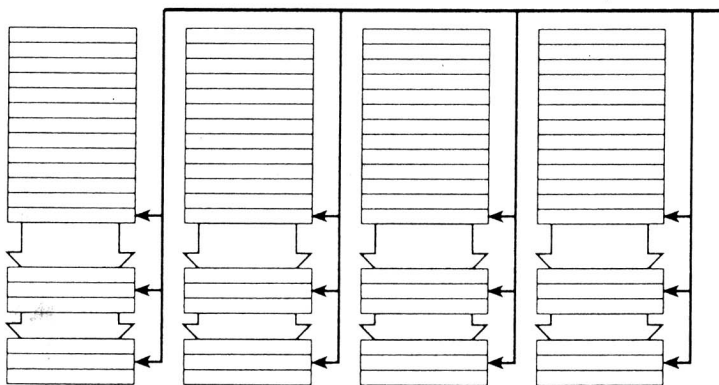


14.1. Bevezetés

Egy percig sem kétséges, hogy a az elmúlt tíz évben, vagy talán még ennél is régebben, az adatpárhuzamos megoldások legsikeresebb megvalósítását a vektoros szuperszámítógépek jelentik. Az első Cray (a Cray-1) 1976-os megjelenésétől kezdve egészen a legújabb több-processzoros rendszerekig, mint például a CM-5, amelyet a 11. fejezetben mutattunk be, a vektoros szuperszámítógépek *kváziszabvánnyá* váltak a nagy teljesítményt igénylő tudományos számítások során. Noha ez nagyobb részt az ilyen rendszerekhez használt modern technológiából ered, a sikerük azonban inkább a vektorozott felépítésből adódó előnyöknek köszönhető. A vektorizálási paradigma fő komponense négy részből áll (14.1. ábra).



14.1. ábra: A vektoros szuperszámítógép elemei

Először is, a paradigma alapvető számítási objektumainál a nagyfokú adatpárhuzamosság forrása a szószervezés. A vektoros szuperszámítógépek jóval hatékonyabban képesek kihasználni a szóhosszt, mint a többi adatpárhuzamos gép a hasonló paraméterüket, és a szavak nagyobbak is lehetnek, mint a hagyományos soros számítógépek esetén. Ez a nagy szóhossz azután visszaköszön az ilyen rendszerek központi számítási architektúrájának minden részében, beleértve a memóriaszervezést és az összes kommunikációs csatornát is.

Másodszor az adatszavak vektorokban helyezkednek el, amelyek szavak százait is tömöríthetik, és ezeket a vezérlésfolyam egyedi adatokként kezeli. Így ezek az adatelemek akár 64 kB-nyi adatot is tartalmazhatnak. Amikor egy utasítás a fő programban végrehajtódik, akkor ez (ideális esetben) mindig ilyen adatmennyiségen hajtódik végre. Ezzel a vezérlés potenciális szűk keresztmetszeteit kiküszöbölték, és így az ilyen architektúrák vezérlőiben kevésbé van szükség RISC-processzorokra.

Harmadszor, a futószalagos feldolgozást a maga teljességében alkalmazzák, mind mikroszinten az egyes műveleti elemeknél, mind makroszinten több műveleti elem esetén.

Ezen kívül a futószalagelvet alkalmazzák az adat ki- és bevétel során az előbbiekből ismertetett adatvektorok esetén mindig, amikor ez alkalmazható és lehetséges.

Negyedszer, a műveleti elemek párhuzamosságát azonos és eltérő módon is alkalmazzák, ahogy szükséges. Így a vektorok egymással párhuzamos árama is feldolgozható, például több lebegőpontos összeadó vagy szorzó, és több (nagyon hosszú) egész típusú műveleti egység segítségével.

Noha ezeknek a technikáknak az egyenkénti alkalmazásával nyert sebességnövekedés szerény a párhuzamos számítógépek kategóriájában (talán egy nagyságrend minden elemre nézve, ellentétben a SIMD-paradigma tízezerszeres tényezőjével), de ha a négy tényezőt együttesen alkalmazzuk, akkor a leghatékonyabb, együttes sebességnövekedés ezerszeres, vagy még több is lehet az egyes műveleti elemekre nézve. Meg kell azonban jegyezni, hogy ez az előnyös helyzet nem magától alakul ki. A megfelelő alkalmazások kiválasztása, és a programok, illetve az adatok megfelelő formátumú összeállítása nem mindig egyértelmű feladat. Bár e a feladatok legnagyobb részét a párhuzamos fordítók átveszik, ez csak akkor lehet eredményes, ha mindenek előtt az alkalmazást gondosan választjuk ki. Ezen túlmenően, noha egy ilyen rendszer csúcsteljesítménye 10 GFlop is lehet, általában azonban az átlagos teljesítmény ennél egy nagyságrenddel kevesebb – 1 GFlop, épp úgy, mint a CM5-típusú számítógépek esetében.

A könyv ezen részében tárgyalt vektoros szuperszámítógépek a számítógép-architektúra szempontjából messze a legbonyolultabb adatpárhuzamos rendszerek. Ez természetes következménye annak a bonyolult technikának, amelyet a nagy teljesítmény elérése érdekében használnak, de ez azt is jelenti, hogy felépítésük megértése ennek megfelelően, nehéz. A következő részben megpróbáljuk a különböző jellemzőket, amennyire csak lehet, különválasztani, de a kölcsönhatások elkerülhetetlenek. Az Olvasónak azt javasoljuk, hogy próbálja meg teljesen megérteni a rendszer felépítésének minden jellemzőjét, mielőtt a két példát áttekinti a fejezet végén.

A vektoros szuperszámítógépek egyes jellemzőit különválasztva tárgyaljuk. Ezen túlmenően, beiktattunk egy rövid ismertetőt az ilyen rendszerek technológiájáról is, mivel ez egy lényeges elem a tervezésük során.

14.2. Szóhossz

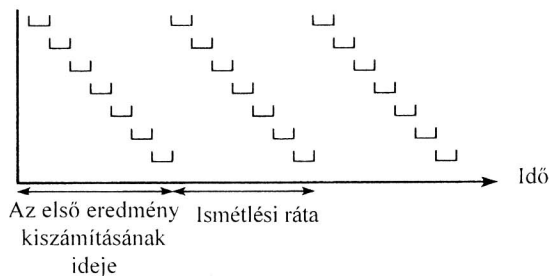
Két fő mérlegelési szempont van, ami befolyásolja ennek a jellemzőnek a tervezését. Az első nem is annyira technikai, mint inkább szabványosítási kérdés. A programok nagy részét lebegőpontos műveletek alkotják, és az ezekhez használt számok alakja különböző lehet. Az egyik magától értetődő stratégia lehet a vektoros számítógépek hosszának kiválasztása során, hogy szabványos hosszt választunk. Két alapvető szóhosszúság-szabvány létezik az IEEE ajánlások alapján, de sajnos számos gyártó a szabványokon belül saját konfigurációt valósított meg. Lehetőség van ennél hosszabb szavak megalkotására is közvetlen módon, vagy két rövidebb szó összefűzésével, ami növeli a rendszer rugalmasságát. Végül nyilvánvalóan ezeknél nagyobb szavak is kialakíthatók, még megfelelő előírások hiányában is. A jellemző tervezési tere a 14.2. ábrán látható.

14.4. Futószalagos adatkezelés

A futószalagos adatkezelés, az 5. fejezet szerint, funkciópárhuzamos megoldás. Az adatpárhuzamos vektoros szuperszámítógépek esetében azonban annyira fontos szerepe van, hogy szükséges bemutatni, hogy ez a két jellemző milyen kölcsönhatásban van egymással. Az alapelveket ismét csak az idődiagramon lehet a legjobban bemutatni. A 14.6. ábrán egy vektorizált függvény két lehetséges megvalósítása látható, az első hagyományos megoldású, míg a második teljes mértékben alkalmazta a futószalagot.

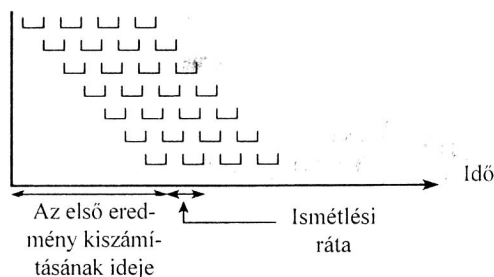
Az első cím kiszámítása
Az első adat betöltése
A második cím kiszámítása
A második adat betöltése
Az eredmény kiszámítása
A célmemória címének betöltése

(a)



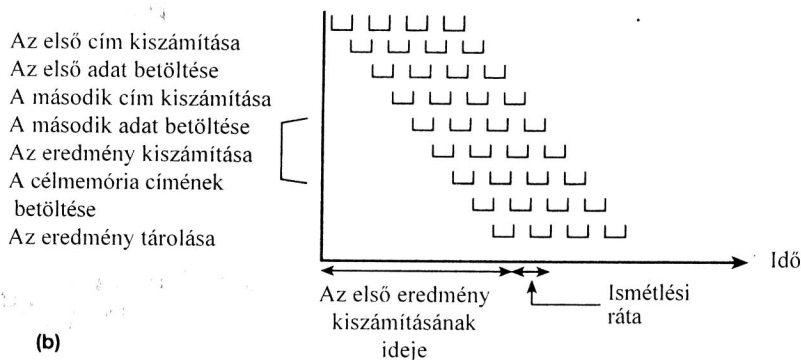
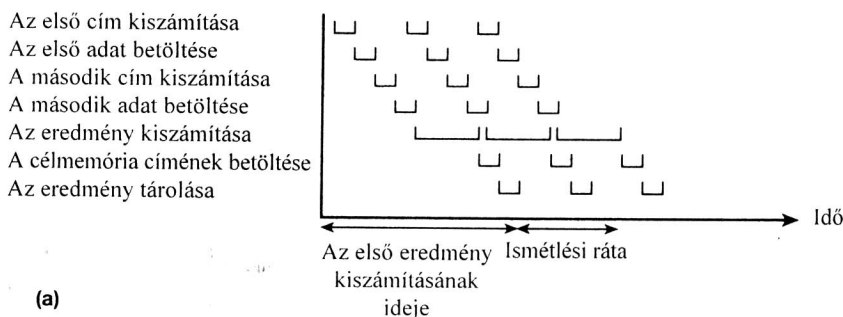
Az első cím kiszámítása
Az első adat betöltése
A második cím kiszámítása
A második adat betöltése
Az eredmény kiszámítása
A célmemória címének betöltése

(b)



14.6. ábra: Sebességnövelés futószalag segítségével.

A teljesítmény legnagyobb mértékben az olyan adatfolyam ismétlési rátája révén növelhető, amelynek nem kell várni az első kiszámított eredményre. Valójában az első eredmény megkapásának ideje mind a két rendszerben ugyanannyi. Mivel a futószalagos rendszerekben az egymást követő eredményeket ugyanabban az ütemben lehet kiszámíttatni, mint ahogy azok belépnek a rendszerbe – ideális esetben minden órajelre kaphatunk egy eredményt. Természetesen a legnagyobb teljesítmény elérése függ a futószalag pontos tervezésétől. Valószínűnek tűnik azonban (sőt majdnem bizonyos), hogy az elérhető maximális ismétlési ráta a futószalag egyes elemei esetében nem azonos. Ha ez így van, akkor háromféle megoldás alkalmazható. Először is lehetőség van arra, hogy a futószalag szemcsézettségének finomságát növeljük azzal, hogy a leglassúbb elemeket részműveletekre bontjuk. Másodszor, további tervezési megfontolásokkal, vagy csúcstechnológia alkalmazásával meggyorsíthatjuk a leglassúbb elemeket. És végül, az egész futószalagot a leglassúbb elem sebességének megfelelően működtetjük.



14.7. ábra: A futószalag szemcsézettségének növelése

A futószalag tervezése során a különböző technológiák felhasználásából adódó változásokat nehéz elemezni, hiszen a technológia javulása lépcsőzetes folyamat. A futószalag szemcsézettségének növekedését azonban lehet ábrázolni, ahogy az a 14.7. ábrán látható.

A meglévő futószalagon végzett műveletek minden szintjét, építőelemeit elemezni kell, és szét kell őket választani. A legfelső szinten valójában ez a hagyományos rendszerek futószalagos rendszerré való átalakításának első lépése.

14.5. Párhuzamos számítási folyamatok

A vektoros számítógépek negyedik szerkezeti eleme a funkciópárhuzamos, az adatpárhuzamos, vagy a kettőt egyesítő megoldás, amely az adott feladattól, valamint a rendszerben használt adatok mennyiségétől és típusától függ. Ennek az elvnek a legegyszerűbb megvalósítását mutatja be a 14.8. ábra. A rendszerben különálló elemek találhatók a lebegőpontos és fixpontos számításokhoz. Ezek a megfelelő adatok felhasználásával párhuzamosan működhetnek, ha azok megjelennek a számítási feladat megfelelő helyein, és – amennyiben a megfelelő átlagok rendelkezésre állnak (akár fizikailag, akár előzetes számítások eredményeként) – megfelelő időben megkapják a helyes adatokat. A diagramban a fizikai átlagok szerepelnek pufferek formájában, és ezen a ponton talán könnyű bemutatni, hogy a vektorizálás hogyan alkalmazható, amikor ugyanazt a regisztert mindkét feladatra felhasználjuk.