

Hajautetun laskennan vakaantuminen häiriöiden kestäessä

Jussi Brunberg

Perjantaina 9. marraskuuta 2007

Vakaantuminen häiriöiden kestäessä

- ▶ self-stabilizing ~
 - ▶ itsestabiloi(tu)va
 - ▶ itsevakautuva
 - ▶ toipumiskykyinen
 - ▶ *vakaantuva*/vakautuva

Vakaantuminen häiriöiden kestäessä

- ▶ self-stabilizing ~
 - ▶ itsestabiloi(tu)va
 - ▶ itsevakautuva
 - ▶ toipumiskykyinen
 - ▶ *vakaantuva*/vakautuva
- ▶ määriteltävä laskennan vakaantuminen vain järjestelmän (oikein toimivan) osan suhteen

Vakaantuminen häiriöiden kestäessä

- ▶ Dolev, luku 6, Convergence in the Presence of Faults
- ▶ Esimerkkinä kellojen synkronointi *synkronisessa järjestelmässä*
 - ▶ torakahtelu
 - ▶ kaikkien oikein toimivien synkronoiduttava
 - ▶ bysanttilaiset häiriöt (*byzantine* ~ 'ovela', 'juonikas')
 - ▶ oikein toimivat $> 2/3$ synkronoituvat
 - ▶ bysanttilaiset kenraalit (Lamport, Shostak, Pease 1982)

Rajoittamattomat kellot

- (1) **upon pulse**
- (2) **foreach** $j \in Neighbours(i)$
- (3) **send**($j, clock_i$)
- (4) $max \leftarrow clock_i$
- (5) **foreach** $j \in Neighbours(i)$
- (6) **receive**($clock_j$)
- (7) **if** $clock_j > max$
- (8) $max \leftarrow clock_j$
- (9) $clock_i \leftarrow max + 1$

Rajoittamattomat kellot, torkahtelusietoinen

- (1) **upon pulse**
- (2) **foreach** $j \in \text{Neighbours}(i)$
- (3) **send**(j, clock_i)
- (4) $\text{max} \leftarrow \text{clock}_i$
- (5) **foreach** $j \in \text{Neighbours}(i)$
- (6) **receive**(clock_j)
- (7) **if** $\text{clock}_j > \text{max}$
- (8) $\text{max} \leftarrow \text{clock}_j$
- (9) $\text{clock}_i \leftarrow \text{max} + 1$

Rajoitetut kellot, $M > (n + 1)d$

- (1) **upon pulse**
- (2) **foreach** $j \in \text{Neighbours}(i)$
- (3) **send**(j, clock_i)
- (4) $\text{max} \leftarrow \text{clock}_i$
- (5) **foreach** $j \in \text{Neighbours}(i)$
- (6) **receive**(clock_j)
- (7) **if** $\text{clock}_j > \text{max}$
- (8) $\text{max} \leftarrow \text{clock}_j$
- (9) $\text{clock}_i \leftarrow (\text{max} + 1) \bmod M$

Rajoitetut kellot, $M > 2d$

- (1) **upon pulse**
- (2) **foreach** $j \in \text{Neighbours}(i)$
- (3) **send**(j, clock_i)
- (4) $\text{min} \leftarrow \text{clock}_i$
- (5) **foreach** $j \in \text{Neighbours}(i)$
- (6) **receive**(clock_j)
- (7) **if** $\text{clock}_j < \text{min}$
- (8) $\text{min} \leftarrow \text{clock}_j$
- (9) $\text{clock}_i \leftarrow (\text{min} + 1) \bmod M$

Rajoitetut kellot

- ▶ Liian pieni M voi estää vakaantumisen
- ▶ 13 laskuyksikön rengas, $M = 5$:

$\hookrightarrow 0 \leftrightarrow 1 \leftrightarrow 1 \leftrightarrow 1 \leftrightarrow 2 \leftrightarrow 2 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 3 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 4 \leftrightarrow 4 \leftrightarrow$

$\hookrightarrow 1 \leftrightarrow 1 \leftrightarrow 2 \leftrightarrow 2 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 3 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 4 \leftrightarrow 4 \leftrightarrow 0 \leftrightarrow 1 \leftrightarrow$

Rajoitetut kellot

- ▶ Liian pieni M voi estää vakaantumisen
- ▶ 13 laskuyksikön rengas, $M = 5$:

$\hookrightarrow 0 \leftrightarrow 1 \leftrightarrow 1 \leftrightarrow 1 \leftrightarrow 2 \leftrightarrow 2 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 3 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 4 \leftrightarrow 4 \leftrightarrow$

$\hookrightarrow 1 \leftrightarrow 1 \leftrightarrow 2 \leftrightarrow 2 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 3 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 4 \leftrightarrow 4 \leftrightarrow 0 \leftrightarrow 1 \leftrightarrow$

- ▶ Renkaissa $n = O(M^3)$ riittää yleisesti estämään vakaantumisen

Torkahtelevat kellot

- ▶ Oletetaan täydellinen verkko
 - ▶ kaikki solmut ovat naapureita
- ▶ Toimivan yksikön on synkronoiduttava rajatussa ajassa kaikkien muiden yhtä pitkään toimineiden yksiköiden kanssa
- ▶ *vartoilematon* (wait-free)
 - ▶ synkronoitunut kello ei odota synkronoituvaa

Rajoitettu, vartoilematon: kivi, paperi, sakset

- (1) **upon pulse**
- (2) **in**($\{clock_j\}, \{order_{jk}\}$)
- (3) $\mathcal{NB} \leftarrow \{j \mid \text{millään } k \text{ ei ole } (order_{jk} + 1) \bmod 3 = order_{kj}\}$
- (4) **if** $\mathcal{NB} \neq \emptyset$
- (5) $clock_i \leftarrow \max_{j \in \mathcal{NB}} \{clock_j\} + 1$
- (6) **foreach** $j \neq i$
- (7) **if** $order_{ij} \neq (order_{ji} + 1) \bmod 3$
- (8) $order_{ij} \leftarrow (order_{ij} + 1) \bmod 3$
- (9) **out**($clock_i, \{order_{ij}\}$)

Bysanttilaiset häirikkökellot

- ▶ Korkeintaan $f < n/3$ virheellistä yksikköä
- ▶ 2 perussääntöä
 - ▶ kasvatetaan kun $\geq n - f$ kelloa täsmää
 - ▶ muuten nollataan
- ▶ onneakin tarvitaan

Bysanttilaiset häirikkökellot

- (1) **upon pulse**
- (2) **foreach** $j \in \text{Neighbours}(i)$
- (3) **send**(j, clock_i)
- (4) **foreach** $j \in \text{Neighbours}(i)$
- (5) **receive**(clock_j) (* aikarajoitettu *)
- (6) **if** $|\{j \mid \text{clock}_i = \text{clock}_j\}| < n - f$
- (7) $\text{clock}_i \leftarrow 0$
- (8) $\text{LastIncrement}_i \leftarrow \text{false}$
- (9) **else if** $\text{clock}_i \neq 0$
- (10) $\text{clock}_i \leftarrow (\text{clock}_i + 1) \bmod M$
- (11) $\text{LastIncrement}_i \leftarrow \text{true}$
- (12) **else if** LastIncrement_i
- (13) $\text{clock}_i \leftarrow 1$
- (14) **else**
- (15) $\text{clock}_i \leftarrow \text{random}\{0, 1\}$
- (16) **if** $\text{clock}_i = 1$
- (17) $\text{LastIncrement}_i \leftarrow \text{true}$

Bysanttilaiset häirikkökellot

- ▶ Algoritmi vakaantuu odotusarvoisesti $2^{2(n-f)}M$ sykäyksessä
- ▶ $n = 4$, $f = 1$ vaatii odotusarvoisesti $64M$ sykäystä
- ▶ Isoillakin M järjellisessä ajassa vakaantuva algoritmi saadaan koostamalla kiinalaisen jäännöslauseen avulla
- ▶ Jokainen yksikkö päivittääkin yhden sijasta useaa pienin, keskenään jaottomin M -arvoin rajoitettua kelloa
- ▶ $n = 4$, $f = 1$, $M = 2 \cdot 3 \cdot 5 = 30$
 - ▶ koostamatta $64 \cdot 30 = 1920$ sykäystä
 - ▶ koostetusti $64 \cdot (2 + 3 + 5) = 640$ sykäystä

Kiinalainen jäännöskello

$$(0, 0, 0) \sim 0 \rightarrow$$

$$(1, 1, 1) \sim 1 \rightarrow$$

$$(0, 2, 2) \sim 2 \rightarrow$$

$$(1, 0, 3) \sim 3 \rightarrow$$

$$(0, 1, 4) \sim 4 \rightarrow$$

$$(1, 2, 0) \sim 5 \rightarrow$$

$$(0, 0, 1) \sim 6 \rightarrow$$

$$(1, 1, 2) \sim 7 \rightarrow$$

$$(0, 2, 3) \sim 8 \rightarrow$$

$$(1, 0, 4) \sim 9 \rightarrow$$

$$(0, 1, 0) \sim 10 \rightarrow$$

$$(1, 2, 1) \sim 11 \rightarrow$$

$$(0, 0, 2) \sim 12 \rightarrow$$

$$(1, 1, 3) \sim 13 \rightarrow$$

$$(0, 2, 4) \sim 14 \rightarrow$$

$$(1, 0, 0) \sim 15 \rightarrow$$

$$(0, 1, 1) \sim 16 \rightarrow$$

$$(1, 2, 2) \sim 17 \rightarrow$$

$$(0, 0, 3) \sim 18 \rightarrow$$

$$(1, 1, 4) \sim 19 \rightarrow$$

$$(0, 2, 0) \sim 20 \rightarrow$$

$$(1, 0, 1) \sim 21 \rightarrow$$

$$(0, 1, 2) \sim 22 \rightarrow$$

$$(1, 2, 3) \sim 23 \rightarrow$$

$$(0, 0, 4) \sim 24 \rightarrow$$

$$(1, 1, 0) \sim 25 \rightarrow$$

$$(0, 2, 1) \sim 26 \rightarrow$$

$$(1, 0, 2) \sim 27 \rightarrow$$

$$(0, 1, 3) \sim 28 \rightarrow$$

$$(1, 2, 4) \sim 29 \rightarrow$$

Loppuhuomioita

- ▶ (vahvat) oletukset kriittisiä
- ▶ synkronisuus
 - ▶ asynkronisuus vaikeuttaa oleellisesti
- ▶ täydellinen verkko