Memóriaelérés Optimalizálása, Irreguláris Gráfokon Értelmezett Számítások Gyorsítása

A Ph.D. Disszertáció tézisei

Hiba Antal M.Sc.

Témavezetők:

Dr. Szolgay Péter , Dr. Ruszinkó Miklós

Pázmány Péter Katolikus Egyetem

Információs Technológiai és Bionikai Kar

Roska Tamás Műszaki és Természettudományi Doktori Iskola

Budapest, 2015

Célkitűzések

Az újabb processzor architektúrák számos párhuzamos processzor magot tartalmaznak. A nagy számítási teljesítményű sokprocesszoros chipek széles körben elterjedtek, a tudományos és ipari szuperszámítógépek is ilyen csomópontokból épülnek fel. Az elméleti maximális számítási teljesítmény átlépte az 1 TFLOPS/chip értéket, viszont az alkalmazások többségében a chipek kihasználtsága mindössze 10-15% között mozog. Ennek elsődleges oka a külső memóriák sávszélessége és a processzortömbök számítási teljesítménye közötti különbség. A számítás sebességét már nem a műveletvégző egységek határozzák meg, hanem a memória interfészek áteresztő képessége, így ezek egyfajta fal képződik az off-chip memória és a processzortömb között [R1]. A hatás erősödik, mivel a számítási teljesítmény gyorsabban növekszik, mint a memória sávszélesség. Számos kutatás tűzi ki célul a memória fal (memory wall) lebontását [R2], melyek közül véleményem szerint a memória és a processzortömb integrációja 3D chip technológia segítségével a legígéretesebb [R3, R4]. A sebesség korlátozása mellett a fogyasztásból is egyre jelentősebb részt tesz ki a kommunikáció (chipen belül is), ami környezetvédelmi (green computing) és anyagi szempontból is fontossá teszi a memória interfészekkel kapcsolatos kérdéseket.

Memóriaelérés optimalizálása alatt egyrészt a processzorok lokális (cache) memóriakezelő eljárásait értjük, amelyek célja a lassú külső (off-chip) memóriaelérések minimalizálása, valamint azon módszereket, amelyek a rendelkezésre álló memória-sávszélesség hatékony kihasználását támogatják. A memóriaelérések számát az adatok cache memóriában való ideiglenes tárolásával csökkenthetjük, abban az esetben, ha egy adatot többször is fel kell használni a számítás során. A rendelkezésre álló off-chip memória-sávszélesség hatékony kihasználását pedig megfelelő olvasási mintázatok alkalmazásával érhetjük el. A jelenleg használatos DRAM technológiák számára a soros olvasási mintázat a legmegfelelőbb, ezért adatainkat érdemes úgy szervezni az off-chip memóriában, hogy a számítások során a szükséges adatok egymás után sorban helyezkedjenek el. A helyzet még összetettebbé válik, ha több processzortömb (chip) között kell egy számítási feladatot felosztani. Ebben az esetben megjelenik a processzorok közötti kommunikáció is, amit szintén minimalizálnunk kell a hatékonyság növelése érdekében.

A klasszikus megközelítés egy nagyméretű számítás csomópontok közötti felosztásánál egyedül a csomópontok közötti kommunikáció minimalizálását, és a feladatok egyenletes elosztását tűzi ki célul. A számítási feladatot elemi részfeladatokra bontva egy gráfot kapunk, amelyben a csúcsok reprezentálják a feladatokat, az élek pedig a feladatok közötti adatfüggéseket (kommunikáció). Ezzel eljutottunk a gráf particionálási feladathoz, amely azonos méretű részgráfok létrehozását tűzi ki célul minimális élvágással. A memóriaelérés optimalizációja a feladatok és a hozzájuk tartozó adatok sorbarendezésével történik a particionálást követően (adatlokalitás növelése). Ez a feladat a Mátrix Sávszélesség Minimalizáció problémára vezethető vissza, ahol a gráf szomszédsági mátrixát hozzuk sávmátrix alakra. Sajnos mindkét probléma NP-teljes [R5, R6], így optimális megoldás megtalálása polinom időben nem lehetséges, feltéve hogy P!=NP. Sok nagyszerű kutató munkájának köszönhetően mindkét problémára léteznek hatékony heurisztikus eljárások. Mivel az adatlokalitás egyre fontosabb tényezővé válik, felmerül a kérdés, hogy nem kellene-e már a particionálás fázisában figyelembe vennünk.

A disszertációm elsődleges célja, hogy feltárjam a gráf particionálás és az elérhető adatlokalitás közötti összefüggéseket, és ezek segítségével módszereket adjak a két optimalizációs cél együttes kezelésére. Egy ilyen módszer biztosíthatja a processzortömbök hatékonyabb kihasználását, az erőforrások pazarlásának elkerülését.

Láthatjuk, hogy a számítások gyorsításához összetett optimalizációs feladatokat kell megoldanunk. Evidens, hogy nem fordíthatunk sok időt és energiát magára az optimalizációra. Korunk hatékony algoritmusait szokás metaheurisztikáknak hívni, ami arra utal, hogy nem csupán feladat-specifikus ügyes módszerekről van szó, hanem valami 'többről'. Az optimalizációs probléma lehetséges megoldásai kifeszítenek egy keresési teret. A metaheurisztikus módszerek a dimenzió csökkentést (multi-level) vagy a keresési térben való mozgási stratégiát (gradiens, változó szomszédsági keresés, szimulált lehűtés, genetikus algoritmus, stb.) írják le feladattól függetlenül [R7, R8]. A gráf particionálás esetén a dimenziócsökkentéses módszerek váltak egyeduralkodóvá [R9], miközben a mátrix sávszélesség minimalizációnál hatékonynak bizonyultak a specifikus nem metaheurisztika alapú módszerek [R10, R11].

A disszertációm második felének célja, a metaheurisztikus módszerek dimenziócsökkentéses gyorsításának vizsgálata, új lehetőségek keresése, nem kizárólag particionálási problémák kezelésére.

Vizsgálati Módszerek

Kísérleteim többségét c++ nyelven megírt saját keretrendszerekben végeztem, melyhez a Microsoft Visual Studio 2008/2012 fejlesztőrendszert használtam, ezen kívül még Matlab 2008-ban dolgoztam. A grafikus felületeket Windows Forms technológiával oldottam meg. A külső nem saját készítésű programokkal szabványos file formátumokon (.msh) keresztül került átvitelre az információ. A particionáláshoz használt mintahálókat a GMSH [R12] ingyenes hálógeneráló programmal készítettem, valamint az eredmények vizualizációjához is ezt a programot használtam fel.

Az elérhető fontosabb particionálási csomagok METIS, CHACO, SCOTCH, JOSTLE, PARTY [R13] közül a METIS [R14]-t és a SCOTCH [R15]-ot választottam ki mélyebb analízisre. Eredményeimet a METIS eredményeivel hasonlítottam össze.

Az irreguláris hálókra megadott adatfolyam architektúrát az AlphaData ADM-XRC-6T1 újrakonfigurálható fejlesztő rendszerén vizsgáltam. További memória interfész tesztekhez egy Xilinx Zedboard fejlesztő kártyát használtam, amelyen egy Xilinx Zinq SoC processzor található. Ez a környezet beágyazott FPGA-t is tartalmaz, amelyet a Vivado HLS 14.4 eszközeivel programoztam.

Az általános dimenziócsökkentő eljárás teszteléséhez két tudományos mintahalmazt használtam fel, a TSPlib [R16]-et és az SOPlib [R17]-et. Eredményeimet a leghatékonyabb ismert heurisztikákkal vetettem össze, köztük a DPSO-val [R18], amely a legjobb az említett két teszthalmazon.

Sajnos csak speciális esetekben volt lehetőségem nem empirikus vizsgálatokra. Itt főként egyszerűbb gráfokkal kapcsolatos összefüggések belátásáról volt szó.

Új tudományos eredmények

1. Téziscsoport: Memória-elérés és processzorközi kommunikáció együttes kezelése rácsokon értelmezett számítások felosztásakor.

Kapcsolódó publikációk: [J1, C1, C2, C3, C4]

Az irodalomban fellelhető particionálási módszerek célfüggvényeiben nem jelenik meg közvetlenül a létrejövő részgráfok sávszélessége (adatlokalitása), ami meghatározó a memória-elérés szempontjából. A programcsomagok a particionálást követően adnak lehetőséget a lokalitás növelésére, a létrejött részgráfokon belül. A létrejövő részhálók azonban meghatározzák az elérhető adatlokalitást, ezért a particionálás nagy hatással van az adatlokalitásra.

Célom az volt, hogy az adatlokalitásra korlátot lehessen meghatározni, amely korlátot a sokprocesszoros architektúra fizikai paraméterei (On-chip memória mérete) határoznak meg. A korlát szükségessége az FPGA-n megvalósított adatfolyam processzorok (Dataflow Machines) használata során merült fel, amelyek mind sebességben, mind pedig energiahatékonyságban a legjobb architektúráknak bizonyultak parciális differenciál-egyenletek explicit numerikus megoldása során [R19, J1]. Az adatfolyam processzorok sikerének kulcsa a memória interfész és az on-chip cache optimális kihasználása, viszont ehhez garantálni kell egy adott adatlokalitást. Mivel a korlát bevezetése megváltoztatta a létrejövő felosztást, várható volt a processzorközi kommunikáció növekedése. Fontossá vált a két optimalizálási cél közötti összefüggések vizsgálata.

1.1. Megmutattam, hogy a processzorközi kommunikáció minimalizálása és az adatlokalitás maximalizálása egymással ellentétes célok, ezért az adatlokalitást is figyelembe kell venni a particionáláskor. Kísérletileg igazoltam, hogy minimális processzorközi kommunikáció-növekedés (<1%) árán az adatlokalitás jelentősen növelhető (>30%).

Egy gráf sávszélesség igényének meghatározásához meg kell adni a hozzá tartozó optimális sávszélességű rendezést. Ez NP-teljes feladat, így ennek előállítására nincs hatékony

eszközünk. Empirikus és elméleti vizsgálatok során viszont sikerült meghatároznom pár összefüggést, amelyek segítségével két gráf sávszélesség igénye becslés szinten könnyen összehasonlítható. Az alapvető összefüggések már rég ismertek voltak, mint a gráf 'átmérője' és a sávszélesség közötti összefüggés, melyen a CM [R10] és GPS [R11] újrarendező módszerek is alapszanak. A processzorközi kommunikáció oldaláról ismert volt, hogy a jó felosztásokban gömbszerű részrácsoknak kell létrejönni, hiszen ezeknek lesz a legkisebb a felülete (élvágása). Én csupán összekötöttem és kibontottam a két terület eredményeit.

Az adatlokalitás gyakorlatilag tetszőleges mértékben javítható a processzorközi kommunikáció rovására. Ha több processzorunk van, mint ahány színnel a gráf színezhető, és a színosztályok szerint osztjuk fel a rácsot, minden adatfüggés processzorközi kommunikációt fog jelenteni. Ezzel szemben a processzorközi kommunikáció nem csökkenthető le egy korlát alá (minimális élvágású felosztás), csak a processzorszám csökkentésével. Itt a szélsőséges eset, amikor 1 processzor van és nincs processzorközi kommunikáció. A mérési eredmények alátámasztották, hogy a particionálás nagy hatással van az adatlokalitásra. Több példarács esetén is elég volt csupán 0,002 kommunikációs arány[1] növelés 30-40%-os adatlokalitás javuláshoz.

1.2. Kísérletileg igazoltam, hogy ha a rács határpontjainak halmaza ismert, akkor az ebből indított szélességi keresési fa legmélyebb szintjein található pontok határozzák meg azt a kritikus mély régiót, amelyet a szeparátoroknak át kell vágniuk a jobb adatlokalitás eléréséhez. Olyan módszert adtam meg, amely ezen régiók vágását kihasználva akár 30-40%-al jobb adatlokalitás elérésére is képes biparticionálás esetén, mint a METIS a processzorközi kommunikáció növekedése árán.

A legtöbb gyakorlati esetben (PDE numerikus megoldásához használt térbeli rácsok), ismert a határpontok halmaza (peremfeltételek). Ha ebből a halmazból indítunk egy szélességi keresést, akkor a létrejövő keresési fa meghatároz egy mélységi szintstruktúrát (DLS), ahol a legmélyebb szintekhez tartozó csúcspontok a lehető legtávolabb vannak a határpontoktól. A DLS struktúra megmutatja, hogy hol vanak a rácsban azok a

[1] a kimenő élek és a belső pontok hányadosa

mély régiók, amelyek a legnagyobb befoglalt 'gömbök' középpontjait határozzák meg. A lokalitás szempontjából legjobb szeparátoroknak ezeket a gömböket kell kettévágni.
A partíciók sávszélesség igényének gyakorlati becslésére a Gibbs Pole Stockmeyer (GPS) módszert alkalmaztam, amely egy hatékony és olcsón számítható heurisztika.

1.3. Megadtam a gráfparticionálási feladat egy olyan kiterjesztését (Korlátos Sávszélességigényű Particionálás - BLP), amely tartalmazza a processzorközi kommunikáció és az adatlokalitás együttes kezelését, továbbá optimalizálja a felhasználandó processzorok számát. Igazoltam hogy a BLP feladat megoldásai jobbak az FPGA alapú adatfolyam processzorok számára, mint a klasszikus feladat megoldásai.

Esetünkben a gráf particionálás gyakorlati célja, hogy megadja a gráf által reprezentált feladatok egy olyan felosztását fizikai processzorok között, ami minimális futásidőt eredményez. Az alapfeladat által kitűzött egyenletes feladat felosztás és minimális élvágás mellett több más tényezőt is érdemes figyelembe venni.

1. Definíció (Sávszélesség-Korlátos Particionálás). *Adott egy gráf $G(V, E)$, V ($|V| = n$) csúcshalmazzal és E élhalmazzal. BW_Bound, $COMM\_Bound$ és K szintén adottak. Keressük $Q = \{P_1, P_2, .., P_k\}$ felosztást a lehető legnagyobb k-val melyre a következő feltételekkel:*
$Out(P_i)$ jelöli P_i részgráf kimenő éleinek halmazát.

$$k \leq K \tag{1}$$

$$\max_i \left\{ \frac{|Out(P_i)|}{|P_i|} \right\} \leq COMM\_Bound \tag{2}$$

$$\max_i \{2 \cdot B_{f_i}(P_i) + 1\} \leq BW\_Bound \tag{3}$$

$$|P_i| \approx \frac{n}{k} \tag{4}$$

A processzorközi kommunikációra és az adatlokalitásra megadott korlátok (2)(3) biztosítják a kívánt magas processzor kihasználtságot [C1, C4].

A korlátok bevezetése miatt előfordulhat, hogy nincs illeszkedő megoldása a feladatnak. Ezekben az estekben valamelyik korlátot egy magasabb értékre kell állítani, mivel a feltételek nem teljesíthetőek.

A kommunikációs arány $COMM\_R$ pont azt a vonalat határozza meg, ahol a processzorközi kommunikáció jelentősége megegyezik a belső kommunikációval. $COMM\_R$ a processzorok számát is korlátozza, ugyanis több processzor esetén a részhálók processzorközi kommunikációs szükséglete növekszik.

1.4. Kidolgoztam egy Gibbs Pole Stochmeyer (GPS) módszeren alapúló algoritmust (AM1), amely sávszélességkorlátos partícionálást hajt végre a részgráfok számának optimalizálása nélkül (részleges BLP).

A módszer lényegében a GPS sorszámozási mechanizmusát egészíti ki egy pontos sávszélesség igény előrejelzéssel. Ha a megsorszámozott részgráf sávszélesség igénye meghaladná a korlátot, új részgráfot kezd, amíg fel nem osztja a teljes bemeneti gráfot [J1].

1.5. Megadtam a teljes BLP feladat közel optimális megoldását 2-3D téglalap tartományú strukturált rácsokra, rácstípusú BLP felosztások segítségével.

A strukturált téglalap rácsokon végzett kutatások segítettek a BLP feladat nehézségeinek és hasznának megértésében.

Egy $a \times b$ háló rácstípusú felosztása egy $g_a \times g_b$ rács, amely azonos $\frac{a}{g_a} \times \frac{b}{g_b}$ részrácsokból áll. Az ilyen rácstípusú felosztások esetén a BLP feladat egyenlőtlenségei egyszerűen ellenőrizhető alakot öltenek (5), ahol $s_a = \left\lfloor \frac{a}{g_a} \right\rfloor$, $s_b = \left\lfloor \frac{b}{g_b} \right\rfloor$ és $S_a = \left\lceil \frac{a}{g_a} \right\rceil$, $S_b = \left\lceil \frac{b}{g_b} \right\rceil$ jelölik a lehetséges oldalszélességeket. $|Out(P_i)|$ meghatározásakor csak a részrácsok közötti felületeket vesszük figyelembe ezért az egyes oldalak szorzói $\{0, 1, 2\}$. m_a, m_b jelöli a partícióhoz tartozó maximális szorzókat.

$$\begin{aligned} 2 \cdot \min\{S_a, S_b\} + 1 &\leq BW\_Bound \\ \frac{m_a \cdot s_a + m_b \cdot s_b}{s_a \cdot s_b} &\leq COMM\_Bound \end{aligned} \tag{5}$$

Mivel **ga x gb K**-nál kisebb kell legyen, a lehetséges felosztások száma (6) szerint alakul, ami alkalmasan kevés ahhoz, hogy kimerítő kereséssel kiválasszuk közülük a legjobbat. A

módszer még 3D esetben is elég gyors marad mivel ott **ga x gb x gc** kell hogy kisebb legyen mint **K**, ami csak még egy harmonikus sor komponenst ad (6)-hoz.

$$K - \left\lfloor \frac{K}{2} \right\rfloor + \sum_{i=1}^{\lfloor K/2 \rfloor} \left\lfloor \frac{K}{i} \right\rfloor \tag{6}$$

A legjobb rácstípusú megoldás nem feltétlenül optimális. Van ellenpélda, ahol **ga** vagy **gb** nem osztója **a**-nak illetve **b**-nek, és létezik jobb megoldás.

1.6. Kidolgoztam egy METIS-AM1 hybrid módszert irreguláris rácsok BLP particionálására.

Mivel a processzorközi kommunikáció nem csökkenthető csak a processzorszám csökkentésével, nem meglepő hogy ez ad egy felső korlátot a processzorszámra. A processzorközi kommunikációt kezelő módszerekben már évtizedes kutatómunka van. Ha egy k-utas METIS partíció nem teljesíti a processzorközi kommunikációra megszabott korlátot, akkor arra rendkívül kevés az esély, hogy létezik k+1-utas partíció, amely teljesíteni tudja. A METIS-AM1 hybrid módszer lényege, hogy METIS-el meghatározzuk **k**-t. Rekurzív biszekcióval vágjuk a rácsot, amíg még teljesül a korlát, majd a két utolsó lépés között intervallumfelezés módszerével meghatározzuk a legnagyobb **k**-t, amelyre még teljesül a korlát. Az így kapott k-utas METIS partíció minden részrácsára meghívjuk az AM1 algoritmust.

2. Téziscsoport: Valósidejű Kombinatorikus Optimalizáció Megvalósítása Felhasználható Részmegoldások Előállításával.

Kapcsolódó publikációk: [C5]

Egy optimalizálási feladat megoldási ideje nagyban függ a döntési változók számától, azaz a megoldások által kifeszített tér dimenziójától.

Az alkalmazható részmegoldások generálása (APSG) egy új lehetőség a dimenzió csökkentésére. Egy részmegoldás mint részleges kimenet készül el kötött idő alatt, és a részmegoldás felhasználása közben a fennmaradó rész optimalizálása folytatódik. A részmegoldás előállításának ideje határozza meg a válaszidőt mert a megoldás hasznosítása ez után már megkezdődhet. Mivel számos CO probléma $\mathcal{NP}$-nehéz, a leghatékonyabb

megoldók heurisztikák, amelyek egy exponenciálisan növekvő megoldási térben végeznek keresést. A megoldások minősége függ a rendelkezésre álló időtől, emiatt a válaszidő javítása és a megoldás minősége csak egymás rovására javítható. Az alkalmazható részmegoldások generálása egy jobb átváltást biztosít az optimalizálásra fordított idő és a megoldások minősége között, mert lehetővé teszi, hogy csak egy részmegoldás előállítását követeljük meg adott idő alatt, ahelyett hogy a teljes optimalizációt meg kellene szakítanunk a rendelkezésre álló idő leteltekor. A több-szintes módszerek képesek megfelelően gyors particionálást biztosítani, ennek a tézis csoportnak az eredményeit nem használtam a gráf particionálás gyorsítására, viszont más CO alkalmazások esetén hasznos.

2.1 Megadtam a felhasználható részmegoldások generálásának egy metaheurisztikus modelljét (Részmegoldás Összefésülés - VSM). A soros rendezési problémán (Sequential Ordering Problem - SOP) a lemez ütemezési problémán (Disk Scheduling Problem - DSP) és az általános hozzárendelési problémán (Generalized Assignment Problem - GAP) keresztül bemutattam annak alkalmazhatóságát és korlátait. Megadtam a legfontosabb választható alacsonyabb rendű heurisztikákat, és egy hibridizációs módot, amely segítségével az eddig használt valósidejű megoldók eredményei garantáltan javíthatóak.

A Részmegoldás Összefésülés (VSM - Variable Subset Merger) több részmegoldás kiegészítését írja le, ahogy ezek egyesülve egy teljes megoldást alakítanak ki. A VSM folyamat közben minden egyes részmegoldás egy szabad döntési változót tartalmaz a maradék döntési változó értéke már rögzített.

A VSM megközelítés olyan optimalizációs problémára alkalmazható, ahol egy részmegoldásnak van hasznosítási lehetősége és az optimalizációs eljárás válaszideje lényeges. Ha hatékonyan nem igazolható egy illeszkedő teljes megoldás létezése (GAP), a megkötéseket gyenge megkötésekként tudjuk csak kezelni. Minden más esetben validálni tudjuk a részmegoldásokat biztosítva egy illeszkedő teljes megoldás létrejöttét (SOP). A köztes megoldások validációja az egyetlen lényegi korlátozó tényező az alkalmazható részmegoldások generálásában, és ez a korlát független a VSM-től.

Az SOPLIB mintahalmazon elért eredmények azt bizonyítják, hogy a VSM használata lehetővé teszi a valósidejű választ még nagy problémák esetén is.

A metaheurisztikus megadás megkönnyíti a hibridizációt a legjobb elérhető valósidejű és nem valósidejű megoldókkal. A legjobb valósidejű módszerek megoldásai tovább javíthatóak azonos válaszidő mellett, és a VSM keret lehetővé teszi nem valósidejű megoldók alkalmazását valósidejű válasszal.

2.2 Kísérletileg igazoltam, hogy a részmegoldás kiválasztás lehetősége javítja a konstruktív optimalizációs módszerek megoldásait. A TSPLIB (n=71..380) mintahalmazon 7,5%-al míg az SOPLIB-en (n=200..700) 26%-al jobb megoldások érhetőek el a VSM-PLoss kiválasztási stratégiával a VSM-Const kiválasztás nélküli esethez képest.

Az SOP megoldókkal végzett kísérletek megerősítették, hogy a részhalmaz kiválasztó heurisztika növeli a részhalmaz megoldó heurisztika hatékonyságát. Az SOPLIB mintahalmaz példáinak megoldásai azt mutatják, hogy a VSM-Ploss részhalmaz kiválasztó heurisztika (VSM-SOP) jóbb megoldásokat eredményez, mint a VSM-Const. A VSM-SOP 26%-al jobb megoldásokat eredményezett ezeken a nagyobb méretű példákon. A VSM koncepcióban célom olyan részhalmaz kiválasztó stratégiák megadása volt, amelyek nem vezetnek rossz minőségű teljes megoldásokhoz. A VSM-SOP algoritmus azt bizonyítja, hogy hatékonyabb heurisztikák definiálhatóak VSM alapon az alkalmazható részmegoldások generálására.

2.3 Kísérletileg igazoltam, hogy cselekvési sorok időköltségének optimalizációjakor kapott SOP feladat egy VSM-alapú megoldóval oldható meg a leghatékonyabban, ha az optimalizációra szánt időt is költségnek tekintjük, és egy elemi cselekvés időköltsége relatíve kicsi. Ez 1-20 mp-et jelent az SOPLIB megkötéseket tartalmazó példáira, a DPSO-t választva referenciaként.

A VSM elsődleges feladata hogy gyorsan alkalmazható részmegoldásokat generáljon egy hibrid optimalizálóban, viszont vannak szituációk, amikor önmagában is képes a legjobb teljes megoldás előállítására. Ha a célfüggvény időigényt határoz meg, az optimalizációra fordított idő egy additív költséget jelent. A legnagyobb 700 változós SOPLIB példán a VSM-SOP teljes megoldása mindössze 5%-al rosszabb, mint a 364 másodperc alatt a DPSO által szolgáltatott megoldás. Feltéve hogy a DPSO megoldásának költsége 700 mp, ez egy 735 mp-es VSM-SOP megoldást jelent, miközben a válaszidő 320-mp-el rövidebb, ami azt jelenti, hogy az összesített időszükséglet 285 mp-el kisebb a VSM-SOP esetén. Ez 26%-os sebességnövekedést és kilencszeres válaszidő javulást jelent.

$$(DPSO\_resp1 - VSM\_SOP\_resp1) = (VSM\_SOP - DPSO) \cdot C$$
$$oper\_avg = \frac{1}{n} \cdot C \cdot DPSO \tag{7}$$

Tegyük fel, hogy az SOPLIB mintafeladatai időminimalizálási problémákat írnak le.

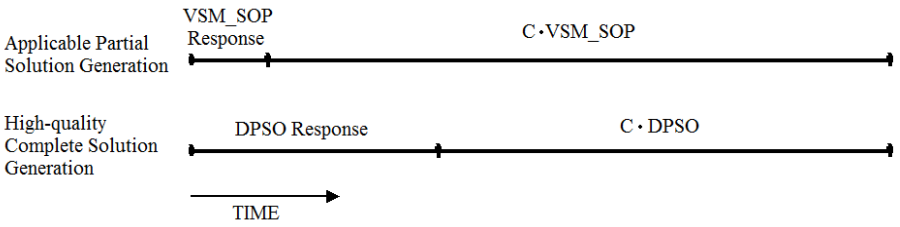

Figure 1. Az optimalizációra fordított idő és a megoldás időszükségletének együttes kezelésének összevetése az SOPLIB mintafeladatain. A C időátváltó konstanst úgy adjuk meg, hogy a két vizsgált módszer összesített időszükséglete megegyezzen.

Legyen C az átváltás az SOPLIB megoldások költsége és az idő között. A (7) egyenletben C-t úgy választjuk, hogy a teljes időszükséglet a VSM-SOP és DPSO esetben megegyezzen (Fig. 1). C segítségével megkapjuk az egy változóra jutó átlagos időköltséget. Ha az egy változóra jutó időköltség kisebb, mint az előbbi átlag, akkor a VSM-SOP jobb összesített időköltéget ér el.

Alkalmazási területek

Az első téziscsoport eredményei az adatfolyam architektúrák alkalmazását támogatják hálókon értelmezett számítások esetén. Az AM1 algoritmus korlátozott adatlokalitású elérési mintázatokat generál. Az optimalizált korlátozott elérési mintázatok nélkülözhetetlenek az adatfolyam architektúrák számára és lehetővé teszik számukra a nagyobb méretű hálók kezelését. Az AM1 növeli az 1-chip-es adatfolyam architektúrák alkalmazhatóságát. Az első téziscsoport második fele sávszélesség-korlátos particionálási technikákkal foglalkozik. Ismertek voltak már több-chip-es adatfolyam architektúrák strukturált hálókra, viszont a hozzájuk tartozó particionálási feladat definiálása (BLP) és a nem strukturált esetre kidolgozott megoldók eddig nem voltak adottak.

A BLP particionálás nélkülözhetetlen az adatfolyam architektúrák számára, viszont más

architektúrák esetén is fontossá válhat, ha az egy chip-re jutó részháló mérete elég nagy (>300k csomópont). Kis részhálók esetén a processzorközi kommunikáció fontosabb, mint az adatlokalitás. A újabb processzorok azonban egyre több és több számítási kapacitással és off-chip DRAM-al rendelkeznek, amely trend a BLP particionálást más architektúrák számára is ajánlottá teheti. Az első téziscsoport eredményei felhasználhatóak az optimális processzorszám meghatározására a particionálás előtt, az erőforrások pazarlásának elkerülése érdekében.

A második téziscsoport módszereket ad a válaszidő javítására kombinatorikus optimalizáció esetén felhasználható részmegoldások generálásával. A VSM megközelítés olyan optimalizációs problémára alkalmazható, ahol egy részmegoldásnak van hasznosítási lehetősége és az optimalizációs eljárás válaszideje lényeges. A metaheurisztikus megadás megkönnyíti a hibridizációt a legjobb elérhető valósidejű és nem valósidejű megoldókkal. A legjobb valósidejű módszerek megoldásai tovább javíthatóak azonos válaszidő mellett, és a VSM keret lehetővé teszi nem valósidejű megoldók alkalmazását valósidejű válasszal. A módszer hibridizáció nélkül is hatékonynak bizonyult feladat ütemezés esetén, ha több száz rövid (1-20 mp) feladat adott sorrendi megkötésekkel.

Referenciák

A szerző folyóirat publikációi

[J1] Nagy, Z. Nemes, C. **Hiba, A.** Csík, Á. Kiss, A. Ruszinkó, M. Szolgay, P. "Accelerating unstructured finite volume computations on field-programmable gate arrays". In: *Concurrency and Computation: Practice and Experience* 26.3 (2014), pp. 615–643.

[J2] Zsedrovits, T. Bauer, P. **Hiba, A.** Nemeth, M. Pencz, B. J. M. Zarandy, A. Vanek, B. Bokor, J. "Performance Analysis of Camera Rotation Estimation Algorithms in Multi-Sensor Fusion for Unmanned Aircraft Attitude Estimation". In: *Journal of Intelligent & Robotic Systems* (2016), pp. 1–19.

[J3] Zsedrovits, T. Bauer, P. Pencz, B. J. M. **Hiba, A.** Gozse, I. Kisantal, M. Nemeth, M. Nagy, Z. Vanek, B. Zarandy, A. Bokor, J. "Onboard Visual Sense and Avoid System for Small Aircraft". In: *IEEE Aerospace and Electronic Systems Magazine (accepted)* (2016).

A szerző konferencia publikációi

[C1] **Hiba, A.** Nagy, Z. Ruszinko, M. "Memory access optimization for computations on unstructured meshes". In: *Proc. 13th International Workshop on Cellular Nanoscale Networks and their Applications*. 2012.

[C2] Nagy, Z. Nemes, C. **Hiba, A.** Kiss, A. Csík, Á. Szolgay, P. "FPGA based acceleration of computational fluid flow simulation on unstructured mesh geometry". In: *Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on*. IEEE. 2012, pp. 128–135.

[C3] Nagy, Z. Nemes, C. **Hiba, A.** Kiss, A. Csík, Á. Szolgay, P. "Accelerating Unstructured Finite Volume Solution of 2-D Euler Equations on FPGAs". In: *Conference on Modelling Fluid Flow (CMFF'12)*. 2012.

[C4] **Hiba, A.** Nagy, Z. Ruszinkó, M. Szolgay, P. "Data locality-based mesh partitioning methods for dataflow machines". In: *14th International Workshop on Cellular Nanoscale Networks and their Applications*. IEEE, 2014.

[C5] **Hiba, A.** Ruszinko, M. "Real-time combinatorial optimization with applicable partial solution generation". In: *1st International Conference on Engineering and Applied Sciences Optimization*. 2014, pp. 590–599.

[C6] Bauer, P. **Hiba, A.** Vanek, B. Zarandy, A. Bokor, J. "Monocular Image-based Time to Collision and Closest Point of Approach Estimation". In: *24th Mediterranean Conference on Control and Automation*. 2016.

[C7] **Hiba, A.** Zsedrovits, T. Bauer, P. Zarandy, A. "Fast horizon detection for airborne visual systems". In: *2016 International Conference on Unmanned Aircraft Systems*. 2016.

[C8] **Hiba, A.** Orzo, L. "Retina simulator challenges, image processing with a varying resolution sensor". In: *15th International Workshop on Cellular Nanoscale Networks and their Applications*. 2016.

[C9] **Hiba, A.** Zarandy, A. Pencz, B. "Remote Aircraft Detection against Sky Background". In: *15th International Workshop on Cellular Nanoscale Networks and their Applications*. 2016.

[C10] Orzo, L. **Hiba, A.** Zarandy, A. "Deconvolution as a model of blur adaptation in the visual cortex". In: *15th International Workshop on Cellular Nanoscale Networks and their Applications*. 2016.

[C11] Zsedrovits, T. Zarandy, A. Pencz, B. **Hiba, A.** Nameth, M. Vanek, B. "Distant aircraft detection in sense-and-avoid on kilo-processor architectures". In: *Circuit Theory and Design (ECCTD), 2015 European Conference on*. IEEE. 2015, pp. 1–4.

Kapcsolódó publikációk

[R1] Wulf, W. A. McKee, S. A. "Hitting the Memory Wall: Implications of the Obvious". In: *SIGARCH Comput. Archit. News* 23.1 (Mar. 1995), pp. 20–24. ISSN: 0163-5964. DOI: 10.1145/216585.216588. URL: http://doi.acm.org/10.1145/216585.216588.

[R2] Xie, Y. "Future memory and interconnect technologies". In: *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*. 2013, pp. 964–969. DOI: 10.7873/DATE.2013.202.

[R3] Huang, Y.-J. Li, J.-F. "Yield-enhancement Schemes for Multicore Processor and Memory Stacked 3D ICs". In: *ACM Trans. Embed. Comput. Syst.* 13.3s (Mar. 2014), 106:1–106:22. ISSN: 1539-9087. DOI: 10.1145/2567933. URL: http://doi.acm.org/10.1145/2567933.

[R4] Borkar, S. "Thousand Core Chips: A Technology Perspective". In: *Proceedings of the 44th Annual Design Automation Conference*. DAC '07. San Diego, California: ACM, 2007, pp. 746–749. ISBN: 978-1-59593-627-1. DOI: 10.1145/1278480.1278667. URL: http://doi.acm.org/10.1145/1278480.1278667.

[R5] Garey, M. R. Johnson, D. S. *Computers and Intractablility: A Guide to the Theory of NP-completeness*. W. H. Freeman, 1979. ISBN: 0-7167-1044-7.

[R6] Papadimitriou, C. H. "The NP-completeness of the bandwidth minimization problem." In: *Computing* 16 (1976), pp. 263–270.

[R7] Blum, C. Aguilera, M. Roli, A. Sampels, M. *Hybrid Metaheuristics: An Emerging Approach to Optimization*. Studies in Computational Intelligence. Springer, 2008. ISBN: 9783540782940.

[R8] Blum, C. Puchinger, J. Raidl, G. R. Roli, A. "Hybrid metaheuristics in combinatorial optimization: A survey". In: *Applied Soft Computing* 11.6 (2011), pp. 4135–4151.

[R9] Karypis, G. Kumar, V. "Multilevel k-way partitioning scheme for irregular graphs". In: *Journal of Parallel and Distributed Computing* 48.1 (1998), pp. 96–129.

[R10] Cuthill, E. McKee, J. “Reducing the bandwidth of sparse symmetric matrices”. In: *Proceedings of the ACM National Conference, Association for Computing Machinery, New York*. 1969, pp. 157–172.

[R11] Gibbs, N. Poole, W. Stockmeyer, P. “An algorithm for reducing the bandwidth and profile of sparse matrix”. In: *SIAM Journal on Numerical Analysis* 13.2 (1976), pp. 236–250.

[R12] Geuzaine, C. Remacle, J.-F. “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities”. In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331.

[R13] Margoules, F. *Mesh Partitioning Techniques and Domain Decomposition Methods*. Saxe-Coburg Publications, 2007. ISBN: 978-1-874672-29-6.

[R14] Karypis, G. Kumar, V. “A fast and high quality multilevel scheme for partitioning irregular graphs”. In: *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 359–392.

[R15] Pellegrini, F. “Graph partitioning based methods and tools for scientific computing”. In: *Parallel computing* 23.1 (1997), pp. 153–164.

[R16] WEB, *TSPLIB95 SOP problem package, http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/sop/*. 2014.

[R17] WEB, *SOPLIB problem package, http://www.idsia.ch/∼roberto/SOPLIB06.zip*. 2014.

[R18] Anghinolfi, D. Montemanni, R. Paolucci, M. Gambardella, L. “A hybrid particle swarm optimization approach for the sequential ordering problem”. In: *Computers and Operational Research* 38 (2011), pp. 1076–1085.

[R19] Pell, O. Bower, J. Dimond, R. Mencer, O. Flynn, M. J. “Finite-Difference Wave Propagation Modeling on Special-Purpose Dataflow Machines”. In: *Parallel and Distributed Systems, IEEE Transactions on* 24.5 (2013), pp. 906–915. ISSN: 1045-9219. DOI: 10.1109/TPDS.2012.198.