

# PB165 Grafy a sítě: 8. Plánování s komunikací

# Plánování s komunikací

## 1 Plánování s komunikací a s precedencemi

- Popis problému
- Plánování seznamem
- Heuristiky mapování
- Shlukovací heuristiky

## 2 Paralelní úlohy s průběžnou komunikací

- Popis problému
- Rozdělení grafu

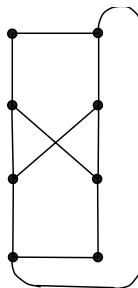
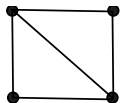




## Popis problému

## Sít' a komunikační zpoždění

- Topologie sítě: příklady



na uzlech dual-core procesor

- $transrate_{j_1, j_2}$ : přenosová rychlost mezi sousedními stroji
- $setupmesg_j$ : (inicializační) doba na posláání zprávy strojem  $j$
- **Komunikační zpoždění**  $c(i_1, i_2, j_1, j_2)$  pro posláání dat z úlohy  $i_1$  úloze  $i_2$  ze stroje  $j_1$  na sousední stroj  $j_2$

$$c(i_1, i_2, j_1, j_2) = \frac{size_{i_1, i_2}}{transrate_{j_1, j_2}} + setupmesg_{j_1}$$

## Popis problému

# Doba provádění. Optimalizace.

- $speed_j$ : rychlost zpracování strojem  $j$
- $setuptask_j$ : (inicializační) doba na nastartování úlohy na stroji  $j$
- **Doba provádění**  $p_{ij}$  úlohy  $i$  na stroji  $j$ :

$$p_{ij} = \frac{p_i}{speed_j} + setuptask_j$$

- **Objektivní funkce (performance measure)**
  - minimalizace makespan (čas dokončení poslední úlohy)
  - do makespan se započítává:
    - doba provádění + komunikační zpoždění

# Plánování seznamem (list scheduling): algoritmus

Jednoduché efektivní algoritmy

- založeny na seřazení úloh v **prioritní frontě**
- složitost algoritmu dána výpočtem priority

## Algoritmus

- ① každému uzlu grafu přiřazena priorita  
prioritní fronta inicializována úlohami bez předchůdců  
úlohy ve frontě seřazeny v klesajícím pořadí dle priority
- ② dokud je fronta neprázdná, prováděj:
  - ① odebrána úloha zepředu z fronty
  - ② je vybrán (nejlepší) volný procesor pro spuštění úlohy
    - vybraný procesor může např. minimalizovat komunikaci s předchůdci úlohy
  - ③ jakmile provedeni všichni přímí předchůdci nějaké úlohy tak je úloha přidána dle priority do fronty

# Priority pro plánování seznamem

- Délka cesty přes uzly  $i_1, i_2 \dots i_n$ :

$$w1 \sum_{i=1}^n p_i + w2 \sum_{i=1}^{n-1} size_{i,i+1}$$

- $w1, w2$ : koeficienty určující poměr mezi dobou trvání a velikostí přenesených dat
- Počáteční uzly: uzly bez předchůdců
- Koncové uzly: uzly bez následníků

## Výpočet priority uzlu (příklady)

- **úroveň (level)**  
délka nejdelší cesty z uzlu do koncového uzlu
- **ko-úroveň (co-level)**  
délka nejdelší cesty z počátečního uzlu do uzlu

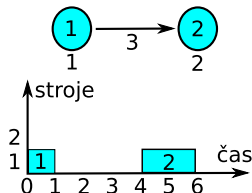


# Plánování seznamem: nastavení

- **Zanedbání doby na komunikaci:**
  - pokud  $i1 \rightarrow i2$  a úlohy  $i1$  a  $i2$  jsou prováděny na stejném stroji, tak dobu na komunikaci zanedbáme
- **Výběr stroje:**
  - pro provádění úlohy  $i$  je vybrán procesor, na kterém bude úloha dokončena nejdříve (v případě více možností je vybrán stroj s nejmenším indexem)
  - pozor, je třeba brát v úvahu, zda na některém stroji nedojde k zanedbání komunikace

# Plánování seznamem: parametry

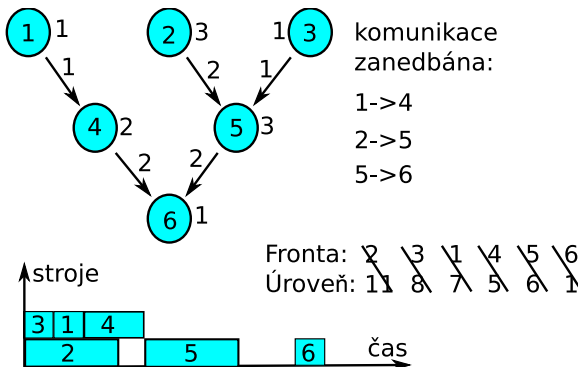
- Nechť  $size_{i,j}$  přímo reprezentuje dobu na přenos dat
- Předpokládejme, že  $w_1 = w_2 = 1$
- **Rychlost stroje**
  - pokud bychom předpokládali, že stroje mají stejnou rychlost, pak lze úlohy přímo rozvrhovat dle zadané  $p_i$  a  $size_{i_1,i_2}$



- **Koncový čas úlohy**
  - spočítán na základě délky cesty (z  $p_i$  a  $size_{i_1,i_2}$ )
  - lze uvažovat možné zanedbání komunikace (úloha je ve frontě až v okamžiku, když jsou všichni předchůdci dopočítáni, tj. známe jejich stroj)

# Plánování seznamem: příklad

Plánování 6 úloh zadaných grafem na 2 stroje se stejnou rychlostí



# Plánování seznamem: cvičení

Vyřešte následující problém plánování s komunikací a s precedencemi pomocí plánování seznamem:

- jsou dány 2 stroje se stejnou rychlostí
- je dáno 6 úloh s dobou trvání  
 $p_1 = 1, p_2 = 1, p_3 = 2, p_4 = 1, p_5 = 2, p_6 = 1$
- jsou dány precedence:  
 $1 \rightarrow 2, 2 \rightarrow 4, 4 \rightarrow 6, 1 \rightarrow 3, 3 \rightarrow 5, 5 \rightarrow 6$
- $size_{i_1, i_2}$  reprezentuje dobu na přenos dat a je zadán takto:  
 $size_{1,2} = 1, size_{2,4} = 3, size_{4,6} = 1,$   
 $size_{1,3} = 2, size_{3,5} = 4, size_{5,6} = 2$

# Plánování seznamem: diskuse

Výběr úlohy s nejvyšší úrovní = **výběr úlohy na kritické cestě**

Problémy při použití heuristiky

- změna kritické cesty
  - ⇐ komunikační zpoždění závisí na alokaci úlohy
  - ⇐ např. při alokaci na stejné stroje se zanedbatelným zpožděním nebo při nestejněměrné vzdálenosti strojů (a počtu hopů)

**Řada heuristik založena na principu (několika) prioritních front**

- rozsáhlé modifikace priorit
- např. produkční plánovací systémy pro plánování úloh na počítačích PBSPro, SGE

# Heuristiky mapování (mapping heuristics)

- Modifikace plánování seznamem
- Více uvažovány reálné parametry:
  - topologie sítě, rychlost procesoru, přenosová rychlost, ...
- Doba provádění úlohy  $i$  na stroji  $j$

$$p_{ij} = \frac{p_i}{speed_j} + setup_{task_j}$$

- Komunikační zpoždění

$$c(i_1, i_2, j_1, j_2) = \left( \frac{size_{i_1, i_2}}{transrate} + setup_{mesg} \right) \times hops_{j_1, j_2} + contdelay_{j_1, j_2}$$

- předpokládán konstantní  $transrate$  a  $setup_{mesg}$
- $hops_{j_1, j_2}$ : počet hopů mezi stroji  $j_1$  a  $j_2$ , (délka nejkratší cesty mezi stroji při jednotkovém ohodnocení hran)
- $contdelay_{j_1, j_2}$ : zpoždění dané zatížením linky mezi  $j_1$  a  $j_2$  (contention delay)

# Heuristiky mapování (mapping heuristics)

- Konstruována a udržována **směrovací tabulka** obsahující
  - odhadované hodnoty  $p_{ij}$  a  $c(i1, i2, j1, j2)$
  - pro každý procesor
    - počet hopů, preferovaná linka pro daný cíl, zpoždění dané zatížením
- **Úlohy plánovány podle nejvyšší úrovně** (podobně jako při plánování seznamem), v případě nejednoznačnosti vybrána úloha s větším množstvím přímých následníků
- Jakmile jsou dokončeni všichni předchůdci úlohy, tak je **úloha naplánována na stroj, kde nejdříve skončí**
- **Koncový čas úlohy na stroji spočítán z**
  - rychlost procesoru, přenosová rychlost, vybraná linka, počet hopů, zpoždění dané zatížením linky

# Rozvrhování se shlukovacími heuristikami (clustering heuristics)

- Rozvrhování řeší
  - ① alokace úloh na stroje
  - ② seřazení úloh na stroji / umístění úloh v čase
- **Shlukovací heuristiky**: řeší alokaci úloh na stroje
- **Shluk**: množina úloh, která bude prováděna na stejném stroji
- **Princip rozvrhování založeného na shlukování**
  - ① Shlukování úloh na nelimitovaný počet plně propojených procesorů
  - ② Namapování shluků a jejich úloh na daný počet procesorů ( $m$ ) prováděním následujících operací
    - **spojování shluků**: pokud je počet shluků vyšší než  $m$
    - **fyzické mapování**: přiřazení shluků na procesory tak, aby byla minimalizována komunikace (na této úrovni uvažována reálná konektivita mezi stroji)
    - **uspořádání úloh**: úlohy jsou na stroji prováděny tak, aby byly splněny závislosti mezi úlohami



# Shlukovací heuristiky

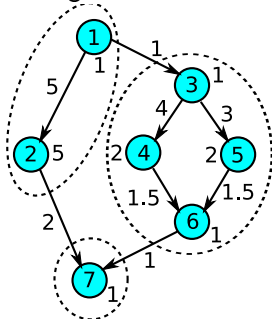
- Shlukovací heuristiky
  - bez backtrackingu (abychom se vyhnuly vysoké složitosti), tj. jakmile jsou jednou shluky spojeny nelze je zpětně rozpojit
- Iniciálně: jedna úloha = jeden shluk
- Typický krok heuristiky
  - spojení shluků + **vynulování hrany**, která je spojuje
  - vynulování hrany: úlohy prováděny na stejném procesoru, kde je čas na komunikaci zanedbatelný
- Kritérium pro výběr hrany na nulování redukce paralelního času rozvrhu

# Naplánovaný graf úloh

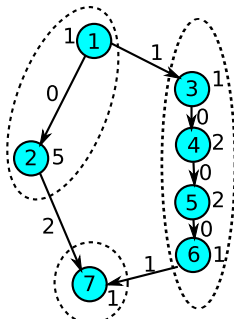
## Naplánovaný graf úloh

- graf úloh **zahrnující** aktuální vynulování hran
- úlohy v jednom shluku jsou seřazeny (jako na procesoru) dle nejvyšší úrovně

původní graf úloh se 3 shluky



naplánovaný graf úloh



# Paralelní čas a dominantní posloupnost

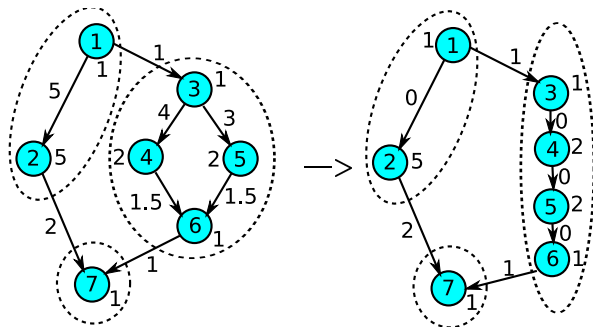
- Připomenutí: délka cesty přes uzly  $i_1, i_2 \dots i_n$ :

$$w1 \sum_{i=1}^n p_i + w2 \sum_{i=1}^{n-1} size_{i,i+1}$$

- **Dominantní posloupnost:**  
nejdelší cesta v naplánovaném grafu úloh
- **Paralelní čas rozvrhu:**  
doba nutná na provedení dominantní posloupnosti
- Pokud  $w1 = w2 = 1$  a  $size_{i,j}$  reprezentuje dobu na přenos  
 $\Rightarrow$  délka cesty odpovídá době na provedení úloh na cestě  
 $\Rightarrow$  paralelní čas rozvrhu = délka cesty dominantní posloupnosti

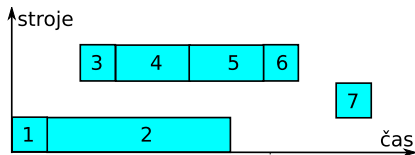
## Shlukovací heuristiky

## Dominantní posloupnost: příklad



Dominantní posloupnost 1,3,4,5,6,7 s délkou 10, tj. paralelní čas rozvrhu je 10

Odpovídající rozvrh na 3 strojích:



# Algoritmus shlukovací heuristiky

- ① Inicializace: všechny hrany označeny jako nevyzkoušené  
každá úloha tvoří jeden shluk
- ② Seřazení všech hran v grafu úloh v klesajícím pořadí dle komunikační ceny
- ③ REPEAT
  - ① vynulování největší nevyzkoušené hrany v seřazeném seznamu, **pokud nevzroste paralelní čas**
  - ② hrana je označena jako vyzkoušená
  - ③ když jsou spojeny dva shluky, úlohy jsou uspořádány dle nejvyšší úrovně

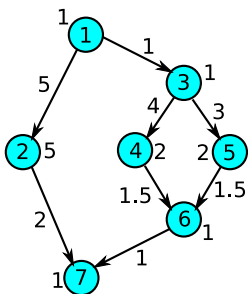
UNTIL všechny hrany jsou vyzkoušené

Komentář: *podobně jako existují různé metody plánování seznamem, tak existují různé varianty shlukovacích heuristik*

## Shlukovací heuristiky

## Shlukovací heuristiky: příklad

Problém:

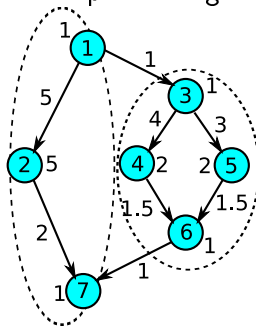


Postup:

seřazení hran, postupné zkoušení na vynulování a spojování shluků:

12 (vynulována, shluk 12), 34 (vynulována, shluk 34), 35 (vynulována, shluk 345), 27 (vynulována, shluk 127), 46 (vynulována, shluk 3456), 56 (vynulována), 13 (nevynulována, paralelní čas by z 10 pro dominantní cestu 1,3,4,5,6,7 vzrostl na 13 s dominantní cestou 1,2,3,4,5,6,7), 67 (nevynulována stejně jako 13)

Řešení pomocí algoritmu:



# Shlukovací heuristiky: cvičení

Vyřešte cvičení z kapitoly plánování seznamem pomocí shlukovací heuristiky.

Je v řešení nějaký rozdíl?

*(náповěda: ano, u shlukovacích heuristik víme, na kterém stroji bude úloha zpracovávána předem, a proto je možné zahájit některé komunikace dříve, a tedy rozvrh skončí dříve)*





## Rozdělení grafu

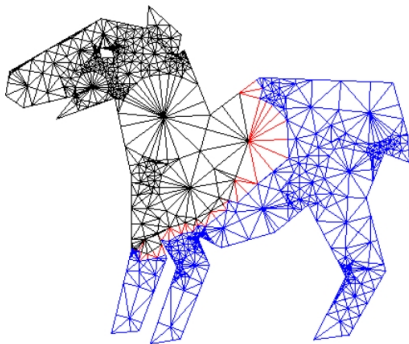
# Rozdělení grafu (graph partitioning)

- Formulace problému vyvažování zátěže jako problému rozdělení grafu
- Rozdělení grafu  $G = (V, E)$  na  $V = V_1 \cup \dots \cup V_m$  tak, že je
  - součet ohodnocení vrcholů „zhruba stejný“
  - součet ohodnocení hran spojující různé  $V_j$  a  $V_k$  minimalizován
- Speciální případ:  $V = V_1 \cup V_2$  **bisekce grafu (graph bisection)**
- Jak nalézt vhodné rozdělení grafu?
  - problém optimálního rozdělení je NP-úplný
    - už pro bisekci: prohledání všech podmnožin množiny vrcholů (podmnožina a její doplněk tvoří  $V_1$  a  $V_2$ )
  - nutné použít dobré heuristiky

## Rozdělení grafu

# Heuristika: opakovaná bisekce grafu

Základní používaný princip při rozdělení grafu:  
rozdělení množiny vrcholů  $V$  na  $2^k$  částí:  
**rekurzivní bisekce grafu  $k$ -krát**

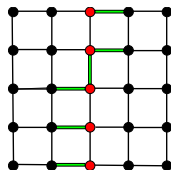


58 dělicích hran

## Rozdělení grafu

# Dělicí hrany (edge separator) vs. dělicí vrcholy (vertex separator)

- **Dělicí hrany:**  $E_S \subseteq E$  dělí  $G$  po odstranění  $E_S$  z  $E$  na dvě stejně velké nesouvisející komponenty  $V$ :  $V_1$  a  $V_2$
- **Dělicí vrcholy:**  $V_S \subseteq V$  dělí  $G$  po odstranění  $V_S$  a všech jejich hran na dvě stejně velké nesouvisející komponenty  $V$ :  $V_1$  a  $V_2$



$E_S =$  zelené hrany

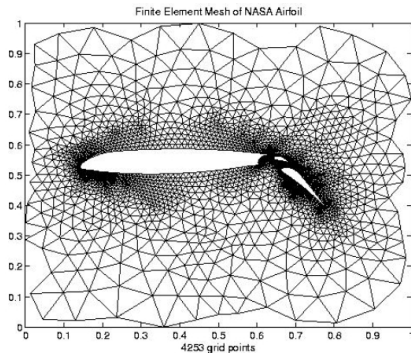
$V_S =$  červené vrcholy

Rozdělení grafu

# Rozdělení se souřadnicemi vrcholů (partitioning with nodal coordinates)

Myšlenka rozdělení pomocí souřadnic vrcholů

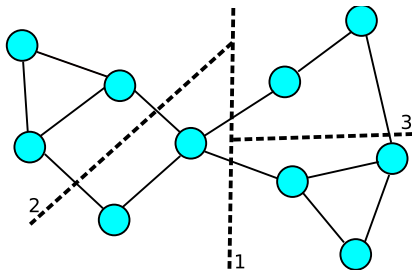
- každý vrchol má souřadnice v prostoru → **rozdělení prostoru**



- pomocí dělicí přímky, která dělí vrcholy v prostoru na poloviny

## Rozdělení grafu

## Opakovaná bisekce grafu s dělicí přímkou: příklad



## Rozdělení grafu

# Grafy a sítě v plánování a rozvrhování: shrnutí

- 1 **Problém rozvrhování:** vlastnosti stroje, omezení, optimalizace
- 2 **Precedenční omezení a disjunktivní grafová reprezentace:** korespondence mezi rozvrhem a grafem
- 3 **Ohodnocené grafy:** obchodní cestující, doba na dopravu, plánování na počítačových sítích
- 4 **Barvení grafu:** algoritmus saturace a problémy přiřazení místností, rezervační problém, plánování operátorů
- 5 **Plánování projektu (precedenční omezení):** kritická cesta, kompromisní heuristika
- 6 **Plánování s komunikací a s precedencemi:** plánování seznamem, heuristiky mapování, shlukovací heuristiky
- 7 **Paralelní úlohy s průběžnou komunikací:** vyvažování zátěže a rozdělení grafu

Rozvrhování na FI: PA167 Rozvrhování, PA163 Omezující podmínky