Da: **Laura Carnevali** laura.carnevali@unifi.it
Oggetto: Re: Project work Sarri
Data: 30 maggio 2025 alle ore 17:03
A: Edoardo Sarri edoardo.sarri@edu.unifi.it
Cc: Imad Zaza imad.zaza@unifi.it

Buonasera Edoardo,

rispondo circa i punti rimasti aperti:

- Confermo che va bene interrompere una traccia una volta che si verifica deadline miss. La ragione è che ci interessa spiegare, in modo automatizzato, la ragione di questo fallimento. I fallimenti successivi sarebbero invece causati a catena dal primo.
- Confermo che è utile registrare nel log eventuali timeframe violations (violazioni del tempo min-max di esecuzione di un chunk).
- Potremmo aggiungere fault nella gestione del PCP, che avvengono con una certa probabilità (come nel caso di timeframe violation), in particolare:
    - innalzare la priorità dinamica di un task ad un valore errato (minore o maggiore di quello corretto) --> questo fault avviene a livello di PCP
    - non acquisire (e quindi nemmeno rilaciare) un semaforo prima dell'uso di una risorsa condivisa --> questo fault avviene a livello di chunk

Per qualsiasi dubbio, ci possiamo vedere o sentire.

Cordiali saluti,

Laura Carnevali

--

Laura Carnevali
Associate Professor
055 2758519

UNIVERSITÀ DEGLI STUDI DI FIRENZE
Dipartimento di Ingegneria dell'Informazione (DINFO)

--

On 27/05/25 12:03, Laura Carnevali wrote:

> Buongiorno,
>
> riporto di seguito le note dell'incontro di oggi:
>
> - Verificare se in ogni traccia vogliamo avere un solo failure oppure anche più di uno (Carnevali)
> - Verificare se ci sono altri fault che vogliamo iniettare nel task-set (Carnevali)
>
> - Rappresentare il fatto che un chunk può essere soggetto a un fault nella selezione del tempo di esecuzione (viene campionato un additional execution time)
> - Implementazione della politica di scheduling EDF
> - Circa come testare:
>   - RM, tempi di esecuzione deterministici
>   - senza risorse condivise: valgono i test di scheudlabilità di LL e HB: si possono testare task-sey schedulabili oppure non schedulabili con fattore di utilizzo >1 (per quelli non schedulabili con fattore di utilizzo <1 servirebbe trovare un'implementazione della response time analysis)
>   - con risorse condivise: stessa cosa con i test di schedulabiltà estesi
>   - EDF, tempi di esecuzione deterministici
>   - senza risorse condivise: si possono avere facilmente task-set schedulability (fattore di utilizzo <=1) e non schedulabili (fattore di utilizzo >1)
>   - con risorse condivise: test di schedulabilità estesi (in maniera analoga a RM) e processor demand approach
>
> Cordiali saluti,
> Laura Carnevali
>
> On 27/05/25 10:20, Laura Carnevali wrote:
>
>> Buongiorno Edoardo,,
>>
>> invece che in laboratorio, ci vediamo al mio ufficio (sempre al secondo piano ma nell'ala sinistra, dal laboratorio devi percorrere tutta la U del corridoio per intenderci). Mandami un ack quando vedi questo messaggio.
>>
>> Cordiali saluti,
>>
>> Laura Carnevali

On 25/05/25 15:02, Laura Carnevali wrote:

Buongiorno Edoardo,

potremmo fare martedì 27 alle 11:00 presso il lab di tecnologie del software. Se hai della documentazione pronta, ce la potresti intanto inviare?

Imad, per te andrebbe bene?

Cordiali saluti,

Laura Carnevali

On 23/05/25 12:08, Edoardo Sarri wrote:

Buongiorno.

Se per voi va bene fissare la prossima settimana io avrei una prima implementazione con la relativa documentazione da farvi vedere per capire a che punto siamo ed eventualmente come andare avanti. Come l'altra volta per evitare rimbalzi di mail vi allego il mio programma della prossima settimana: se riusciamo a fare o prima o dopo o tra due lezioni sarebbe perfetto.

Grazie,
Edoardo Sarri.

Il giorno 20 mag 2025, alle ore 16:21, Laura Carnevali <laura.carnevali@unifi.it> ha scritto:

Buonasera Edoardo,

bene, volentieri!

Cordiali saluti,

Laura Carnevali

On 20/05/25 16:18, Edoardo Sarri wrote:

Ok guardo il paper e vedo se riesco a tirare fuori qualche test. Lo scopo per cui volevo fare questi test era cercare di capire se ci sono errori concettuali nel codice oppure torna tutto.
Magari appena ho perfezionato le ultime cose fissiamo un incontro e vediamo insieme a che punto siamo.

Grazie,
Edoardo Sarri.

Il giorno 20 mag 2025, alle ore 16:12, Laura Carnevali <laura.carnevali@unifi.it> ha scritto:

Buonasera Edoardo,

bene!

La schedulabilità di un task-set con le caratteristiche che abbiamo delineato (in particolare, rilasci periodici o sporadici, tempi di esecuzione compresi fra un minimo e un massimo, risorse condivide da usate in mutua esclusione e relativi protocolli di accesso a priorità di dinamica) non può essere verificata con test o tecniche di soluzione di tipo analitico (per questo non si trova nulla al riguardo nel libro di Buttazzo).

La schedulabilità potrebbe essere verificata attraverso l'analisi dello spazio degli stati di un modello PTPN nel quale le priorità delle transizioni non sono fissate ma sono funzioni della marcatura (in modo tale da rappresentare protocolli a priorità dinamica). Questo però va oltre gli scop di questo project work.

Ai fini di questo project work, per individuare un task-set schedulabile possiamo considerare una variante del task-set riportato in Fig.5 in questo articolo:
https://stlab.dinfo.unifi.it/carnevali/papers/11_CRV_TSE.pdf In particolare, questo task-set è schedulato usando il priority ceiling emulation protocol (un protocollo a priorità statica, che può essere facilmente rappresentato da una PTPN dove le priorità delle transizioni sono fissate). Se è schedulabile usando priority ceiling emulation protocolo, immagino che lo sia anche usando PCP. Considerare PCP invece di priority ceiling emulation protocol è l'unica modifica da apportare al task-set.

Un altro modo di procedere è quello di considerare nei primi esperimenti tempi di esecuzione deterministici e task periodici, così che la schedulabilità possa essere verificata con i test visti a

deterministici e task periodici, così che la schedulabilità possa essere verificata con i test visti a lezione (la versione "extended" se ci sono risorse condivise da usare in mutua esclusione).

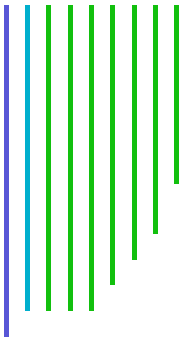
Se qualcosa non è chiaro, possiamo fissare un ricevimento.

Cordiali saluti,

Laura Carnevali

On 19/05/25 12:17, Edoardo Sarri wrote:

> Buongiorno.,
> anche se è passato un po' di tempo in questo mese e mezzo abbondante ho lavorato al progetto.
> Sono arrivato all'implementazione di RM e PCP con test unitari e class diagram.
>
> Prima di scrivere un po' di documentazione e di fissare un incontro per parlarne insieme volevo tastarlo su degli esempi di taskset che so essere feasible o non feasible, sia con risorse che senza; ho guardato il libro "Hard Real-Time Computing System che era indicato su moodle", ma non ho trovato esempi. Ho provato a farmi generare qualcosa anche da ChatGPT, ma il risultato non è stato dei migliori.
> Sapete indicarmi qualche risorsa dove guardare?
>
> Grazie in anticipo,
> Edoardo Sarri.
>
> > Il giorno 26 mar 2025, alle ore 15:00, Laura Carnevali <laura.carnevali@unifi.it> ha scritto:
> >
> > Generazione di tracce di esecuzione di task-set real-time in ambiente simulato (in Java)
> >
> > - Task
> > - pattern di rilascio (distribuzione del tempo fra due arrivi)
> > - numero di chunk (computazioni) che costituiscono un job del task
> > - ogni chunk è caratterizzato da una distribuzione del tempo di esecuzione, eventuale richiesta di risorse da usare in mutua esclusione (prima dell'esecuzione del chunk acquisisco il semaforo, e dopo l'esecuzione lo rilascio)
> > - deadline
> > - rappresentiamo distribuzioni di probabilità (per tempo fra due rilasci e tempo di esecuzione) direttamente con il loro sampler: https://github.com/oris-tool/sirio/blob/master/sirio/src/main/java/org/oristool/simulator/samplers/Sampler.java
> > - aggiungere il deterministic sampler che campiona sempre lo stesso valore
> > - TaskSet: è un insieme di Task
> > - Risorsa da usare in mutua esclusione + semaforo che la gestisce
> > - CPU: assumiamo che sia una sola
> > - Scheduler
> > - Rate monotonic
> > - Earliest deadline first
> > - Protocollo di accesso alle risorse
> > - Priority Ceiling Protocol
> > - (Priority Inheritance Protocol)
> > - Possibilità di iniettare fault
> > - Task aggiuntivo che fa cycle stealing (si aggiungono uno o più task a priorità alta allo scenario)
> > - Task programming defect (gestione errata della priorità o dell'acquisizione delle risorse)
> > - Fallimenti osservati:
> > - deadline miss
> > - violazione del tempo di computazione di un chunk (troppo basso o troppo alto)
> > - Generazione di tracce dell'esecuzione: ogni traccia è una sequenza di coppie <evento, tempo> dove evento può essere
> > - rilascio di un job di un task
> > - acquisizione/rilascio di un semaforo da parte di un job di un task
> > - completamento di un chunk
> > - completamento di un job di un task (in realtà coincide con il completamento dell'ultimo chunk del task)
> >
> > - per la descrizione del fault & failure model:
> >
> > https://stlab.dinfo.unifi.it/carnevali/papers/11_CRV_TSE.pdf
> >
> > - per una descrizione di SIrio/ORIS
> >
> > https://www.oris-tool.org/papers/2019_tse_oris_tool.pdf
>
> --
>
> **Laura Carnevali**



Associate Professor
055 2758519

UNIVERSITÀ DEGLI STUDI DI FIRENZE
Dipartimento di Ingegneria dell'Informazione (DINFO)

--