

Планиране на ресурсите в паралелни компютърни системи

Планирането на ресурсите в паралелните компютърни системи обхваща два етапа:

- ü Етап 1 - групиране на изчислителните ресурси;

- ü Етап 2 - разпределение на задачите между процесорите в рамките на една група;

Първият етап от планирането на изчислителните ресурси е необходим в мултипрограмен режим. Процесорите се разделят на групи като всяка група обработва една програма. Основният проблем при първия етап е да се определи броят на процесорите в рамките на отделните групи за всички изпълнявани програми по такъв начин, че да се постигнат максимална производителност и максимално оползотворяване на изчислителните ресурси. Това зависи от изчислителния системен товар както и от характеристиките на програмата, и по-точно колко ефективно тя може да оползотвори определен брой процесори.

При втория етап на планирането паралелните задачи в програмата се разпределят между процесорите в рамките на една група. Основната цел е постигането на минимално време за изпълнение. Конкретната схема за разпределяне на задачите между процесорите зависи от броя на процесорите в групата, топологията на вътрешносистемната комуникационна мрежа, както и причинно – следствените връзки между задачите в програмата.

Основна архитектурна особеност на паралелните машини е, че те имат модулна структура с осигурена възможност за разпределяне на процесорите в групи. Групирането в процесорната мрежа трябва да бъде прозрачно за потребителя и неговата среда за програмиране. С оглед на тези изисквания са по-ефективни паралелните архитектури с разпределена памет. Архитектурите с обща памет имат своите предимства, но не допускат ефективно прегрупиране на изчислителните ресурси.

Вътрешносистемната комуникационна структура е ключов фактор при проектирането на паралелните компютри. Комуникационната структура трябва да бъде регулярна (модулна) за да осигури ефективно прегрупиране на процесорите, както и да има малък диаметър и малък брой на връзките.

Топология хиперкуб и нейните варианти са много популярни и са заложи в множество съвременни комерсиални паралелни компютри. Диаметърът на хиперкуб с N възела, както и неговата степен са

$O(N \log$

2

$N)$

.

Ключовите фактори при паралелните компютърни архитектури са броят на процесорите, пропускателната способност на вътрешносистемната комуникационна мрежа, както и пропускателната способност на входно-изходния тракт.

С увеличаване на броя на процесорните елементи, проблемът за тяхното планиране става още по критичен. Аналогията с ефективното използване на паметта при еднопроцесорни системи показва, че дори при голям капацитет на паметта лошата стратегия за нейното разпределяне може да доведе до значителна деградация на системната производителност.

За изготвянето на ефективна стратегия за планирането за паралелните изчислителни ресурси е необходима предварителна оценка на работния изчислителен товар за изпълнението на конкретната програма. В случая оценката на работния товар е свързана с изчислителните и комуникационните изисквания като функция от броя на процесорите, изпълняващи програмата. Ефективната стратегия за планиране трябва да осигурява адаптиране към конкретния работен товар. За съжаление, съществуват допълнителни системни загуби свързани с прилагането на такива динамични схеми за планиране, особено в масивно паралелни системи. Горепосочените проблеми в съчетание с архитектурните особености на паралелния компютър трябва да бъдат взети под внимание през първия етап на планирането на ресурсите.

През вторият етап на планирането трябва да се вземат под внимание разходите за комуникация в конкретната паралелна архитектура, както и причинно – следствените

връзки в графа на задачата. Естествено, най-простата схема е да се разпределят задачите между процесорите по случаен принцип. При разпределянето на много голям брой задачи между много голям брой процесори допълнителните разходи за комуникация могат да нарастнат драстично, което да доведе до значително увеличаване на времето за изпълнение на програмата. Намаляване на времето за изпълнение може да се постигне с оптимално разпределение на задачите, цената на което значително нараства с увеличаване на размера на задачата и броя на процесорите.

В общия случай, сложността на проблема за разпределяне на изчислителните ресурси е NP

Следователно, целта е разработването на евристични схеми за разпределяне, които не изискват големи допълнителни разходи (интензивни изчисления).

Планиране на процесорите

От гледна точка на планирането на ресурсите, паралелната комютърна архитектура логически може да се разглежда като йерархична структура с две нива на управление. На горното ниво управлението се осъществява от фронт-енд система, осигуряваща интерфейса с потребителя.

На ниското ниво, в рамките на дадена група процесори един от процесорите координира дейността на групата паралелно с участието си в изчислителния процес.

Анализът на йерархичния модел показва следните особености:

• Процесорите се разглеждат като съдържание на контейнер, от който могат да бъдат формирани групи. Размерът на конкретните групи зависи от архитектурните особености на паралелния комютър.

ÿ След формирането на дадена група, един от процесорите в нея се определя за групов координатор. Изборът на координатор може да се направи при формирането на групата за изпълнението на конкретната програма или да е фиксиран предварително.

Архитектурите с обща памет се разделят на *архитектури с унифициран достъп и машини с неунифициран достъп*. Паралелните архитектури, които нямат обща памет и комуникират чрез обмен на съобщения (наричат се архитектури с разпределена памет или машини с комуникиращи процеси), могат да бъдат класифицирани също като машини с унифициран достъп и машини с неунифициран достъп. При паралелните компютри с неунифициран достъп, времето за достъп се определя от разстоянието между произволна двойка процесори. Така например към архитектурите с разпределена памет с унифициран достъп спадат паралелните компютри с многостъпални вътрешносистемни връзки. Архитектурите с комуникиращи процеси и топология хиперкуб спадат към класа с неунифициран достъп. Различните архитектурни стилове имат различно влияние върху отделните етапи на планирането на процесорите.

Представяне на работния изчислителен товар.

Много стратегии за планиране при еднопроцесорните системи изискват предварителна информация за параметрите, характеризиращи различните изисквания към ресурсите за изпълнението на конкретна програма. Фактът, че тази информация може да липсва при реалните системи не намалява важността на изследването на характеристиките на товара или на приложимостта на тези стратегии за планиране. Дори и грубата оценка на работния изчислителен товар може значително да повиши системната производителност. Естествено, това е в сила и при мултипроцесорите. Определянето на характеристиките на работния товар е много по-сложен проблем при паралелните компютърни архитектури в сравнение с еднопроцесорните системи. Това се дължи главно на голямото разнообразие на съвременните паралелни архитектури, които се различават в много аспекти: структурата на управление, организацията на паметта, вътрешносистемната комуникационна мрежа и организацията на входно-изходните операции.

Различните подходи, които се използват за оценка на характеристиките на работния изчислителен товар при изпълнението на паралелната програма могат да бъдат класифицирани в две основни групи. Първата група се основава на сигнатурата на

изпълнението, докато втората група се базира на профила на паралелизма.

Сигнатурата на изпълнението представя зависимостта на времето за изпълнение на паралелния алгоритъм от различен брой процесори. Времето за изпълнение $T(p)$ на паралелната програма от

p процесори може да се разглежда като съставена от две компоненти:

$$T(p) = T_{comp}(p) + T_{comm}(p) \quad (1)$$

където $T_{comp}(p)$ представя времето за изчисления, а $T_{comm}(p)$ е времето за допълнителните разходи за комуникация.

В общия случай $T_{comp}(p)$ е монотонно намаляваща функция докато $T_{comm}(p)$ е растяща функция на

p (в най-добрия случай е константа).

Концепцията за работното множество процесори се основава на сигнатурата на изпълнението и е създадена с цел да се характеризира поведението на паралелната програма при изпълнението ѝ от конкретната архитектура.

Работното множество процесори за дадена паралелна програма се дефинира по следния начин: нека $C(p)$ е функция, отразяваща цената на изпълнението, за която

$$C(p) = p T(p) / T(1) = p / S(p) \quad (2)$$

Отбележете, че ако ускорението е линейно т.е. $S(p) = p$, то цената за използване на повече процесори е малка (т.е. е равна на 1). От друга страна, ако ускорението е малко, то цената е пропорционално по-висока.

Характеристиката ефикасност $\eta(p)$ се дефинира като ускорението за единица цена:

$$\eta(p) = S^2(p) / p \quad (3)$$

Работното множество процесори се дефинира като минималният брой процесори, който максимизира $\eta(p)$.

Работното множество процесори е параметър, който добре характеризира поведението на паралелната програма при изпълнението ѝ в специфична архитектура. Този параметър е чувствителен както към промените в алгоритъма, така и към такива в архитектурата. Максималните стойности на $\eta(p)$ и $S(p)$ за конкретно приложение могат да бъдат достигнати за различен брой на процесорите (p).

Другият подход за характеризиране на поведението на паралелната програма се основава на *профил на паралелизма*. Той дава броя на заетите процесори като функция на времето и се получава като се изпълнява паралелният алгоритъм от идеализирана машина с неограничен брой процесори без допълнителни системни разходи. От профила на паралелизма могат да бъдат определени множество други мерки. Например, *средният паралелизъм A* се дефинира като

$$A = T(1) / T(p) \quad (4)$$

където $T(1)$ е времето за изпълнение от един процесор и $T(p)$ е времето за изпълнение от неограничен брой процесори. В тази дефиниция допълнителните разходи за комуникация не са включени в профила на паралелизма. Профилът на паралелизма се проследява за специфична архитектура с фиксиран брой процесори. Друго по-просто представяне е *формата на паралелизма* (α)

Shape),

която може да бъде извлечена от профила на паралелизма. Тя представлява графично представяне на сумарното време, в рамките на което определен брой процесори са заети.

Проблемът за планирането на процесорите изисква информация за причинно-следствените връзки между задачите в програмата. Тази информация в общия случай се представя под формата на насочен ацикличен граф, в който възлите представят изчисленията, а дъгите отразяват причинно-следствените връзки между задачите. Дъгите могат да бъдат маркирани с тегловни коефициенти представлящи количеството данни, които трябва да бъдат обменени между съответните възли. Повечето стратегии за планиране изискват познаване на графа на задачите и вътрешната комуникационна структура на паралелния компютър.

Стратегии за групиране на процесорите

Съществуват два основни подхода за групиране на процесорите в мултипрограмен режим. При статичните стратегии фиксиран брой процесори изпълняват конкретната програма до нейното завършване. При динамичните стратегии броят на процесорите, заети с изпълнението на дадена програма, варира по време на нейното изпълнение в съответствие с промените на броя на изпълняваните задачи в програмата.

Статични стратегии. Информация за характеристиките на програмата

При еднопроцесорните системи, всяка предварителна информация за програмата значително улеснява планирането на ресурсите и впоследствие влияе ползотворно върху системната производителност. За мултипроцесорите е показано, че при вземане под внимание на характеристиките на програмата планирането на ресурсите е много по-ефективно. При прилагане на елементарни стратегии за планиране от вида на *“първ и дошъл, първи обслужен” (First Come First Served - FCFS)*

или

кръгово обслужване (Round-Robin - RR)

, които не зависят от характеристиките на програмата се получава по-ниско бързо действие в сравнение със стратегиите, изискващи информация за броя на паралелните задачи в програмата и кумулативните изисквания за обслужване на задачите. Такива стратегии са

“най-малкият брой процеси се обслужва първи” (*Shortest Number of Processes First - SNPF*)

и

“процесите с най-кратко кумулативно обслужване се обработват първи” (*Shortest Cumulative Demand First-SCDF*).

Подходите, които се основават на множество параметри и характеристики, осигуряват по-добра системна производителност от тези, базиращи се на оценката на единствен параметър. Препоръчително е, планирането на ресурсите да се основава на профила на паралелизма на програмата, както и оценката на производните характеристики като среден паралелизъм и максимален/минимален паралелизъм. Важният въпрос, който възниква тук е количеството на информацията, осигуряваща ефективен механизъм на планиране.

Влияние на системния товар

Начинът, по който методът за групиране на процесорите трябва да се адаптира към системния работен товар е от изключителна важност. В случаите, при които товарът е малък т.е. в даден момент в паралелния компютър се изпълнява само една програма, стратегията трябва да осигурява използването на всички процесори от тази програма. При голям товар, обаче, т.е. когато няколко програми чакат обслужване, най-добрата стратегия е за всяка програма да се използват по малък брой процесори. В случая се максимизира използването на процесорите тъй като обикновено паралелните програми имат ускорение по-малко от линейна зависимост.

Следните три стратегии са от най-често използваните:

1. **Първо съвпадение (*First Fit - FF*)**: При тази стратегия планиращата програма анализира чакащите програми през прозореца и избира първата програма, чието работно множество процесори (pws) съвпада или се удовлетворява от броя на незаетите процесори. Поради факта, че за всяка програма се заделят толкова процесора, колкото е нейното pws

, възможно е да има престояващи процесори докато в опашката има чакащи програми.

2. **Стратегия с комбинирано прилагане на принципите “Първо попадение” и “пръв влязал, пръв излязъл” (*FF+FIFO*)**

:

При тази стратегия, за изпълнението на дадена програма се определят процесори, чийто брой е най-много равен на работното множество процесори

rws

на тази програма. Планиращата програма избира през прозореца от опашката първата програма, чието

rws

е по-малко или равно на броя на свободните процесори. След планирането, ако останат свободни процесори, те се използват за изпълнението на първата програма в опашката.

3. Сегментиране на процесорната мрежа: При тази стратегия процесорите се разпределят на фиксиран брой групи, като всяка група съдържа 16 процесора т.е. заделят се по 16 процесора на програма или се конфигурират 16 еднопроцесорни групи. За всяка програма се определя един сегмент независимо от нейното

rws

Резултатите от моделирането показват най-добра производителност при стратегията за планиране *FF+FIFO*. При нея за всяка програма се заделят толкова процесори, колкото е нейното работно множество процесори *rws* и по един процесор на програма при интензивно натоварена система.

Времеделене

Основният проблем при стратегиите *RTC* и тези, базирани на *rws* е, че програмите не са равнопоставени. При тях кратките програми трябва да чакат дълго време за достъп до изчислителните ресурси докато големите програми монополизират процесорите. Равнопоставеност между програмите може да се достигне като се предоставят на всички програми в опашката фиксирани интервали от време за обработка с кръгова дисциплина на обслужване (

time – slicing

). При тази стратегия, наричана още копланиране, процесите на всички програми се съхраняват в една обща опашка на процесите. Планирането се извършва като се мести прозорец с размер равен на броя на процесорите през опашката. Всеки процес в прозореца получава квант обслужване от един процесор. В края на кванта, прозорецът се премества по-нататък по дължината на опашката докато първият слот на прозореца покрие първия процес, който не е бил планиран по време на предишния квант.

Съществува вариант на този метод, при който кръговото обслужване се извършва на

ниво програми. Всички процесори обслужват процесите на една и съща програма в рамките на един квант. Обслужването на програмите последователно по квантове се извършва с кръгова дисциплина на обслужване. *Този метод се нарича **групово планиране***

Методите с кръгово обслужване не са адаптивни по отношение на специфичния товар. Допълнителните разходи за контекстно превключване в края на всеки квант и информационните липси в кеша при новите процеси в началото на всеки квант води до деградация на системната производителност и ефективност.

Стратегии с използване на предимство

В еднопроцесорните системи осигуряването на предимство на програмите води до повишаване на системната производителност и дава възможност за “справедливо” разпределяне на изчислителните ресурси в мултипроцесорните системи. Висок приоритет се дава на програми с по-ниски изисквания за кумулативно обслужване. Аналогично на еднопроцесорните системи, в паралелните компютри ефектът с даване на предимства е значителен само когато коефициентът на промяна на изискването за обслужване е висок. Тук съществуват два ключови фактора, които влияят върху производителността. На първо място, процесите с предимство, които подлежат на синхронизация, могат да доведат до значително спадане на производителността. Второ, информационните липси в кеша свързани със стартирането на нов процес също имат отрицателен ефект върху бързо действието.

Конфликти за заемане на ресурсите

Двата основни ресурса в паралелната система, освен процесорите, са вътрешно системната комуникационна мрежа и входно – изходната система. Необходимо е да се изследва конкуренцията между процесите за тези ресурси и съответното влияние върху производителността за различните стратегии за планиране.

Друг ключов ресурс за стратегиите за планиране е общата опашка до която осъществяват достъп процесорите за да получат процеси/програми. Конфликтите при едновременни заявки за достъп до тази опашка също могат значително да снижат

производителността. За решаването на този проблем се използват два подхода: автономно планиране и кооперативно планиране (със сътрудничество). При автономното планиране всеки път, когато даден процесор завърши изпълнението на текущия процес, той осъществява достъп до общата опашка и поема за изпълнение определен брой процеси. Този брой всеки път зависи от товара, така че да се осигури равномерно разпределяне на товара. При слабо натоварване на системата процесорът се заема с изпълнението на малък брой процеси, при силно натоварване той взема повече процеси.

При кооперативното планиране един процесор изпълнява функциите по разпределяне на задачите между процесорите. Всеки процесор има своя собствена опашка от задачи. При стартирането на програма с определен брой задачи, планиращият процесор разпределя задачите между процесорите с най-късите опашки като по този начин разпределението се адаптира динамично към системния товар. При слабо натоварване на системата задачите на една програма се разпределят между голям брой процесори, докато при силно натоварена система за всяка програма ще бъдат определени малък брой процесори.

Динамични стратегии

Динамичните стратегии позволяват процесорите да бъдат ангажирани и освобождавани по време на изпълнението на програмата. Ако програмата не използва всички процесори, които са разпределени за нейното изпълнение, някои от тях могат да бъдат освободени и преразпределени за изпълнението на други задачи. Аналогично, ако на дадена програма са й необходими повече процесори от планираните, се прави опит да се преразпределят повече процесори за нейното изпълнение.

Архитектурни аспекти

Най-общият подход за решаването на проблема с разпределянето на задачите между процесорите изисква да се вземат предвид отношенията на предшества между задачите в графа на програмата, броя на процесорите, заети с изпълнението на програмата, както и топологията на вътрешно-системната комуникационна мрежа. Една задача е готова за изпълнение, когато нейните предшественици са изпълнени. Някоя задача не може да бъде изпълнявана на повече от един процесор. Характеристиките на графа на зависимостите на задачите трябва да бъдат известни преди компилацията на програмата. Стратегията, която се използва за планиране на процесорите се основава на два основни критерия: оптимално използване на хардуерните ресурси и минимизиране на разходите за комуникация. Основната цел е намирането на оптимално

съответствие между графа на задачите, и съответния граф на компютърната структура.