr}$ artır, çünki segmentlərin hər birinin uzunluğu L_{pr} ola bilər). Lakin şəbəkənin uzunluğunun belə artması sonsuz davam edə bilməz. Uzunluq məhdudiyyətləri siqnalların rabitə xətləri üzrə yayılmasının son sürəti ilə bağlıdır.



Şəkil 5.9. Təkrarlayıcıdan²⁶ istifadə edərək şin tipli şəbəkənin segmentlərinin birləşdirilməsi²⁷.

Ulduz topologiyası

Ulduz simli (kabel) və simsiz bağlantılar üçün ən çox yönlü ümumi topologiyadır. Bütün digər iş obyektlərinin bağlandığı aydın görünən bir mərkəzlə təmsil olunur. Məlumatın ötürülməsi prosesi yalnız əsas kompüter vasitəsilə

²⁶ Təkrarlayıcı şəbəkənin əhatə dairəsini genişləndirmək və siqnalın gücünü artırmaq, həmçinin "ölü zonaları" aradan qaldırmaq üçün nəzərdə tutulmuş siqnal gücləndiricisidir.

²⁷ <https://intuit.ru/studies/courses/57/57/lecture/1672?page=4>

baş verir, bunun nəticəsində o, böyük bir yük alır və bu, ona digər funksiyaları yerinə yetirmək imkanı vermir. Əlbəttə ki, şəbəkə cihazı mərkəzi abunəçi üçün daha uzaq (periferik) olanlara nisbətən daha mürəkkəbdir. Mərkəzi abunəçi mühüm funksiyanı - mübadilə idarəsini idarə edir. Bu topologiyada heç bir iş münaqişəsi baş verə bilməz, çünki mərkəzləşdirilmiş idarəetmə bunu təmin etmir²⁸.

İş prinsipi

Aparat stansiyası göndərməli olan məlumatları şəbəkə mərkəzinə göndərir. Qeyd etmək lazımdır ki, müəyyən bir anda yalnız bir işləyən maşın məlumatları yayımlaya bilər, əks halda hər iki paket alınmaya bilər və o, bu təsadüfi vaxt intervalını gözləməli və sonra məlumat ötürülməsini təkrarlamaq olacaq.

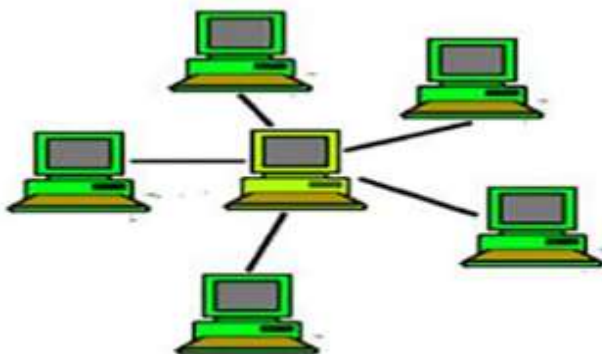
Şəbəkə kommunikatoru

Bu, belə qüsurları olmayan yüksək səviyyəli bir cihazdır. O, məlumatları bütün portlara ötürən keçiddən (kommutatorlardan) fərqli olaraq, paketi yalnız alıcıya göndərir. Kommunikatorun bu xüsusiyyəti kompüterin gücünü artırır və şəbəkə təhlükəsizliyini artırır. Eyni zamanda bir neçə məlumat paketini ötürmək mümkündür və onların sayı yalnız kommunikatordan asılıdır.

Aktiv ulduz

²⁸ Məmmədov C.F. "Kompüter şəbəkələri və sistemləri" fənnindən mühazirələr konspekti. Sumqayıt, 2006, -44 səh.

Bu topologiyada xüsusi server kompüteri şəbəkənin mərkəzində yerləşir²⁹.

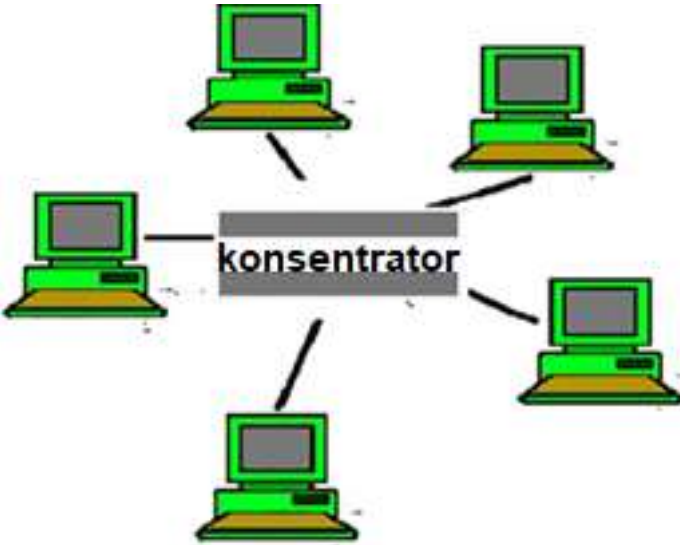


Şəkil 5.10. Aktiv ulduz topologiyası

Passiv ulduz

Şəbəkənin mərkəzində təkrarlayıcı rolunu oynayan bir hub və ya kommunikator var. Bu halda bütün şəbəkə abunəçiləri ümumi hüquqlara malikdirlər. Passiv ulduz aktiv (əsl) ulduzdan daha populyardır və Ethernet şəbəkəsində istifadə olunur.

²⁹ <https://plusiminusi.ru/plyusy-i-minusy-topologii-zvezda/>



Şəkil 5.11. Passiv ulduz topologiyası

Topologiya seçərkən mümkün üstünlükləri və mənfi cəhətləri nəzərə almaq lazımdır. Cihazların təcrid olunması rasionallıq baxımından əsas üstünlük olduğundan, müəyyən bir düyünün mərkəzə tabe olması narahatlıq doğurur³⁰.

"Ulduz" funksional vəzifələrlə bağlı ən sadə trafik növüdür. Lakin bu onun əsas üstünlüyü deyil. Bu şəbəkə topologiyasına xas olan aşağıdakı əsas üstünlükləri ayırd etmək olar:

-bu tip şəbəkələrin ümumi məhsuldarlığı yalnız (əsasən) mərkəzi qovşağın məhsuldarlığından asılı olduğundan yüksək sürətlidir;

³⁰ http://netaz.byethost9.com/sebeke_tesnifati.html?i=1

-verilənlər abonentlər və mərkəzi qovşaq arasında ayrı-ayrı kanallar vasitəsi ilə ötürüldüyündən şəbəkədə ötürülən verilənlər “toqquşmur”

-zədələnmiş şəbəkə seqmentinin tapılması və təmiri üçün sadələşdirilmiş olması. Çünki mərkəzi ofisdən birbaşa asılı olan bütün kompüterlərin tapılması əsas stansiyada bəzi əməliyyat nöqtələrinin işləməməsinin öhdəsindən gəlməyə kömək edir.

Çatışmazlıqları:

-ümmum şəbəkənin etibarlılığı mərkəzi qovşağın etibarlılığı ilə müəyyən edildiyindən bu tip şəbəkələrdə etibarlılıq aşağı olur. Əgər mərkəzi kompüter sıradan çıxarsa, onda bütün şəbəkənin işi dayanar;

- mərkəzi qovşaqdan yeni xətt çəkmək zərurəti olduğundan şəbəkəyə yeni qovşağın (abonentin) qoşulması baha başa gəlir³¹.

Halqa topologiyası

Halqa şəbəkəsi, hər bir qovşağın tam olaraq iki digər qovşaq ilə birləşdirildiyi, hər bir kompüter vasitəsilə siqnallar üçün vahid davamlı bir yol təşkil edən bir konfigurasiyadır - halqa. Məlumat qovşaqdan qovşağına keçir, hər biri yol boyu hər bir paketi emal edir³².

Halqa topologiyasının xüsusiyyətləri:

iş prinsipi

³¹ <http://e-derslik.edu.az/books/651/units/unit-1/page304.xhtml>

³² <https://plusiminusi.ru/osnovnye-plyusy-i-minusy-topologii-kolco/>

Ring topologiyası bütün şəbəkə kanallarının bir ayrılmaz zəncirə birbaşa qoşulmasını nəzərdə tutur. Bu, onun tipik bir dairə olduğu demək deyil. Halqa şəbəkəsinin mahiyyəti ondan ibarətdir ki, elektron məlumatların ötürülməsi üçün bir kompüter blokunun çıxışı və digərinin girişindən istifadə olunur. İnformasiyanın hərəkəti bir axınla baş verir. Çıxışda məlumat varsa və girişdə qəbul edilmirsə, o, girişə çatmaq üçün sonrakı cəhdlə yenidən çıxışa qaydır. Yəni məlumat həmişə eyni marşrutla göndərəndən alıcıya və əksinə hərəkət edir.

Məntiqi halqa bağlanmağa meyllidir. Halqa şəbəkəsinin əsas üstünlüyü onun qurulmasının çox asan olmasıdır. Ancaq gözlənilməz qəzalara qarşı etibarlı deyil. Dövrə bir qüsurlarsa, məlumat ötürmə halqası kəsilir. Çox vaxt praktikada İT mütəxəssisləri dəyişdirilmiş üzük topologiyasının layihələrini həyata keçirirlər³³.

Topologiyanın üstünlükləri

Asan quraşdırma. Halqa topologiyasını qurmaq olduqca asandır. Kompüterləri bir-birinə bağlamaq üçün heç bir server və ya mərkəzi iş stansiyası tələb olunmur. Bir cihazı digərinə qoşmaqla onları asanlıqla bir-birinə bağlamaq olar. Bu, ulduz və ya ağac topologiyalarından daha ucuzdur, hər ikisi qovşaqları idarə etmək üçün mərkəzi və ya master cihaz tələb edir.

Məlumat trafiki. Halqa topologiyası böyük miqdarda trafik idarə edə bilər, çünki məlumatlar bir istiqamətlidir. Bu,

³³ <http://computerologia.ru/topologii-seti-osnovnye-xarakteristiki/>

məlumat axınıni asanlaşdırır və şəbəkə sıxlığının qarşısını alır. O, həmçinin məlumatların korlanması şansını azaldır.

Problemlərin aradan qaldırılması. Bir səhv baş verdikdə, harada baş verdiyini müəyyənləşdirmək asandır, çünki ardıcıl məlumat ötürülməsi qovşaqların hansında əlaqənin pozulduğunu aydınlaşdırır³⁴.

Yük zamanı işin keyfiyyəti. Bütün sistemlər şəbəkə yükünə dözə bilmir. Məsələn, halqanı şinlə müqayisə etsək, birincisi daha yaxşı işləyəcək. Halqa topologiyası yükün artması şəraitində kifayət qədər sakit işləyə bilər.

Mərkəzi qovşaq (düyün) yoxdur. Bəzi sistemlər mərkəzi kompüterin olmasını nəzərdə tutan quruluşa malikdirlər, lakin bu cür mərkəzləşdirmə bəzən şəbəkəyə zərər verə bilər. Halqa topologiyasının əsas mərkəzi qovşağı yoxdur ki, bu da düyünlər arasındakı əlaqəni idarə etmək və nəzarət etmək üçün nəzərdə tutulub.

Ötürmə qabiliyyəti. Əlavə qovşaqların qoşulması şəbəkə ötürmə qabiliyyətinə çox az və ya heç bir təsir göstərmir.

Nizamlılıq. Bu topologiya deməkdir ki, şəbəkə çox nizamlı olacaq, burada hər cihaz tokenə və ötürmə imkanına malikdir.

Suallar.

1. Kommunikasiya mühitinin neçə komponenti var və hansılardır?
2. Şəbəkə adapterlərinin əsas funksiyaları nədir?

³⁴ https://studbooks.net/2029357/informatika/topologii_vychislitelnoy_seti

3. Node prosessorunu neçə şəbəkə adapteri təmin edir və hansılardır?
4. Klassik olaraq neçə növ şəbəkə vardır?
5. Lokal şəbəkələr (Local Area Networks) nədir?
6. Şəbəkənin neçə əsas topologiyası vardır?
7. Şin topologiyasının üstün və mənfi cəhətləri?
8. Hansı topologiyaya ulduz topologiyası deyilir?
9. Aktiv ulduz topologiyası ilə passiv ulduz topologiyasının fərqi nədir?
10. Halqa topologiyasının xüsusiyyətləri və iş prinsipləri nədir?

Praktiki məşğələ № 6

Mövzu: Paralel proqramlaşdırmanın tətbiqi sahələri

İşin məqsəd: paralel proqramlaşdırmanın əsas anlayışlarını və tətbiq sahələrini öyrənmək.

Nəzəri hissə

Bəşəriyyətin bir çox digər böyük ixtiraları kimi, kompüterlər də hərbi məqsədlər üçün icad edilmişdir. Əvvəlcə o, yalnız düşməndən gələn mesajların şifrəsini açmaq üçün istifadə edilən “sürətli hesablama”dan ibarət idi. Sonralar zaman keçdikcə onların hesablama gücü o qədər artdı ki, kompüterlərdən biznes və istehsalatda istifadə etmək mümkün oldu. O vaxtdan kompüter texnologiyasının təkamülü Mur qanununa tabe olmağa başladı: “kristaldakı tranzistorların sayı (eləcə də kompüterin səmərəliliyi) hər iki ildən bir iki dəfə artır”.

Son dövrlərdə Mur qanunu ilə təsdiq edilmiş hesablama gücünü reallaşdırmaq imkanı paralel hesablama sistemlərinin yaradılmasına səbəb olmuşdur. İllər ərzində prosessor istehsalçıları təlimat səviyyəsində paralellik və takt tezliyini davamlı olaraq artırdılar. Yəni yeni prosessorlarda proqram kodunda heç bir dəyişiklik edilmədən tək nüvəli proqramlar daha sürətli həyata keçirilirdi. Hazırda müxtəlif səbəblərə görə prosessor istehsalçıları çoxnüvəli arxitekturalara üstünlük verirlər. Özü də belə çoxnüvəli prosessorlar üçün müxtəlif

proqramlar hazırlanır. Lakin fundamental səbəblərə görə bu həmişə mümkün olmur.

Paralel proqramlaşdırma termini bir qədər geniş anlayışdır və çoxsaylı prosessorlardan ibarət hesablama sistemlərində hesablamaların təşkilinə aiddir. Belə sistemlərə çoxnüvəli prosessorlar, ortaq yaddaşa malik çoxnüvəli maşınlar, paylanmış yaddaş və ya hibrid arxitekturaya malik yüksək məhsuldar hesablama klasterləri daxildir.

Paralel proqramlaşdırma kompüterin hesablama problemlərini həll etmək üçün eyni vaxtda bir neçə resursdan istifadə etməsinə imkan verən proqramlaşdırma modelidir. Proqramların əvvəlki versiyaları ardıcıl prosesi izləyərkən, yəni onlar öz resurslarını yalnız bir problemin həllinə yönəldə bilərdilər, paralel proqramlaşdırma kompüterlərə eyni vaxtda bir neçə problemi həll etməyə imkan verir. Müasir kompüterlərin əksəriyyəti bu proqramlaşdırma növündən istifadə edir və müxtəlif sahələrdə geniş istifadə olunur³⁵.

Son zamanlar paralel hesablamalara çox diqqət yetirilir. Bu, iki əsas amillə bağlıdır. Birinci amil elmi-texniki proses ilə xarakterizə olunur. Nəticədə riyazi modelləşdirmə tələb edən yeni bilik sahəsi yarandı. Modellərin özləri də eyni zamanda daha mürəkkəb hala gəldi. Nəticədə, yalnız paralel metodlar və ya yüksək məhsuldarlıqlı texnologiya əsasında paylanmış hesablamalar vasitəsilə həyata keçirilən çoxlu sayda resursların hesablanması sahəsində tələbatın danılmaz artması müşahidə edilmişdir.

³⁵ <https://buom.ru/parallelnoe-programmirovaniye-opredeleniye-preimushhestva-i-primeneniye-v-otrasli/>

İkinci amil, yəni paralel hesablamalara marağ xeyli artdı ki, bu da paralel kompüterlərin bütün sahələrdə geniş yayılmasına səbəb oldu. Son zamanlar çoxprosessorlu serverlərə tez-tez orta və iri müəssisələrdə, banklarda, elmi-tədqiqat institutlarında və mərkəzlərdə rast gəlmək olur. Çox yivli prosessorların meydana çıxması ilə bir çox istifadəçi öz iş yerlərində mini-superkompüterlərdən istifadə etməyə başladı.

Şəbəkə texnologiyalarının əhəmiyyətli inkişafı təşkilatın yerli şəbəkəsini və paralel hesablamalar üçün kompüter otaqlarını birləşdirərək ucuz hesablama klasteri yaratmağa imkan verdi.

Nəhayət, əminliklə demək olar ki, paralel proqramlaşdırma (həmçinin paralel informasiya) texnologiyaları dar mühtdən müasir proqram təminatının yaradılması biliklərinin əsas kompleksinə çevrilmişdir.

Bir çox sənaye müxtəlif funksiyaları yerinə yetirmək üçün paralel proqramlaşdırmadan istifadə edir. Elm, mühəndislik, tədqiqat, sənaye, ticarət və pərakəndə satış da daxil olmaqla müxtəlif sənaye sahələri problemlərin həlli, məlumatların işlənməsi, modelləşdirmə və maliyyə proqnozlaşdırılması üçün paralel hesablama proqramlarını həyata keçirir. Sənaye istifadəsinə əlavə olaraq, bir çox fərdi kompüterlər axtarış motorlarını işə salmaq və ya video konfrans proqram təminatını yerləşdirmək kimi gündəlik funksiyaları dəstəkləmək üçün bu tip proqramlaşdırmadan da istifadə edirlər.

Real dünyada paralel emaldan istifadənin bəzi digər nümunələrinə aşağıdakılar daxildir³⁶:

- *Layihə (dizayn) işlərinə dəstək sistemləri (CAD - Computer Aided Design)*. Bu kateqoriyaya yeni model layihələrin hazırlanması daxildir. Məsələn, avtomobil nəqliyyatı sahəsində, binaların layihələndirilməsində, ümumiyyətlə, bütün layihələrin həyata keçirilməsində rahat və effektiv modelin əldə edilməsi məsələlərini həll edir.

- *Mühəndislik tətbiqləri*. Bu sinifə müxtəlif tapşırıqların kompakt modeli, fəvqəladə vəziyyətin modeli və s. daxildir.

- *Fiziki proseslərin riyazi modelləşdirilməsi*. Bu sinifə dinamika, mayelər və qazlar, elektromaqnit və nüvə qarşılıqlı təsirləri, yanma prosesləri və digər tapşırıqlar daxildir.

- *Yer planetində qlobal proseslərin modelləşdirilməsi*. İlk növbədə, bu, proqnozlaşdırma və hava dəyişikliyiini əhatə edir. Yeraltı proseslərdə və binalarda kommunikasiyaların təhlili, geoloji problemlərin həllində mürəkkəb hesablama işləri aparılır.

- *Kimya hesablamaları*. Bu proseslərdə molekulun dinamikası özünə münasibətdə geniş istiqamətləri əhatə edir. Bu kateqoriyada DNT-nin genişlənməsinə, zülalların konfigurasiyasına və bir çox kimyəvi hesablamada problemlərin həllinə kömək edir.

- *biznes proqramlarında*. Bu kateqoriyaya maliyyə bazarına nəzarət və valyuta məzənnələrinin proqnozlaş-

³⁶ Соснин В.В., Балакшин П.В. Введение в параллельные вычисления. – СПб: Университет ИТМО, 2015. – 51 с.

dırılması daxildir. Müəssisələrin rayonlarda yerləşdirilməsi və bir çox başqa vəzifələri yerinə yetirir.

Paralel proqramlaşdırmanın geniş yayılması onu müasir kompüterlərin getdikcə daha vacib xüsusiyyətinə çevirir³⁷.

Paralel proqramlaşdırmanın üstünlükləri

Bu proqramlaşdırma növünün əsas üstünlükləri bunlardır³⁸:

Səmərəlilik

Paralel proqramlaşdırmadan istifadə edən kompüter problemləri emal etmək və həll etmək üçün öz resurslarından daha yaxşı istifadə edə bilər. Müasir kompüterlərin əksəriyyətində çoxlu nüvələr, və ya prosessorları özündə birləşdirən avadanlıqlar var ki, bu da onlara eyni vaxtda bir çox prosesləri idarə etməyə və hesablama potensialını maksimum dərəcədə artırmağa imkan verir. Kompüterlər problemi həll etmək və ya məlumatı emal etmək üçün bütün resurslarından istifadə etdikdə, tapşırıqları daha səmərəli yerinə yetirirlər.

İqtisadi səmərəlilik

Bundan əlavə, paralel proqramlaşdırmaya imkan verən aparat arxitekturaları yalnız seriyalı emal etməyə imkan verən sistemlərdən daha sərfəli olur. Paralel proqramlaşdırma aparat

³⁷ Илья Федотов: Параллельное программирование. Модели и приемы. Солон-пресс, 2017 г. Страниц: 390. Подробнее:

<https://www.labyrinth.ru/books/581646/>

³⁸ <https://cyberleninka.ru/article/n/soderzhanie-kursa-osnovy-parallelnogo-programmirovaniya>

sistemi serial emal sistemindən daha çox təfərrüat tələb edə bilsə də, onlar tapşırıqları yerinə yetirməkdə daha səmərəlidir. Bu o deməkdir ki, onlar seriyalı proqramlardan daha az vaxt ərzində daha çox nəticə verir və zamanla daha çox maliyyə dəyərinə malikdirlər.

Sürət

Paralel hesablamanın digər üstünlüyü onun mürəkkəb məsələləri həll etmək qabiliyyətidir. Paralel proqramlar mürəkkəb problemləri daha kiçik tapşırıqlara bölə və eyni zamanda həmin fərdi tapşırıqları emal edə bilər. Böyük hesablama işlərini daha kiçik tapşırıqlara bölmək və onları eyni vaxtda emal etməklə, paralel emal kompüterlərin daha sürətli işləməsinə imkan verir.

Paralel emal məhdudiyyətləri

Paralel emal bir çox üstünlüklərə malik olsa da, məhdudiyyətlərə də malikdir. Bu məhdudiyyətlərin bəziləri aşağıdakılardır³⁹:

Kodlaşdırma tələbləri: Paralel emal kodlarının yazılmasını öyrənmək proqramçılar üçün daha çox çətinlik yarada bilər, lakin bu kompüterlərin kodlaşdırılmasının mürəkkəbliyi proqramçılar üçün maraqlı problem ola bilər və mürəkkəb emal sistemlərinə gətirib çıxara bilər.

Baxım ehtiyacları: Paralel emal sistemlərinin kodlaşdırılması işinin keyfiyyətini qorumaq üçün daha tez-tez

³⁹ <https://buom.ru/parallelnoe-programmirovanie-opredelenie-preimushhestva-i-primeneniye-v-otrasli/>

yeniləmələr və düzəlişlər tələb edə bilər. Bununla belə, bu kompüterlərin bir çoxunun xüsusi xüsusiyyətlərinə görə, texniki xidmət tələbləri çox vaxt onlardan istifadənin faydalarından daha az əhəmiyyət kəsb edir.

Mürəkkəbliik: Bəzi hallarda serial sistemlər daha yaxşı işləyə bilər, çünki onlar prosessorlar arasında daha az əlaqə və koordinasiya tələb edir. Paralel emal sisteminin mürəkkəbliyi mürəkkəb məsələlərin həlli üçün effektivdir, sadə sistemlər isə sadə məsələlərin həlli üçün adətən kifayətdir.

Paralel emala yanaşmalar

Paralel emalları dəstəkləyən dörd müxtəlif kompüter arxitekturası var. Kompüter alimləri bu modelləri iki amili necə həyata keçirdikləri əsasında müəyyənləşdirirlər: təlimat axını və məlumat axını. Təlimat axını proqramların problemləri həll etmək üçün istifadə etdiyi təlimatlar ardıcılığı olan bir alqoritmdir. Məlumat axını kompüterin yaddaşından götürdüyü məlumatdır. Kompüterlər məlumat axınından məlumatları emal etmək və tapşırıqları yerinə yetirmək üçün təlimat axını tərəfindən təmin edilən alqoritmlərdən istifadə edirlər.

Suallar:

1. Paralel proqramlaşdırma nəyə deyilir?
2. Paralel proqramlaşdırmadan hansı sahələrdə istifadə edirlər?
3. Paralel proqramlaşdırma modeli nədən ibarətdir?
4. Hansı prosessorlara çox yivli prosessorlar deyilir?
5. Paralel proqramlaşdırmanın nə kimi üstünlükləri var?
6. Paralel proqramlaşdırmanın iqtisadi səmərəliliyi nədən ibarətdir?
7. Fiziki proseslərin riyazi modelləşdirilməsinə nələr daxildir?
8. Hansı sistemə çoxnüvəli prosessorlar, ortaq yaddaşa malik çoxnüvəli maşınlar, paylanmış yaddaş və ya hibrid arxitekturalara malik çoxnüvəli yüksək məhsuldar hesablama klasterləri daxildir?
9. Mur qanunu ilə təsdiq hesablama gücünü reallaşdırmaq imkanı hansı sistemlərin yaradılmasına səbəb olmuşdur?
10. Dünyada paralel emalın istifadəsinə nələr daxildir?

Praktiki məşğələ № 7

Mövzu: Open MP proqram alətində paralel proqramlaşdırma

İşin məqsəd: OpenMP mühiti, əsas pəncərə, komanda sətiri ilə tanış olmaq, interfeysin öyrənilməsi, yaradılış tarixini bilmək.

Nəzəri hissə

OpenMP (Open Multi - Processing) əsasən paylaşılan yaddaşa malik paralel hesablama sistemləri üçün hazırlanmış çox axınlı proqramlar yaratmaq üçün nəzərdə tutulmuş praktik proqram interfeysidir. OpenMP Tərtibçilər üçün bir sıra direktivlərdən və xüsusi funksiyalar kitabxanasından ibarətdir. OpenMP standartı növbəti 15 il ərzində yaradıldı və paylaşılan yaddaş arxitekturalarına tətbiq edildi. Son illərdə paylanmış yaddaş paralel hesablama sistemləri üçün OpenMP standartı genişləndirilmiş formada hazırlanmışdır. 2005-ci ilin sonunda Intel, Genişləndirilmiş OpenMP-nin inkişaf etmiş paralel paylanmış yaddaş hesablama sistemləri üçün hazırlandığı Cluster OpenMP məhsulunu təqdim etdi⁴⁰.

Bir neçə iri tərtibatçı şirkətlər (Intel, Hewlett-Packard, Silicon Graphics, Sun, IBM, Fujitsu, Hitachi, Siemens, Bull) OpenMP Architecture Review Board adlı qeyri-kommersiya təşkilatı tərəfindən idarə olunan hesablama və proqram

⁴⁰ <https://habr.com/ru/company/intel/blog/82486/>

texnologiyaları üçün OpenMP spesifikasiyaları yaradırlar. (ARB).

OpenMP Fortran və C/C++ alqoritmik dillərində çox axınlı tətbiqlərin sürətli və asan yaradılmasını həyata keçirir. OpenMP ilk versiyası 1997-ci ildə Fortran dili üçün yaradılmışdır. C/C++ proqramlaşdırma dilləri üçün isə 1998-ci ildə yaradılmışdır. 2008-ci ildə isə OpenMP 3.0 versiyası təqdim edilmişdir⁴¹.

Hal-hazırda, bir çox təchizatçıların kompilyatorlarında OpenMP standartının tətbiqləri mövcuddur (bax: <http://www.openmp.org/drupal/node/view/9>).

OpenMP-nin üstünlükləri bunlardır:

-*İstifadə rahatlığı*-tərtibatçı (geliştirici, proqramçı) yeni bir paralel proqram yaratmır, əksinə ardıcıl proqramın mətninə lazımi direktivləri və bəlkə də tərtibçiyə hesablama və məlumatları axınlar arasında bölüşdürmə yollarını göstərən kitabxana funksiyası çağırışlarını əlavə edir. OpenMP-nin əsas ideyası, ümumiyyətlə əsas hesablama yükünü daşıyan dövrlərin paralelləşdirilməsidir.

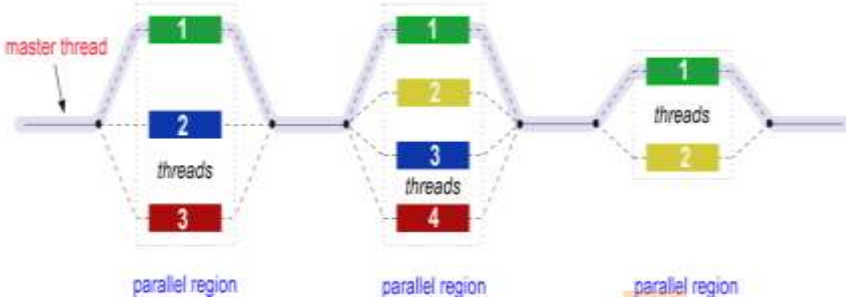
-*Çeviklik* - OpenMP tərtibatçıya paralel proqramın davranışı üzərində kifayət qədər çox nəzarət təmin edir.

⁴¹ Сысоев А.В., Мееров И.Б., Свистунов А.Н., Курылев А.Л., Сенин А.В., Шишков А.В., Корняков К.В., Сиднев А.А. «Параллельное программирование в системах с общей памятью. Инструментальная поддержка». Учебно-методические материалы по программе повышения квалификации «Технологии высокопроизводительных вычислений для обеспечения учебного процесса и научных исследований». Нижний Новгород, 2007, 110 с.

-*Təkrar istifadə* - OpenMP proqramı bir prosessor platformasında işləməsini təmin etmək lazımdırsa, bir çox hallarda müntəzəm serial proqram kimi təkrar istifadə edilə bilər. Eyni zamanda, icra olunan modulda OpenMP-nin tətbiqindən xilas olmaq üçün onu serial kompilyatorla yenidən qurmaq kifayətdir. OpenMP direktivləri nəzərə alınmayacaq və kitabxana funksiyası çağırışları mətni standart spesifikasiyalarda verilmiş stublarla əvəz edilə bilər.

OpenMP proqramlaşdırma modeli⁴².

OpenMP vasitəsilə tərtib etmək istədiyiniz proqram şəkildə təsvir olunduğu kimi (şəkil computing.llnl.gov saytıdan götürülmüşdür) kodun ardıcıl və paralel hissələrindən (bölgələrindən) ibarət olmalıdır:



Bu şəkildə soldan sağa proqram icrasının gedişi göstərilir. Bir üfüqi xəttin çəkildiyi yerdə proqram ardıcıl hərəkətlər edir. Sütunda bir neçə üfüqi xətt («parallel region» yazılıb) göstərildiyi yerdə proqramda bir neçə axın yerinə yetirilir. Paralel bölgənin son nöqtəsində bütün iş salınmış

⁴² <http://ccfit.nsu.ru/arom/data/openmp>

axınların icrası gözlənilir və yalnız bundan sonra paralel bölgənin başa çatması gözlənilir. Paralel bölgə başa çatdıqdan sonra proqramın ardıcıl rejimdə icrası davam etdirilir. Paralel bölgədəki hər bir ifadə budağı öz içərisində eyni şəkildə qurula bilər (yəni paralel bölgənin daxilində paralel bölgələr var). «Master thread» yazısına diqqət yetirmək olmaz, çünki bu vəsaitdə biz bu cür incəlikləri təsvir etməyəcəyik⁴³.

Əsas axın zəruri hallarda törəmə axınları yaradır.

Fork-join modeli.

Proqramın mənbə kodunun əsas yerlərinə kompilyator direktivlərini daxil etməklə proqramlaşdırma. Kompilyator bu direktivləri şərh edir və kodun bölmələrini paralelləşdirmək üçün proqramın müvafiq yerlərinə kitabxana çağırışlarını daxil edir:

Ardıcıl kod	Paralel kod
<pre>void main(){ double x[1000]; for(i=0; i<1000; i++){ calc_smth(&x[i]); } }</pre>	<pre>void main(){ double x[1000]; #pragma omp parallel for ... for(i=0; i<1000; i++){ calc_smth(&x[i]); } }</pre>

#Pragma omp parallel for Direktivi göstərir ki, bu dövr axınlar arasındakı təkrarlamalara görə bölünməlidir.

⁴³ <https://www.cyberforum.ru/blogs/18334/blog2965.html>

Mövzuların sayı proqramdan və ya proqramın işləmə müddəti - **OMP_NUM_THREADS** mühit dəyişənindən idarə edilə bilər.

Qeyd etmək lazımdır ki, tərtibatçı axınların sinxronlaşdırılmasına və məlumatlar arasındakı asılılığa cavabdehdir.

Proqramı dəstəkləmək üçün kompilyator əlavə açarı göstərməlidir:

```
icc -openmp prog.c  
ifc -openmp prog.f
```

Paylaşılan yaddaş modelində axınların qarşılıqlı əlaqəsi paylaşılan dəyişənlər vasitəsilə baş verir. Proqramdakı bu cür dəyişənlərlə diqqətsiz davranarsanız, rəqabət səhvləri (yarış vəziyyəti) yarana bilər. Bu, axınların paralel olaraq həyata keçirilməsindən və buna görə paylaşılan dəyişənlərə giriş ardıcılığının bir proqram başlanğıcından digərinə fərqli ola biləcəyindən qaynaqlanır⁴⁴.

Qeyd etmək lazımdır ki, sinxronizasiya proqramdan əlavə yüklü xərclər tələb edə bilər və məlumatları daha yaxşı düşünüb bölüşdürə bilər ki, sinxronlaşdırma nöqtələrinin sayı minimuma endirilsin.

Bu cür modeldə işləyən bir proqram nümunəsi olaraq əvvəlcə faydan və ya şəbəkədən (ardıcıl hərəkət) verilənləri oxuyan, sonra bir neçə axın (paralel hərəkət) işləyən, nəticəni fayla yazan və ya şəbəkə vasitəsilə (ardıcıl hərəkət) göndərən bir proqram göstərmək olar.

OpenMP-də paralellik və ardıcılıq

⁴⁴ https://ikt.muctr.ru/images/info/2Lecture_8.pdf

Paralel mühitə daxil olduqdan sonra yeni OMP_NUM_THREADS-1 axınları yaradılır, hər bir axının öz unikal sayğacı olacaq, burada başlanğıc axını 0 ədədlə işarələnəcək və bu baş axını (master) olacaqdır. Qalan axınlar 1 - dən OMP_NUM_THREADS-1-ə qədər tam ədədlər olacaqdır. Mövzuların sayı göstərilən paralel mühidə aparılır və bu mühidən çıxışa qədər dəyişməz qalır. Paralel mühidən çıxdıqdan sonra baş axınından başqa bütün axınlar sinxronizasiya yolu ilə aradan qaldırılır.

Aşağıdakı nümunə paralel direktivin necə işlədiyini göstərir. Nəticədə, əsas axın ekrana "1 - ardıcıl mühit" mətnini çap edir, sonra paralel direktiv yeni axınlar yaradır və bu axınların hər biri ekrana "paralel mühit" mətnini çap edir, sonra yaradılmış axınlar dayandırılır, və ekranda "2 - ardıcıl mühit" əsas axını çap olunur⁴⁵.

```
#include "stdafx.h"
#include <omp.h>
using namespace System;
int main(array<System::String ^> ^args)
{
    Console::WriteLine("1 – ardıcıl mühit");
    #pragma omp parallel
    {
        Console::WriteLine("paralel muhit");
    }
    Console::WriteLine("2 – ardıcıl mühit");
```

⁴⁵ <https://learn.microsoft.com/ru-ru/cpp/parallel/openmp/a-examples?view=msvc-170>

```
}
```

Bəzi hallarda sistemin özü sistem resurslarını optimallaşdırmaq üçün paralel mühitdə işləyən axınların sayını dinamik olaraq dəyişə bilər. Axınların sayının dinamik şəkildə dəyişdirilməsi `OMP_DYNAMIC` dəyişənini `true` olaraq təyin etməklə həyata keçirilir. Məsələn, Linux əməliyyat sisteminin `bash` əmr sətirində bu qiymət aşağıdakı əmrdən istifadə etməklə həyata keçirilə bilər:

```
export OMP_DYNAMIC = true;
```

Dinamik dəyişən sistemlərdə axınların sayı adətən dəqiqləşdirilmir, onun qiyməti isə `false`-dir.

`omp_in_parallel()` funksiyası aktiv vəziyyətdə paralel mühitdən çağırılsa, 1 qiyməti qaytarır.

Aşağıdakı nümunədə `omp_in_parallel()` funksiyası tətbiq edilmişdir. `Mode` funksiyası hansı mühitdən çağırıldığından asılı olaraq "paralel mühit" və ya "ardıcıl mühit" sətirlərini çap edərkən istifadə olunur⁴⁶.

```
#include "stdafx.h"  
#include <omp.h>  
using namespace System;  
void mode(void)  
{  
    if(omp_in_parallel())
```

⁴⁶ <https://cc.dvfu.ru/ru/364-2/>

```

        Console::WriteLine("paralel mühit");
    else
        Console::WriteLine("ardıcıl mühit");
}
int main(array<System::String ^> ^args)
{
    mode();
#pragma omp parallel
    {
#pragma omp master
        {
            mode();
        }
    }
    return 0;
}

```

C/C++ proqramlaşdırma dillərində yuxarıda göstərilən şərtlərin hamısı vahid direktivlə birgə elan olunacaq.

Proqramın müəyyən edilmiş hissəsini hansı axının idarə edəcəyi təsvir edilmir. Əgər hal-hazırda vəziyyət elan edilmirsə, bir axın müəyyən edilmiş fraqmenti yerinə yetirəcək, qalan axınlar isə onun bitməsinə gözləyir. Ümumi dəyişənlərlə işləyərkən vahid direktivə ehtiyac duyulur.

Baş direktiv kodun konkret hissəsini yalnız baş axını ilə müəyyən edir. Qalan axınlar bu hissəni buraxır və aşağıda olan operatorla proqramı işə salmağa davam edir. Bu

direktivdə sinxronizasiya yerinə yetirilmir. C/C++ Direkt proqramlaşdırma dilində direktiv aşağıdakı kimi elan edilir:

```
#pragma omp master
```

Aşağıdakı nümunə master direktivinin necə işlədiyini göstərir. N dəyişən yerli (lokal) hesab olunur və hər bir axın öz nüsxələri ilə işləyir. Əvvəlcə bütün axınlar N dəyişəninə 1 qiyməti təyin edir. Daha sonra əsas axın N dəyişəninə 2 qiyməti təyin edir və bütün axınlar bu dəyəri ekrana çap edir. Nümunədə gördüyünüz kimi, master direktivi həmişə bir axınla yerinə yetirilir. Bu misalda, bütün axınlar 1 rəqəmini çap edərkən, əsas axın əvvəlcə 2, sonra isə 3 rəqəmini çap edir:

```
#include "stdafx.h"
#include <omp.h>
using namespace System;
int main(array<System::String ^> ^args)
{
    int n;
    #pragma omp parallel private(n)
    {
        n=1;
    }
    #pragma omp master
    {
        n=2;
    }
}
```

```

        Console::WriteLine("n -in birinci qiyməti: "+ );
#pragma omp barrier
#pragma omp master
    {
        n=3;
    }
    Console::WriteLine("n -in növbəti qiyməti: "+
n);
    }
    return 0;
}

```

Aşağıdakı nümunədə private şərti yerinə yetirilir. Bu nümunədə paralel mühitdə N dəyişəni yerli dəyişən kimi elan edilir. Bu o deməkdir ki, hər axın N nüsxələri ilə işləyir və hər axının əvvəlində N dəyişənləri başlatır. Programı yerinə yetirərkən N dəyişkəninin dəyəri dörd fərqli yerdə çap olunur. Birinci dəfə N dəyişkəninin dəyəri 1-dən sonra ardıcıl mühitdə çap olunur, ikinci dəfə bütün axınlar paralel mühitin əvvəlində N dəyişkəninin nüsxəsini çap edir. Sonra bütün axınlar öz sıra nömrəsini çap edir, onun `omp_get_thread_num()` funksiyasından alınan N dəyərini verir. Paralel mühit başa çatdıqdan sonra N dəyişkəninin qiyməti bir daha çap olunur:

```

#include "stdafx.h"
#include <omp.h>
using namespace System;

```

```

int main(array<System::String ^> ^args)
{
    int n;
    Console::WriteLine("ardıcıl mühitə girişdə n qiyməti: "
+ n);
    #pragma omp parallel private(n)
    {
        Console::WriteLine("paralel mühitin girişində n
qiyməti: " + n);
        n=omp_get_num_threads();
        Console::WriteLine("paralel mühitdən çıxarkən
n qiyməti: " + n);
    }
    Console::WriteLine("ardıcıl mühitdən çıxarkən n
qiyməti: " + n);
    return 0;
}

```

C/C++ proqramlaşdırma dillərində proqramın paralel mühitində müəyyən edilmiş statik dəyişənlər paylaşılan dəyişənlərdir. Dinamik olaraq ayrılmış yaddaş da paylaşılır, lakin göstərici həm ortaq, həm də yerli ola bilər.

Thread private Direktivi C / C++ proqramlaşdırma dillərində yayılmış dəyişənləri yerli görünüşlərinə ötürə bilər. Qlobal obyektlərin yerli dəyişənlərindən düzgün istifadə edildikdə, onların proqramın müxtəlif hissələrində eyni axınlarda istifadə olunduğundan əmin olmaq lazımdır. Yerli dəyişənlər fərqli paralel mühitlərdən ünvanlandıqda,

dəyərlərini qorumaq üçün toplu paralel mühit olmamalıdır, axınların sayı hər iki mühitdə vahid olmalıdır və `omp_dynamic` dəyişəninin dəyəri birinci mühitin əvvəlindən ikinci mühitin əvvəlinə qədər `false` olaraq təyin olunmalıdır. `Thread private` tipində elan olunan dəyişənlər OpenMP direktivləri baxımından `copy in`, `copy private`, `schedule`, `num_threads_if` xaricində istifadə edilə bilməz.

Kilidlər. OpenMP-də sinxronizasiya seçimlərindən biri kilidləmə mexanizminin istifadəsidir. Kilidlər kimi ümumi tam dəyişənlər (ölçüsü ünvanı saxlamaq üçün kifayət olmalıdır) istifadə olunur. Bu dəyişənlər sinxronizasiya primitivləri üçün parametrlər kimi istifadə edilməlidir.

Kilidlər üç vəziyyətdən birində ola bilər: işə salınmamış, kilidli və ya kilidsiz. Bloklanmayan kilidi bəzi iplər tuta bilər. Bundan sonra onun statusu bloklanacaq. Kilidi açılmış kilid eyni iplə buraxıla bilər, bundan sonra kilid açılmış vəziyyətə qayıdacaq.

İki növ kilid var: sadə kilidlər və mürəkkəb kilidlər. Mürəkkəb kilidlər bir iplə buraxılmazdan əvvəl bir neçə dəfə əldə edilə bilər, sadə kilidlər isə yalnız bir dəfə əldə edilə bilər. Mürəkkəb qıfıllar üçün yuva sayı anlayışı təqdim olunur. Onun ilkin dəyəri sıfırdır, hər alınanda dəyəri bir artır, hər buraxılışda isə dəyəri bir azalır. Kompleks qıfıllar, doluluq dərəcəsi sıfır olduqda bloklanmır.

Sadə və mürəkkəb qıfılların initsializasiyası üçün C/C++ proqramlaşdırma dillərində aşağıdakı funksiyalardan istifadə olunur:

```
void omp_init_lock(omp_lock_t *lock);  
void omp_init_nest_lock(omp_nest_lock_t *lock);
```

C/C++ proqramlaşdırma dillərində o, aşağıdakı kimi elan edilir:

```
#pragma omp flush [(siyahı)]
```

➤ bu direktivin icrası zamanı axının registrlərində və keş yaddaşında saxlanılan bütün dəyişənlərin qiyməti əsas yaddaşa daxil edilir;

➤ axının işləməsi zamanı bütün dəyişənlərin dəyişməsi axınların qalan hissəsinə görünəcək;

➤ hər hansı məlumat çıxış buferində saxlanılırsa, buferlər atılır.

Bu zaman əməliyyat axının çağırdığı verilənlərlə yerinə yetirilir və digər axınlar tərəfindən dəyişdirilən verilənlərə toxunulmur.

Direktivdən istifadə nümunəsi olaraq, düzbucaqlı matrisin hər sətiri üçün elementlərin cəminin hesablanması məsələsini nəzərdən keçirək⁴⁷:

```
#include <omp.h>  
#define CHUNK 100  
#define NMAX 1000  
main () {
```

⁴⁷ http://hpc-education.ru/files/lectures/2011/gergel/gergel_2011_lecture03.pdf

```

int i, j, sum;
float a[NMAX][NMAX];
<verilənlərin insalizasiyası>
#pragma omp parallel shared(a) private(i,j,sum)
{
#pragma omp for
for (i=0; i < NMAX; i++) {
    sum = 0;
    for (j=0; j < NMAX; j++)
        sum += a[i][j];
    printf ("%d sətirinin elementlərinin cəmi
%f\n",i,sum);
} /* paralel fraqmentin sonu */
}

```

Suallar:

1. Paralel proqramlaşdırma texnologiyaları nədir?
2. Open MP texnologiyası nədir?
3. Open MP proqram alətində paralel proqramlaşdırma nədir?
4. Open MP-nin hansı üstünlükləri var?
5. Open MP proqramlaşdırma modelinə nələr aid etmək olar?
6. Open MP-də paralelik nədir?
7. Open MP-də ardıcılıq nədir?
8. Kilidlər nədir?
9. Kilidlər neçə vəziyyətdə ola bilər və onlar hansılardır?
10. Neçə növ kilid vardır?

Praktiki məşğələ № 8

Mövzu: Paralel proqramlaşdırma texnologiyaları

İşin məqsəd: *Paralel proqramlaşdırma texnologiyaları haqqında tələbələrə anlayış vermək.*

Nəzəri hissə

Dərhal demək lazımdır ki, bir neçə fərqli paralel proqramlaşdırma texnologiyası var. Bundan əlavə, bu texnologiyalar təkcə proqramlaşdırma dillərində deyil, həm də paralel sistemlərin qurulmasına arxitektura yanaşmalarında da fərqlənir.⁴⁸

Məsələn, bəzi texnologiyalar bir neçə kompüter (həm eyni, həm də müxtəlif növlər) əsasında paralel həllərin qurulmasını nəzərdə tutur, digərləri isə bir neçə prosessor nüvəsi olan bir maşın üzərində işləməyi nəzərdə tutur.

Bir neçə kompüterə əsaslanan sistemlər paylanmış hesablama sistemləri sinfinə aiddir. Bu cür həllər uzun müddətdir istifadə olunur, bu sahənin mütəxəssisləri tərəfindən yaxşı başa düşülür və onlar haqqında çoxlu ədəbiyyat var.

Paylanmış yaddaşa malik paralel kompüterlərin proqramlaşdırılması üçün ən çox yayılmış texnologiya MPI texnologiyasıdır. Belə sistemlərdə paralel proseslərin qarşılıqlı

⁴⁸ Лупин С. А., Посыпкин М. А. Технологии параллельного программирования. Учебное пособие. Издательство: Форум, 2019 г. Страниц: 208 (Офсет)

əlaqəsinin əsas yolu mesajların bir-birinə ötürülməsidir. Bu, texnologiyanın adında - Message Passing Interface-də öz əksini tapıb.⁴⁹

MPI-də paralel proqram eyni vaxtda icra olunan proseslər toplusu kimi başa düşülür. Proseslər müxtəlif prosessorlarda işləyə bilər, lakin eyni prosessorla bir neçə proses yerləşə bilər (bu halda onların icrası vaxt mübadiləsi rejimində həyata keçirilir). Məhdudiyətli halda, bir prosessor paralel proqramı icra etmək üçün istifadə edilə bilər - bir qayda olaraq, bu üsul paralel proqramın düzgünlüyünün ilkin yoxlanılması üçün istifadə olunur.

Paralel proqramın hər bir prosesi eyni proqram kodunun sürəti əsasında yaradılır. İcra edilə bilən proqram kimi təqdim edilən bu proqram kodu paralel proqramın işə salınması zamanı bütün istifadə olunan prosessorlarda mövcud olmalıdır.⁵⁰

İcra olunan proqram üçün mənbə kodu MPI kitabxanasının bu və ya digər tətbiqindən istifadə edərək C və ya Fortran alqoritmik dillərində hazırlanmışdır.

Proseslərin sayı və istifadə olunan prosessorların sayı paralel proqramın işə salındığı anda MPI proqramının icra mühiti vasitəsi ilə müəyyən edilir və dinamik proseslərin

⁴⁹ Основы параллельного программирования с использованием технологий MPI и OpenMP: учебное пособие / Р.В. Жалнин, Е.Н. Панюшкина, Е. Е. Пескова, П.А. Шаманаев. – Саранск: Изд-во СВМО, 2013. – 78 с.

⁵⁰ Основы параллельного программирования с использованием технологий MPI и OpenMP: учебное пособие / Р.В. Жалнин, Е.Н. Панюшкина, Е. Е. Пескова, П.А. Шаманаев. – Саранск: Изд-во СВМО, 2013. – 78 с.

yaradılması və idarə edilməsi üçün xüsusi, lakin nadir hallarda istifadə olunan alətlərdən istifadə etmədən hesablamalar zamanı dəyişə bilməz. MPI versiyası 2.0 standartında ortaya çıxdı. Bütün proqram prosesləri ardıcıl olaraq 0-dan p1-ə qədər nömrələnir, burada p proseslərin ümumi sayıdır. Proses nömrəsinə proses dərəcəsi deyilir.

MPI-nin əsasını mesaj ötürmə əməliyyatları təşkil edir. MPI-də təqdim olunan funksiyalar arasında iki proses arasında qoşalaşmış (nöqtədən nöqtəyə) əməliyyatlar və bir neçə prosesin eyni vaxtda qarşılıqlı əlaqəsi üçün kollektiv (kollektiv) kommunikasiya hərəkətləri var.

Cüt əməliyyatları həyata keçirmək üçün müxtəlif ötürmə rejimlərindən istifadə edilə bilər, o cümlədən sinxron, bloklayıcı və s.

Paralel proqram prosesləri qruplara birləşdirilir. MPI-nin proseslər dəstini təsvir edən digər mühüm anlayışı kommunikator anlayışıdır. MPI-də kommunikator dedikdə xüsusi yaradılan xidmət obyektinə başa düşülür ki, bu da öz tərkibində bir qrup prosesi və bir sıra əlavə parametrləri (kontekst) birləşdirir.

Verilənlərin cüt ötürülməsi yalnız eyni kommunikatora məxsus proseslər üçün həyata keçirilir. Kollektiv əməliyyatlar bir kommunikatorun bütün prosesləri üçün eyni vaxtda tətbiq olunur. Nəticə olaraq, istifadə olunan kommunikatorun göstərilməsi MPI-yə verilənlərin ötürülməsi üçün məcburidir.

Hesablamalar zamanı proseslərin və kommunikatorların yeni qrupları yaradıla və mövcud qruplar silinə bilər. Eyni proses müxtəlif qruplara və kommunikatorlara aid ola bilər.

Paralel proqramda mövcud olan bütün proseslər MPI_COMM_WORLD identifikatoru ilə standart kommunikatora daxil edilir.

Mesaj ötürmə əməliyyatlarını yerinə yetirərkən, MPI funksiyalarında göndəriləcək və ya qəbul ediləcək məlumatları müəyyən etmək üçün göndəriləcək məlumatların növünü müəyyən etmək lazımdır. MPI C və Fortran alqoritmik dillərindəki məlumat növləri ilə əsasən üst-üstə düşən əsas məlumat növlərinin böyük dəstini ehtiva edir. Bundan əlavə, MPI göndərilən mesajların məzmununun daha dəqiq və qısa təsviri üçün yeni əldə edilmiş məlumat növləri yaratmaq imkanına malikdir.

Beləliklə, qısaca desək, MPI mesaj ötürmə mexanizmindən istifadə edərək paralel proseslərin qarşılıqlı əlaqəsini təmin edən funksiyalar kitabxanasıdır. Bu, təxminən 130 funksiyadan ibarət kitabxanadır, o cümlədən:

- ✓ MPI proseslərinin işə salınması və bağlanması üçün funksiyalar;
- ✓ Qoşalaşmış əməliyyatları həyata keçirən funksiyalar;
- ✓ Kollektiv əməliyyatları həyata keçirən funksiyalar;
- ✓ Proses qrupları və kommunikatorlarla işləmək üçün funksiyalar;
- ✓ Məlumat strukturları ilə işləmək üçün funksiyalar;
- ✓ Proseslərin topologiyasının formalaşdırılması funksiyaları.

Lakin bir maşında işləmək üçün paralel proqramlaşdırma sistemləri nisbətən bu yaxınlarda hazırlanmağa başladı. Xeyr, əlbəttə ki, bu barədə tamamilə yeni fikirlər kimi düşünməməlisiniz, lakin masaüstü kompüterlərdə çoxnüvəli sistemlərin (daha doğrusu, onların son görünüşü ilə) yaranması ilə inkişaf etdiricilər OpenMP, Intel Thread Building Blocks, Microsoft Parallel Extensions kimi texnologiyalara diqqət yetirməlidirlər. və s.

Paralel proqramlaşdırma texnologiyasının proqramın tədricən paralelləşdirilməsi imkanını dəstəkləməsi çox vacibdir. Bəli, aydındır ki, ideal paralel proqram bir anda paralel olaraq yazılmalı və ya hətta paralelləşmənin heç bir problemi olmadığı hansısa funksional dildə daha yaxşı yazılmalıdır...

Amma proqramçılar real dünyada yaşayırlar və işləyirlər. Yeni bucaqlı funksional F#10 MB kodu C++ və ya ümumiyyətlə C dilində ən yaxşısıdır. Bu kod yavaş-yavaş paralel olmalıdır. Bu halda, OpenMP texnologiyası (məsələn) çox yaxşı seçim olardı. Tətbiqdə paralelləşdirmə üçün ən zəruri sahələri seçməyə və əvvəlcə onları paralelləşdirməyə imkan verir. Praktikada belə görünür. Bəzi profil alətləri ilə proqramçı proqramda ən uzun müddət davam edən darboğazları axtarır. Niyə alətdən istifadə edin? Çünki naməlum 10 Mb layihədə düşüncə gücü “maneə” tapa bilmir. Daha sonra tərtibatçı OpenMP istifadə edərək bu blokları paralelləşdirir. Bundan sonra, istədiyiniz proqram göstəriciləri əldə olunana qədər növbəti maneələri və s. Paralel versiyanın hazırlanması prosesi dayandırılı, aralıq buraxılışlar buraxıla

və lazım olduqda geri qaytarıla bilər. Buna görə də, xüsusilə, OpenMP texnologiyası çox populyarlaşdı.

OpenMP texnologiyası ənənəvi proqramlaşdırma dillərinə əsaslanan ümumi yaddaşa malik kompüter sistemləri üçün ən populyar proqramlaşdırma vasitələrindən biridir. OpenMP kompilyatorlar və xüsusi funksiyalar kitabxanaları üçün direktivlər dəstindən ibarətdir.

İlk OpenMP standartı 1997-ci ildə asanlıqla daşına bilən çox yönlü proqramların yazılmasına yönəlmiş API kimi hazırlanmışdır. Əvvəlcə Fortran dili əsasında quruldu, lakin sonradan C və C++ dillərini əhatə etdi.

OpenMP interfeysi ən populyar paralel proqramlaşdırma texnologiyalarından birinə çevrilmişdir. OpenMP həm çoxlu prosessorlu superkompüter sistemlərinin proqramlaşdırılmasında, həm də masaüstü istifadəçi sistemlərində və ya məsələn, Xbox 360-da uğurla istifadə olunur.

OpenMP spesifikasiyası işi qeyri-kommersiya təşkilatı olan OpenMP Architecture Review Board (ARB) tərəfindən tənzimlənən bir neçə əsas aparat və proqram təminatçısı tərəfindən hazırlanır.

OpenMP filial birləşməsi ilə paralel icra modelindən istifadə edir. OpenMP proqramı başlanğıc ipi adlanan tək icra xətti kimi başlayır. İp paralel strukturla qarşılaşdıqda özündən və bir sıra əlavə iplərdən ibarət yeni ip qrupu yaradır və yeni iplik qrupunda əsas ipə çevrilir. Yeni qrupun bütün üzvləri (ana daxil olmaqla) paralel quruluş daxilində kodu icra edirlər. Paralel tikintinin sonunda gizli maneə var. Paralel tikintidən sonra yalnız əsas ip xüsusi kodu icra etməyə davam

edir. Digər paralel bölgələr, orijinal bölgənin hər bir axınının öz axınlar qrupu üçün əsas axına çevrildiyi paralel bölgədə yerləşə bilər. Yuvalanmış bölgələr, öz növbəsində, daha dərin yuva səviyyəsində bölgələri ehtiva edə bilər.

Paralel qrupdakı iplərin sayı bir neçə yolla idarə oluna bilər. Onlardan biri `OMP_NUM_THREADS` mühit dəyişənini istifadə edir. Başqa bir yol `omp_set_num_threads()` proseduru çağırmaqdır. Başqa bir yol, paralel təlimatla birlikdə `num_threads` ifadəsindən istifadə etməkdir.

Paralel proqramlaşdırma üzrə qarşıdan gələn qeydlərin elanı

Bu yazı ilə biz OpenMP texnologiyasını və paralel proqramlaşdırma alətlərini təqdim etməyə həsr olunmuş kiçik bir sıra yazılara başlayırıq. Aşağıdakı girişlərdə öyrənəcəksiniz:

- paralel proqramların hazırlanması üçün hansı vasitələrə ehtiyac var;
- sıfırdan paralel proqramı necə yaratmaq;
- OpenMP texnologiyasından istifadə etməklə mövcud proqrama paralel icra necə əlavə edilə bilər;
- OpenMP-tətbiqlərinin işlənilməsi hazırlanmasında hansı tipik problemlər var və onları necə müəyyənləşdirmək olar;
- paralel proqramların optimallaşdırılması.

Suallar:

- 1.** Hansı kompüterlər paylanmış hesablama sistemləri sinfinə aiddir?
- 2.** MPI (Message Passing Interface) nədir?
- 3.** MPI-nin əsasını nə təşkil edir?
- 4.** MPI-də kommunikator anlayışı nədir?
- 5.** MPI mesaj ötürmə mexanizmindən istifadə edərək paralel proseslərin qarşılıqlı əlaqəsini təmin edən funksiyalar hansılardır?
- 6.** Open MP texnologiyası hansı sahələrdə tətbiq olunur?
- 7.** Paralel proqramların hazırlanması üçün hansı vasitələrə ehtiyac var?
- 8.** Sıfırdan paralel proqramı necə yaratmaq olar?
- 9.** Open MP texnologiyasından istifadə etməklə mövcud proqrama paralel icra necə əlavə edilə bilər?
- 10.** Open MP tətbiqlərinin işlənilib hazırlanmasında hansı tipik problemlər var və onları necə müəyyənləşdirmək olar?

ƏDƏBİYYAT

1. Bayramov H.M., Mənsimov H.İ., Məmmədov Ə.S. Kompüter şəbəkələrinin əsasları. Dərs vəsaiti. Bakı: “İqtisad Universiteti” Nəşriyyatı – 2019. -142 səh

2. Məmmədov M.İ., Orucova M.Ü., Bayramova N.M.– Kompüter şəbəkələri. (Dərs vəsaiti). ADAU nəşr., 2014. -136 səh.

3. Məmmədov M. L, Zeynalov Z.H.- Fərdi kompüterlər (Dərs vəsaiti). Gəncə:. AKTA. 2016.-118 səh.

4. Məmmədov C.F.”Kompüter şəbəkələri və sistemləri” fənnindən mühazirələr konspekti. Sumqayıt, 2006, -44 səh.

5. Mənsurov Q.M. Kompüter mühəndisliyinin müasir problemləri (Mühazirələr konspekti). – Sumqayıt: SDU, 2016.- 30 s.

6. OpenMP Architecture Review Board
(<http://openmp.org/wp/>)

7. Rəşid Ələkbərov, Məmməd Həşimov, Şəbəkə mühitində paylanmış hesablama sistemlərinin yaradılması texnologiyaları, 2015

8. Алекперов Р.Г., Гашимов М.А. Технологии разработки распределенных вычислительных систем в сетевой среде. Экспресс-информация. Серия «Информационные технологии». Издательство «Информационные технологии», 2015

9. Архитектура ЭВМ и вычислительных систем : учебник /Н. В. Максимов, Т. Л. Партыка, И. И . Попов. —

5-е изд., перераб.и доп. — М. : ФОРУ : ИНФРА-М , 2013. — 512 с. : ил.

10. Е.Н. Панюшкина, Е. Е. Пескова, П.А. Шаманаев. – Саранск: Изд-во СВМО, 2013. – 78 с.

11. Илья Федотов: Параллельное программирование. Модели и приемы. Солон-пресс, 2017 г. Страниц: 390. Подробнее: <https://www.labyrinth.ru/books/581646/>

12. Лупин С. А., Посыпкин М. А. Технологии параллельного программирования. Учебное пособие. Издательство: Форум, 2019 г. Страниц: 208 (Офсет)

13. Михайлов Н.Л. Архитектура вычислительных систем: Учебное пособие. – Рыбинск, РГАТА, 2008. – 87 с.

14. Основы параллельного программирования с использованием технологий MPI и OpenMP: учебное пособие / Р.В. Жалнин,

15. Сысоев А.В., Мееров И.Б., «Параллельное программирование в системах с общей памятью. Инструментальная поддержка». Учебно-методические материалы по программе повышения квалификации «Технологии высокопроизводительных вычислений для обеспечения учебного процесса и научных исследований». Нижний Новгород, 2007, 110 с.

16. Соснин В.В., Балакшин П.В. Введение в параллельные вычисления. –СПб: Университет ИТМО, 2015. – 51 с.

INTERNET RESURLARI

1. <http://bigor.bmstu.ru/?cnt/?doc=Parallel/ch010101.mod/?cou=Parallel/base.cou>
2. <http://ccfit.nsu.ru/arom/data/openmp>
3. <http://computerologia.ru/topologii-seti-osnovnye-xarakteristiki/>
4. http://db4.sbras.ru/elbib/data/show_page.phtml?77+854
5. <http://e-derslik.edu.az/books/651/units/unit-1/page304.xhtml>
6. http://hpc-education.ru/files/lectures/2011/gergel/gergel_2011_lecture03.pdf
7. <http://kom-seti.narod.ru/index.files/4.htm>
<https://sites.google.com/site/seticomptech/topologia-setej>
8. <http://ks-211.blogspot.com/2015/06/lokal-sbklr.html>
9. http://netaz.byethost9.com/sebeke_tesnifati.html?i=1
10. <http://sebekecomputer.blogspot.com/>
11. <http://www.berkut.ws/comporgtech.html>
12. <https://buom.ru/parallelnoe-programmirovani-opredelenie-preimushhestva-i-primenenie-v-otrasli/>
13. <https://buom.ru/parallelnoe-programmirovani-opredelenie-preimushhestva-i-primenenie-v-otrasli/>
14. <https://habr.com/ru/company/intel/blog/82486/>
15. <https://cc.dvfu.ru/ru/364-2/>
<https://cyberleninka.ru/article/n/soderzhanie-kursa-osnovy-parallelnogo-programmirovaniya>

16. https://dic.academic.ru/dic.nsf/ruwiki/2049?ysclid=18j_uwwf8re241614510
17. https://dic.academic.ru/dic.nsf/ruwiki/2049?ysclid=18j_uwwf8re241614510
18. https://ewikiaz.top/wiki/Supercomputer_architecture
19. <https://halzen.ru/az/windows-7/posledovatel'naya-i-parallelnaya-obrabotka-informacii.html>
20. https://ikt.muctr.ru/images/info/2Lecture_8.pdf
21. <https://intuit.ru/studies/courses/57/57/lecture/1672?page=3#image.1.5>
22. <https://intuit.ru/studies/courses/57/57/lecture/1672?page=3#image.1.6>
23. <https://intuit.ru/studies/courses/57/57/lecture/1672?page=4>
24. <https://intuit.ru/studies/courses/57/57/lecture/1672?page=4>
25. <https://lanfix.ru/clauses/typy-kompjuternyh-setej-i-sposoby-ih-upravlenija/>
26. <https://learn.microsoft.com/ru-ru/cpp/parallel/openmp/a-examples?view=msvc-170>
27. <https://megaobuchalka.ru/1/74.html>
28. <https://net.e-publish.ru/p216aa1.html>
29. <https://parallel.ru/computers/classes.html>
30. <https://plusiminusi.ru/osnovnye-plyusy-i-minusy-topologii-kolco/>
31. <https://plusiminusi.ru/plyusy-i-minusy-topologii-zvezda/>

32. <https://siblec.ru/telekommunikatsii/vychislitelnye-sistemy-seti-i-telekommunikatsii/5-arkhitektury-vysokoproizvoditelnykh-vychislitelnykh-sistem/5-2-simmetrichnaya-mnogoprotsessornaya-arkhitektura-smp?ysclid=I9g29u9ppt104549534>

33. <https://siblec.ru/telekommunikatsii/vychislitelnye-sistemy-seti-i-telekommunikatsii/5-arkhitektury-vysokoproizvoditelnykh-vychislitelnykh-sistem/5-2-simmetrichnaya-mnogoprotsessornaya-arkhitektura-smp?ysclid=I9g29u9ppt104549534>

34. https://studbooks.net/2029357/informatika/topologii_vychislitelnoy_seti

35. <https://studfile.net/preview/1511376/page:5/>

36. https://studme.org/151249/informatika/arkhitektura_parallelnyh_vychislitelnyh_sistem

37. <https://textarchive.ru/c-2917188.html>

38. <https://www.ckofr.com/rabota-v-internete/939-massivno-parallelnye-sistemy-obrabotki-mpp>

39. <https://www.cyberforum.ru/blogs/18334/blog2965.html>

40. <https://www.javatpoint.com/sisd>

MÜNDƏRİCAT

Giriş	3
Praktik məşgələ №1	
Paralel hesablama sistemləri.....	5
Praktik məşgələ №2	
Müasir super kompüterlər və onların növləri.....	30
Praktik məşgələ №3	
Müasir kompüterlərin yaddaşın təşkilinə görə təsnifatı.....	40
Praktik məşgələ №4	
Müasir parallel kompüterlərin əsas sinifləri.....	51
Praktik məşgələ №5	
Kommunikasiya şəbəkələrinin topologiyası.....	61
Praktik məşgələ №6	
Paralel proqramlaşdırmanın tətbiq sahələri.....	81
Praktik məşgələ №7	
Open MP proqram alətində parallel proqramlaşdırma.....	89
Praktik məşgələ №8	
Paralel proqramlaşdırma texnologiyaları.....	104
Ədəbiyyat.....	112
İnternet resursları.....	114

Çap evinin direktoru: Toğrul Abasov
Üz qabığının tərtibatı: Ariz Babayev

Kitab "STAR" çap evində səhifələnmiş və çap edilmişdir.

Çapa imzalanıb: 23.02.2023
Format: 60x90 1/16. Həcmi: 7,25 ç.v.,
Tiraj: 500 əd. Sifariş № 500



Gəncə, Cavadxan küç., 77a
+994 55 624 34 44,
+994 22 266 06 63
+994 55 930 06 63
e-mail:6243444@mail.ru