




Analisi di schedulabilità



Anno accademico 2008/9
Sistemi *Real-Time*

Tullio Vardanega, tullio.vardanega@math.unipd.it

Corso di Laurea Magistrale in Informatica, Università di Padova
1/18




Analisi di schedulabilità

Obiettivi

- ❑ Verificare la soddisfazione dei requisiti temporali del sistema
- ❑ Usando una descrizione astratta del sistema
 - *Workload model*
- ❑ Analisi valida per il modello utilizzato
 - Sotto le assunzioni del modello
 - Attenzione alla relazione tra il modello architetturale e il sistema a tempo di esecuzione

Corso di Laurea Magistrale in Informatica, Università di Padova
2/18



Analisi di schedulabilità


Workload model

- ❑ Rappresentazione astratta del sistema a fini di analisi
 - Con vista esclusivamente sugli aspetti temporali

$$\tau_i(\phi_i, T_i, C_i, D_i)$$

- ϕ_i : fase (tempo di rilascio del primo *job*)
- T_i : periodo o separazione minima
- C_i : *worst-case execution time*
- D_i : *deadline* relativa
- N : numero di *task* del sistema

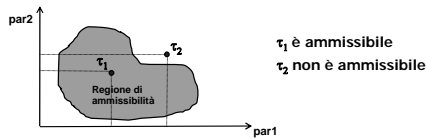
Corso di Laurea Magistrale in Informatica, Università di Padova
3/18




Analisi di schedulabilità

Regione di ammissibilità

- ❑ Concetto geometrico per rappresentare l'insieme dei sistemi schedulabili
 - Spazio N-dimensionale dove N sono i parametri influenti
 - In relazione a uno specifico algoritmo di schedulazione
 - In funzione dei parametri temporali



Corso di Laurea Magistrale in Informatica, Università di Padova
4/18



Analisi di schedulabilità


Test di utilizzazione – 1

- ❑ Limite di utilizzazione [Liu & Layland, 1973]
 - Assunzioni molto restrittive
 - *Task* indipendenti (nessun utilizzo di risorse condivise, nessuna precedenza)
 - $\phi_i = 0, D_i = T_i$
 - Singolo punto di sospensione
 - *Overhead* di sistema considerati ininfluenti
 - Per sistemi FPS e priorità assegnate con algoritmo *Rate Monotonic*

$$U = \sum_{i=1}^N \frac{C_i}{T_i} \leq N(2^{1/N} - 1) \equiv U_{RM}$$
 - Per sistemi EDF

$$U = \sum_{i=1}^N \frac{C_i}{T_i} \leq 1 \equiv U_{EDF}$$

Corso di Laurea Magistrale in Informatica, Università di Padova
5/18

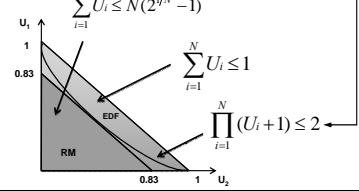


Analisi di schedulabilità

Test di utilizzazione – 2

- ❑ Sono state formulate successive estensioni
 - P.es.: "*Hyperbolic Bound*" [Bini et al., 2000]
 - Per sistemi FPS di *task* periodici, con priorità assegnate secondo RM

$$\sum_{i=1}^N U_i \leq N(2^{1/N} - 1)$$



Corso di Laurea Magistrale in Informatica, Università di Padova
6/18

Analisi di schedulabilità

Test di utilizzazione – 3

□ **Valutazione**

- **Efficiente: complessità O(N)**
 - Utile come *test* di ammissione *on-line* di lavori aperiodici in sistemi aperti
- **Test su RM sufficienti ma non necessari**
 - *Test* su EDF necessario e sufficiente
- **Limitata utilità**
 - Assunzioni restrittive
 - Risponde in modo binario alla domanda "Il sistema è schedulabile?"
 - Difficile trarne indicazioni conclusive

Corso di Laurea Magistrale in Informatica, Università di Padova

7/18

Analisi di schedulabilità

Response Time Analysis – 1

□ **Dalla formulazione di [Pandya et al., 1986]**

- Successive estensioni per incorporare *blocking time* e *release jitter*

$$w_i^{n+1} = C_i + B_i + \sum_{j \in hp(i)} \left\lceil \frac{w_j^n + J_j}{T_j} \right\rceil C_j$$

Assunzioni

Sistema monoproiettore

$D_i \leq T_i$

FPS

Nessuna precedenza fra task

Condizione di terminazione

$w_i^{n+1} = w_i^n$

Condizione di schedulabilità

$R_i \leq D_i$

Corso di Laurea Magistrale in Informatica, Università di Padova

8/18

Analisi di schedulabilità

Response Time Analysis – 2

□ **Estensione per *deadline* arbitrarie [Tindell, 1992]**

- Nello stesso momento possono essere attivi più *job* del *task* in esame

$$w_i^{n+1}(q) = qC_i + B_i + \sum_{j \in hp(i)} \left\lceil \frac{w_j^n(q) + J_j}{T_j} \right\rceil C_j$$

Response Time

$R_i(q) = w_i(q) - (q-1)T_i + J_i$

$R_i = \max(R_i(q))$

Iterazione per accumulo *job*

$\forall q \in \{1, 2, 3, \dots\} w_i(q) > qT_i$

↓

Terminazione

$w_i(q) \leq qT_i$

Corso di Laurea Magistrale in Informatica, Università di Padova

9/18

Analisi di schedulabilità

Response Time Analysis – 3

□ **Valutazione**

- **Complessità pseudo-polinomiale**
 - Adeguata per analisi statica del sistema
- **Formulazione esatta ("*exact response time*") per sistemi monoproiettore**
- **Evidenza quantitativa per ogni *task* del rispetto o della violazione di *deadline***
 - Nessuna indicazione sulle conseguenze causate da una modifica ai parametri dei *task* (periodo/separazione minima e WCET)
 - "Come rendere schedulabile un sistema non schedulabile?"
 - "Come migliorare un sistema schedulabile?"
 - Valutazione non banale anche in sistemi con modesto numero di *task*

Corso di Laurea Magistrale in Informatica, Università di Padova

10/18

Analisi di schedulabilità

Analisi per sistemi Ravenscar – 1

□ **In L06 abbiamo fissato un insieme di restrizioni per un modello di concorrenza predicibile**

- Per sistemi analizzabili staticamente
- Per poter utilizzare tecniche di analisi più sofisticate

□ **Ora vediamo una panoramica dei possibili raffinamenti dell'analisi**

Corso di Laurea Magistrale in Informatica, Università di Padova

11/18

Analisi di schedulabilità

Analisi per sistemi Ravenscar – 2

$$R_i^{n+1} = B_i + CS1 + C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_j^n + J_j^r}{T_j} \right\rceil (CS1 + C_j + TS + CS2) + I_{clock}^{R_i} + I_{extnt}^{R_i}$$

Tempo di blocco (protocollo di sincronizzazione e kernel)

↑

Context switch in ingresso

↑

Jitter di attivazione

↑

Tempo di avvio sospensione

↑

Context switch in uscita

↑

Interferenza orologio

↑

Interferenza interruzioni

↑

$R_1 = B_i + CS1 + C_i$

$R_i = R_i^n + J^w$ ← Jitter di risveglio

Corso di Laurea Magistrale in Informatica, Università di Padova

12/18

Analisi di schedulabilità

Transazione – 1

Stabilisce una relazione causale fra attività

- Aggiunge informazioni rilevanti per l'analisi non presenti nel *workload model* classico
 - Dipendenze di attivazione fra attività

- Nozione originariamente istituita per l'analisi di sistemi distribuiti
 - Utile anche per l'analisi di *pattern* architettrurali risolti nello stesso nodo computazionale

Corso di Laurea Magistrale in Informatica, Università di Padova
13/18

Analisi di schedulabilità

Transazione – 2

Due tipi di relazioni di dipendenza

- Relazione diretta di precedenza (produttore – consumatore)
 - τ_2 non può procedere fino al termine di τ_1
- Dipendenza indiretta da priorità
 - Il task τ_4 non subisce interferenza da parte di τ_3 (assumendo FPS)

Utili per analisi più sofisticate

- Riduzione del pessimismo

Corso di Laurea Magistrale in Informatica, Università di Padova
14/18

Analisi di schedulabilità

Esempio – 1

Callback pattern per sistemi Ravenscar

Corso di Laurea Magistrale in Informatica, Università di Padova
15/18

Analisi di schedulabilità

Esempio – 2

Vogliamo informazioni sul completamento dell'intera sequenza di attività

Corso di Laurea Magistrale in Informatica, Università di Padova
16/18

Analisi di schedulabilità

Sensitivity Analysis – 1

Studia i cambiamenti ammissibili in un sistema dato

- Migliorare un sistema già schedulabile
- Rendere schedulabile un sistema non ammissibile

τ Posizione del sistema nella regione di ammissibilità

ΔC_1^{\max} Massima variazione per il WCET di τ_1 (negativa nell'esempio)

ΔC_2^{\max} Massima variazione per il WCET di τ_2 (negativa nell'esempio)

τ Posizione del sistema nella regione di ammissibilità

ΔC_1^{\max} Massima variazione per il WCET di τ_1 (negativa nell'esempio)

ΔC_2^{\max} Massima variazione per il WCET di τ_2 (negativa nell'esempio)

Corso di Laurea Magistrale in Informatica, Università di Padova
17/18

Analisi di schedulabilità

Sensitivity Analysis – 2

Computazione molto onerosa

Teoria ancora in sviluppo

- Non contempla ancora risorse condivise, sistemi distribuiti, sistemi partizionati

Potenzialmente molto utile

- Esplora lo spazio della soluzione durante la progettazione
 - Applicata per il momento a periodo/separazione minima e WCET
- Studia le conseguenze di modifiche del sistema
 - Come e quanto è possibile migliorare un sistema schedulabile permettendo l'inserimento di funzioni più complesse
 - Come modificare un sistema non schedulabile operando sui periodi e sul WCET dei *task* rinegoziandone i requisiti funzionali o scegliendo altre soluzioni

Corso di Laurea Magistrale in Informatica, Università di Padova
18/18