Institut for Matematik og Datalogi
Syddansk Universitet, Odense

1. juni 2012
LMF

DM205 – Ugeseddel 9

Onsdag d. 6/6

k-server:

1. Opgave 10.1 på side 158.
2. Algoritmen BALANCE er defineret på side 161 i bogen.
   Gennemgå eksemplet på side 163, som viser, at BALANCE ikke er competitive, hvis det metriske rum kan have mere end $k+1$ punkter.
3. Definer og analyser en lazy variant af DC for paging.
   Bemærk: Vi har snakket om, at paging problemet kan modelleres som et k-server problem, hvor alle afstande er 1. Det kan også modelleres som et k-server problem på et træ (se side 158 øverst).

Load balancing, identiske maskiner:

1. Vi så, at Greedy's competitive ratio er $2 - \frac{1}{N}$; d.v.s. dens competitive ratio går mod 2, når N går mod uendelig.
   Den nedre grænse skyldtes et eksempel med fuldstændigt balanceret load, hvorefter et stort job ankommer. Det tyder på, at vi skal forsøge at fordele jobs'ne lidt ujævnt, hvis vi skal opnå en bedre competitive ratio.
   Betragt algoritmen, som forsøger at opnå en bedre competitive ratio C ved at holde en af maskinerne fri til store jobs. Sålænge det ikke giver en ratio større end C, bruger den Greedy på de $N-1$ andre maskiner, og kun når den er tvunget til det, bruger den den sidste maskine.
   Kan denne strategi give en competitive ratio $2-\varepsilon$, for en konstant $\varepsilon > 0$?
   Hvad, hvis vi bruger den samme strategi, men forsøger at holde halvdelen af maskinerne fri til store jobs?
2. Vi så, at for $N = 2$ og $N = 3$ er Greedy optimal (med competitive ratio h.h.v. $3/2$ og $5/3$), men når N bliver stor, går dens competitive ratio mod 2.
   Hvis N kan være vilkårligt stor, ligger den bedste competitive ratio for deterministiske algoritmer mellem ca. $1,88$ og $1,92$.
   Algoritmen med competitive ratio $C = 1 + \sqrt{\frac{1+\ln 2}{2}} \approx 1,92$ hedder MR (opkaldt efter Michaela Wahl og Rudolf Fleischer, som designede og analyserede algoritmen) og fungerer på følgende måde:
   Efter placering af hvert job J_t nummereres maskinerne $M_1^t, M_2^t, \ldots, M_N^t$, så M_i^t er mindst lige så loaded som M_{i+1}^t, $1 \leq i \leq N-1$.
   $L_t(i)$ betegner den gennemsnitlige load på $M_i, M_{i+1}, \ldots, M_N$ (d.v.s. på de $N-i+1$ mindst belastede maskiner) efter de første t jobs.
   D.v.s. $L_t = L_t(1)$ betegner den gennemsnitlige load på samtlige maskiner efter t jobs.
   Lad $m = \lceil \frac{5C-2C^2-1}{C} N \rceil - 1 \approx 0,639N - 1$
   og $k = 2m - N \approx 0,278N$.
   Fordelingen af de første t jobs er *flad*, hvis $\ell_t(k) < \frac{2c-2}{2c-3} D_t(m+1)$, og ellers er den *stejl*.

<u>MR</u>
For hvert job J_t
  Hvis den nuværende fordeling er stejl eller $\ell_{t-1}(m) + w_t > cL_t$
    Placer J_t på M_N
  Ellers
    Placer J_t på M_m

Vis, at med Relative Worst-Order Ratio er Greedy og MR usammenlignelige.

Load balancing, relaterede maskiner:

1. Denne opgave handler om 2 relaterede maskiner. Den ene har hastighed 1, og den anden har hastighed $s \geq 1$. Vi skal analysere Greedy (som er beskrevet nederst på side 204) og algoritmen Fast, som blot placerer alle jobs på den hurtigste maskine.
   (a) Vis, at Fast har competitive ratio
   $$\mathcal{R}(\text{Fast}) = \frac{s+1}{s}$$
   (b) Vis, at Greedy har competitive ratio
   $$\mathcal{R}(\text{Greedy}) = \begin{cases} \frac{2s+1}{s+1}, & \text{hvis } s \leq \frac{1+\sqrt{5}}{2} \\ \frac{s+1}{s}, & \text{ellers} \end{cases}$$
   (c) Sammenlign Greedy og Fast v.h.a. Relative Worst-Order Ratio.
2. Hvis der bliver tid, gennemgår jeg løsningen til opgave 12.2 på side 208.