

Skedulerisimulaattorin implementointi fysiikkatöille ja sen matemaattinen validointi

Rasmus Hotakainen

24.01.2011

Ohjaaja: Tapio Niemi

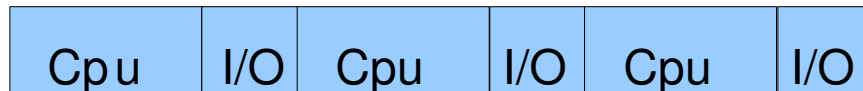
Valvoja: Harri Ehtamo

Tausta ja työn tavoite

- Työ tehtiin *Helsinki Institute of Physics:ille*, Green computing tutkimusryhmälle
- Tarkoitus oli mallintaa fysiikkatöiden laskemista tietokoneessa *simulaattorilla*
- Simulaattori *validoitiin* lopuksi vertaamalla tuloksia oikeisiin testiajoihin
- Simulaattorin takana on energiansäästötavoitteet tietokonelaskennassa jossa pyritään käyttämään *tietokoneresursseja mahdollisimman tehokkaasti*
- Tavoitteena simulaattori joka antaa *käyttökelpoisia tuloksia nopeasti*

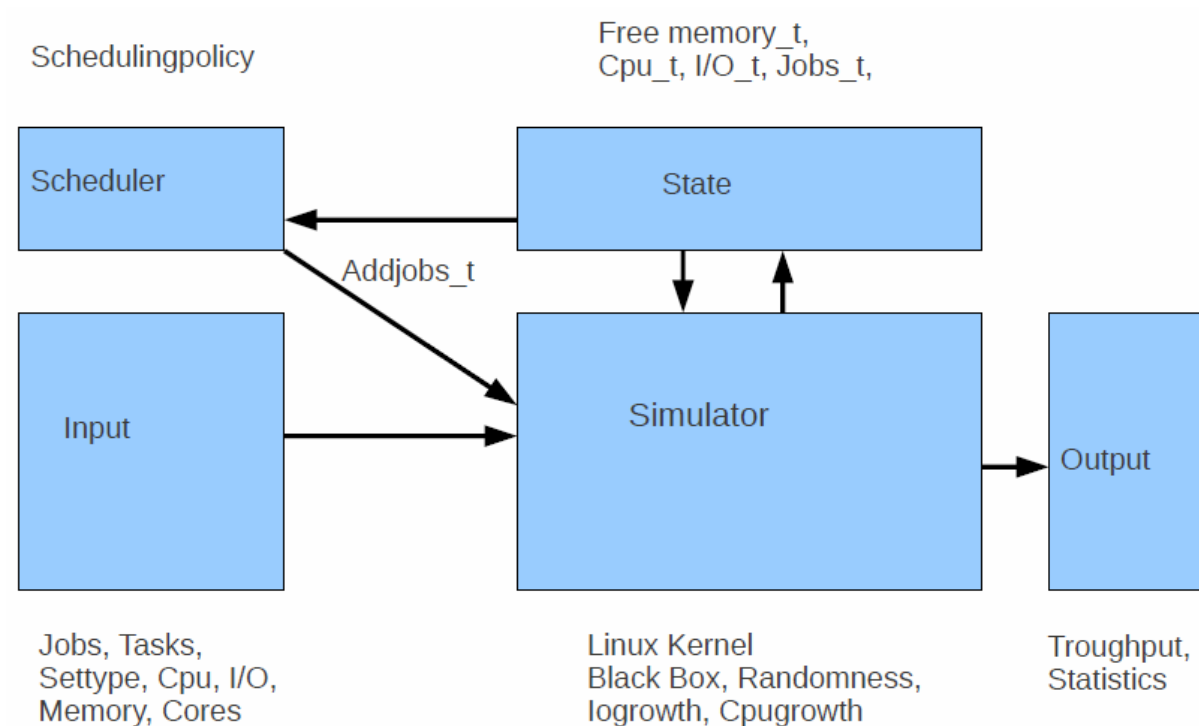
Simulaattorin matemaattinen malli 1/2

- Työt koostuvat pienemmistä tehtävistä jotka vuorotellen käyttävät I/O:ta ja Cpu:ta. Tehtävien pituudet vaihtelevat ja riippuvat töistä



- Työt varaavat keskusmuistia, ei kuitenkaan otettu huomioon simulaattorissa
- Töiden ja tehtävien resursikäyttö on *Linux Kernel Skedulerin* mukaista – *short time scheduling*
- Skedulointi on *Sun Grid Enginen Skedulerin* mukaista – *long time scheduling*
- Malli on nk *Black Box* malli jota on vaikea tutkia analyttisesti mutta jota voidaan kontrolloida.

Simulaattorin matemaattinen malli 2/2



- Simulaattorille syötetään Joukon ja Tietokoneen ominaisuuudet. Skeduleri skeduloi töitä tilan perusteella. Tulosteena on läpimenoaika

Simulaattorin toteutus

- Simulaattori toteutettiin Java kielellä Olio ohjelmoinnilla
- Eri luokat simulaattorin eri osille
- Simulaattorissa eri parametreja joita voidaan säätää simulaattorin toiminnan parantamiseksi. Tärkeimmät niistä *logrowth* ja *Cpugrowth* joita voidaan muuttaa uusien tulosten esiintyessä
- Suurimmat ongelmat numeerisessa laskennassa jotka kuitenkin vältettiin käyttämällä sopivia tietotyyppejä
- Simulaattori debuggattiin / testattiin kunnolla

Simulaattorin validointi – Testiasetelma 1/2

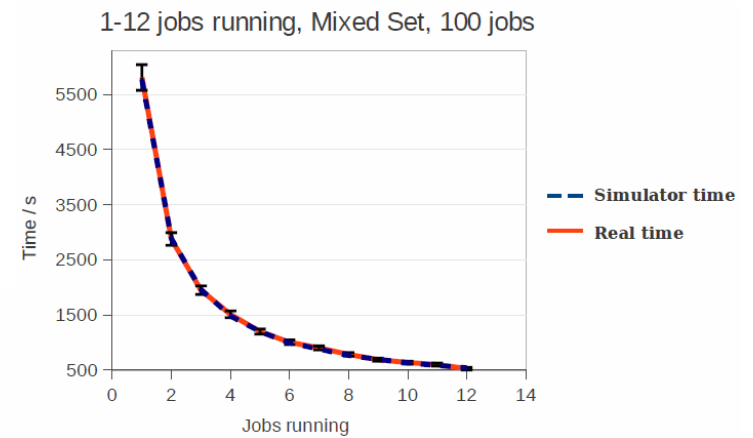
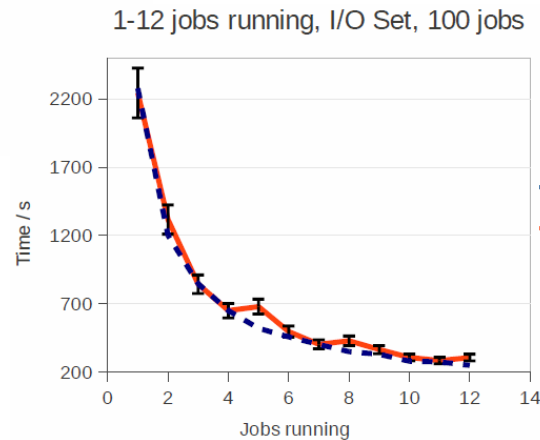
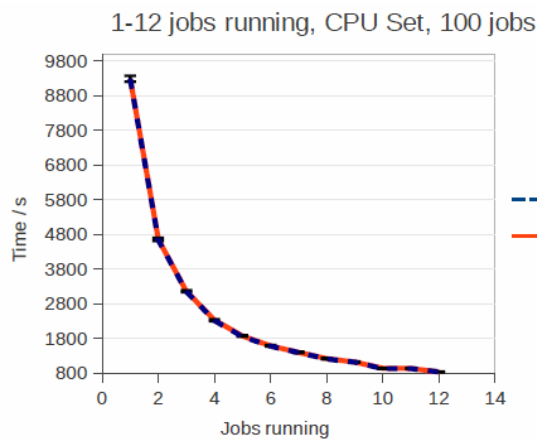
- Validoinnissa käytettiin testiasetelmaa jossa yksinkertaisia töitä, joukkoja sekä yksinkertainen tietokone
- 2 eri työtä: *Cpu_test* ja *Io2*
- 3 eri joukkoa: *I/O Set*, *CPU Set* ja *Mixed Set*
- Jokaisessa joukossa 100 työtä
- Testitietokone joka oli oikea Tietokone jota HIP käyttää. Testikoneessa oli *12 ydintä*
- Validointi tehtiin kahdella eri tapaa

Simulaattorin validointi – Testiasetelma 2/2

- *Vakiomäärä töitä ajossa* asetelmassa pidettiin koko ajan vakiomäärä töitä ajossa. Töiden määrä oli 1-24
- *Dynaaminen skedulointi* asetelmassa skeduloitiin töitä tasaisin väliajoin jolloin töiden määrä ei ole vakio. Aja vaihtelivat setista riippuen
- Validoinnissa verrattiin simulaattorin antamia tuloksia oikeisiin ajoihin
- Simulaattorin parametreja säädettiin ajojen välillä tulosten parantamiseksi

Validointi – Tulokset 1/3

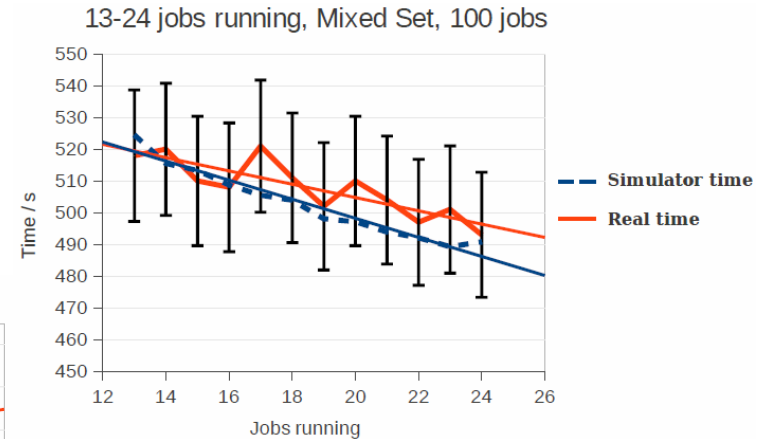
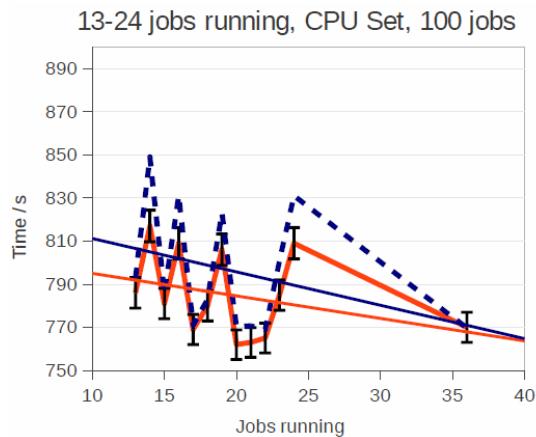
- Alla tulokset ajoista kun vakiomäärä töitä ajossa, 1-12 töitä ajossa kaikille joukoille



Tulokset hyviä

Validointi – Tulokset 2/3

- Alla tulokset ajoista kun vakiomäärä töitä ajossa, 13-24 töitä ajossa kaikille joukoille

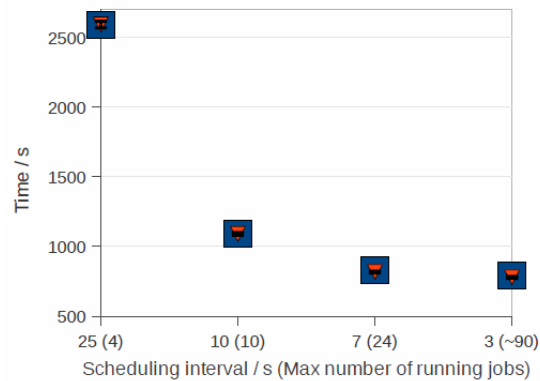


Trendi mallinettu oikein, muistin loppuminen voi olla ongelma

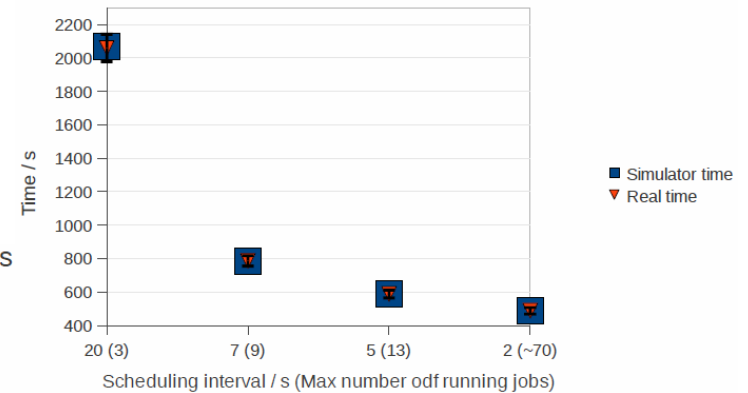
Validointi – Tulokset 3/3

- Alla tulokset ajoista kun dynaaminen määrä töitä ajossa. Skedulointivälit riippuvat joukosta

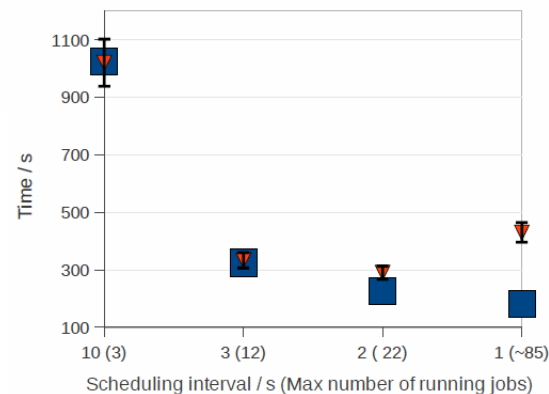
Different scheduling intervals, CPU Set, 100 jobs



Different scheduling intervals, Mixed Set, 100 jobs



Different scheduling intervals, I/O Set, 100 jobs



Hyvät tulokset, muistin loppuminen taas ongelmallista

Yhteenveto

- Kokonaisuudessaan tulokset hyviä
- Heiluntaa tilanteissa jossa paljon I/O:ta
- Muisti voi loppua tietokoneessa mikä johtaa eri tuloksiin
- Seuraavaksi muistinkäytön tutkimista ja implementointia
- Simulaattori hyvä pohja tulevalle tutkimukselle energiatehokkaan skeduloinnin alalla
- Simulaattori antaa käyttökelpoisia tuloksia nopeasti, trendi mallinnettu hyvin

Kiitos!
Kysymyksiä?