RMA I VREMENSKI POBUĐENI SISTEMI

- RMA analiza, III deo

Proširenja osnovne teorije

- 1) Iskorišćenje veće od U(n).
- 2) Kašnjenje zbog smene zadataka.
- 3) Krajnji rok nije jednak periodi.
- 4) Serveri prekida: zadržavaju visok prioritet nezavisno od dužine periode.
- 5) Zadatak u toku izvršenja menja prioritet (u suštini sastoji se od niza podzadataka).

Iskorišćenje veće od U(n): rešenje je analiza vremena odziva zadatka ti

$$a_0 = \sum_{j \in H + \{i\}} C_j \qquad a_{n+1} = C_i + \sum_{j \in H} \left\lceil \frac{a_n}{T_j} \right\rceil C_j$$

- H je skup zadataka čiji prioritet je veći od prioriteta t_i
- $a_{n+1} = a_n$ je kriterijum za završetak rekurzivnog razvoja
- Vreme odziva zadatka je jednako a_{n+1}

2) Kašnjenje zbog smene zadataka: tzv. „Prošireni test vremena odziva“

$$a_0 = B_i + \sum_{j \in H + \{i\}} \left(C_j + 2S \right)$$

$$a_{n+1} = B_i + C_i + 2S + \sum_{j \in H} \left\lceil \frac{a_n}{T_j} \right\rceil \left(C_j + 2S \right)$$

- 2S – vreme potrebno za dve smene zadataka

3) Krajnji rok nije jednak periodi

$$U(n,\Delta_i)=\begin{cases} n\left(\left(2\Delta_i\right)^{1/n}-1\right)+1-\Delta_i, & 0.5<\Delta_i\le 1.0 \\ \Delta_i & \Delta_i\le 0.5 \\ \Delta_i(n-1)\left(\left(\dfrac{\Delta_i+1}{\Delta_i}\right)^{1/(n-1)}-1\right), & \Delta_i=2,3,\ldots \end{cases}$$

- Δ_i je definisano kao odnos D_i i T_i (D_i/T_i)

4) Serveri prekida: zadržavaju visok prioritet nezavisno od dužine periode

$$f_i = \sum_{j \in Hn} \frac{C_j + 2S}{T_j} + \frac{C_i + 2S}{T_i} + \frac{B_i}{T_i} + \frac{1}{T_i} \sum_{k \in Hl} (C_k + 2S)$$

- Hn skup prioritetnijih zadataka, koji imaju periode kraće od zadatka t_i, i iz tog razloga istiskuju t_i više puta;
- Hl je skup prioritetnijih zadataka, koji imaju periode duže od periode zadatka t_i, tako da ga istiskuju samo jednom.
- Test granice iskorišćenja se sastoji u poređenju vrednosti f_i i $U(n, \Delta_i)$. U slučaju kad nema zaključka, tj. kad je f_i između $U(n, \Delta_i)$ i 100%, do konačnog zaključka se može doći primenom proširenog testa vremena odziva zadatka.

5) Zadatak u toku izvršenja menja prioritet (u suštini sastoji se od niza podzadataka)

- U realnim primenama zadatak se ne izvršava samo na jednom nivou prioriteta, već u toku izvršenja on menja nivo prioriteta na kom se izvršava.
- U suštini, zadatak se sastoji od niza podzadataka, koji se izvršavaju serijski, jedan za drugim.
- Ovo je vrlo čest slučaj u praksi.

Kanonički oblik zadatka

Zadatak ima kanonički oblik ukoliko se sastoji od podzadataka čiji prioritet se ne smanjuje – algoritam kanonizacije:

Postavi $P_{im(i)}' = P_{im(i)}$ gde je m(i) broj podzadataka zadatka τ_i

Ako je ($P_{ij}' < P_{ij-1}$) onda postavi $P_{ij-1}' = P_{ij}'$

u protivnom postavi $P_{ij-1}' = P_{ij-1}$

Ponavljaj ovaj postupak pomerajući se od zadnjeg prema prvom podzadatku.

Klasifikacija zadataka

- Ostali zadaci se razvrstavaju na osnovu odnosa prioriteta njihovih podzadataka i posmatranog zadatka t_i.
- Ključan kriterijum je odnos prioriteta prvog podzadatka i minimalnog prioriteta svih podzadataka zadatka t_i, $Pmin_i$.

H i L segmenti zadataka

- Niz uzastopnih podzadataka čine segment.
- H segment čine uzastopni podzadaci čiji prioritet je veći ili jednak $Pmin_i$.
- L segment se sastoji od niza uzastopnih podzadataka čiji prioritet je striktno manji od $Pmin_i$.
- Efekat prekidanja od strane prvog segmenta koji je H segment se naziva *efekat istiskivanja*, a efekat koji se dobija kad se H segment pojavi posle L segmenta se naziva *efekat blokiranja*.

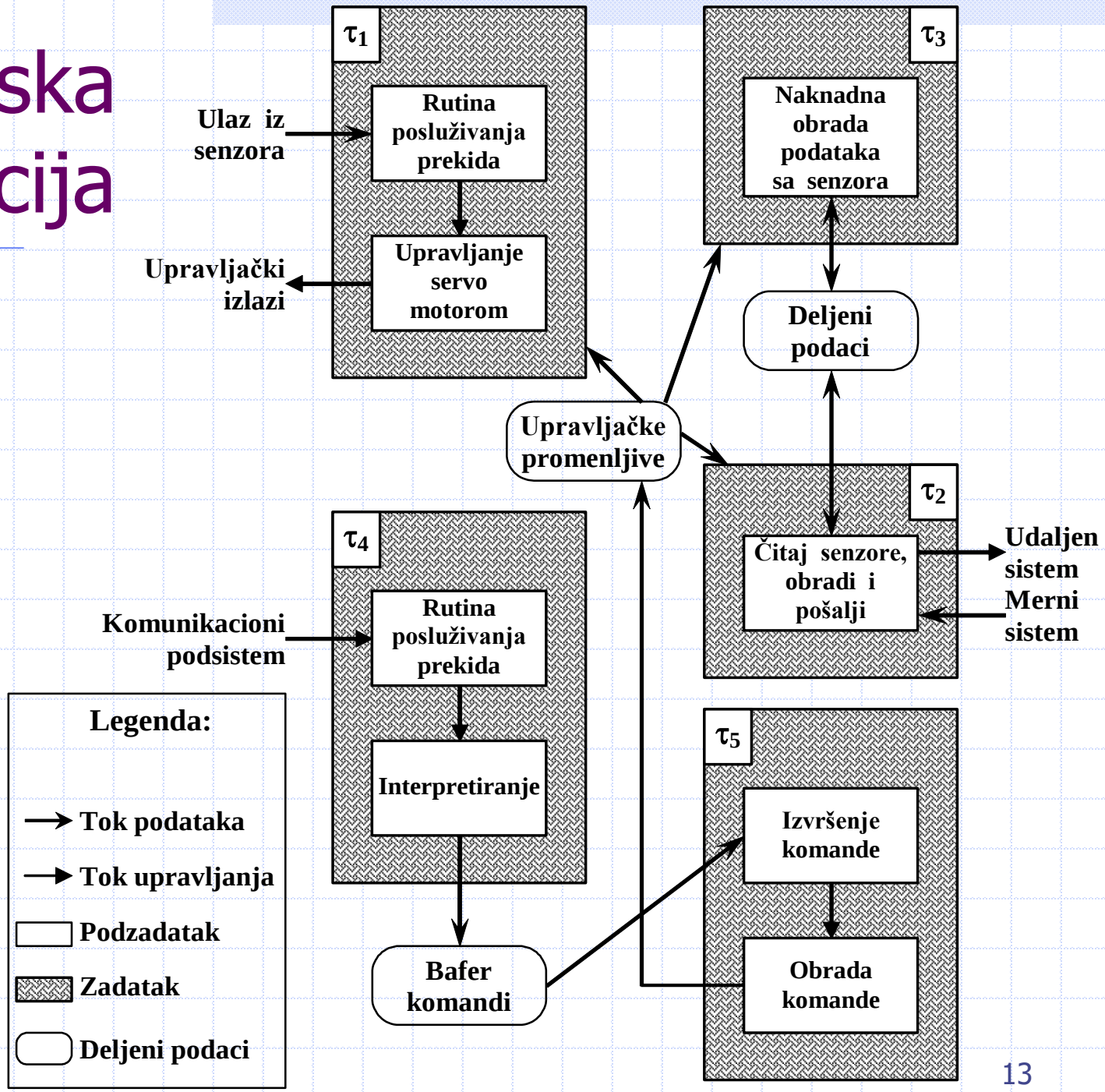
Postoji četiri tipova zadataka u odnosu na posmatrani zadatak

- Tip 1: zadaci koji imaju samo jedan H segment što znači da mogu istisnuti t_i više puta. Svi zadaci iz ove grupe određuju najgore vreme završetka t_i.
- Tip 2: kod ovog tipa, svaki H segment je praćen L segmentom, i zbog toga zadaci ovog tipa mogu prekinuti t_i samo jednom.
- Tip 3: kod ovog tipa, svakom H segmentu predhodi L segment, i zbog toga samo zadaci iz ove grupe mogu doprineti efektu blokiranja.
- Tip 4: zadaci koji imaju samo jedan L segment, i otuda ne utiču na vreme završetka t_i, i mogu se ignorisati.

Koliko puta treba uračunavati efekat blokiranja?

- Dokazano je da najviše jedan blokirajući segment (niz LH) nekog drugog zadatka može zakasniti završetak zadatka t_i.

Robotska aplikacija



Osobine zadataka za dati primer robotske aplikacije

Zadatak	T_i	C_{i1}	C_{i2}	C_{i3}	C_{i4}	D_i	P_{i1}	P_{i2}	P_{i3}	P_{i4}
τ_1	40	1	5	-	-	40	10	7	-	-
τ_2	50	7	11	2	-	50	5	8	5	-
τ_3	100	10	5	5	-	100	4	8	4	-
τ_4	200	8	18	3	2	200	9	2	3	2
τ_5	400	2	12	10	-	400	3	1	6	-

Osobine podzadataka

- Najniži nivo prioriteta svakog zadatka je dodeljen prema RM šemi.
- Zadaci t_1 i t_4 započinju rutinama za posluživanje prekida, tako da su prioriteti njihovih početnih podzadataka određeni fizičkom arhitekturom sistema.
- Zadaci t_2 i t_3 se sinhronizuju u svojim srednjim podzadacima, koji se izvršavaju na istom nivou prioriteta. Zadaci t_4 i t_5 se takođe sinhronizuju svojim trećim i prvim podzadatkom.

Kanonizacija zadatka t2’

- Zadatak t_2' ima jedan jedini podzadatak, čije vreme izvršenja je jednako sumi vremena izvršavanja originalnih podzadataka, i čiji prioritet je jednak $P_{23}=5$.
- Kanonizovan zadatak t_2' ima sledeće osobine:
  - $C_2' = 20$; $T_2' = 50$; $P_2' = 5$

Klasifikacija drugih zadataka u odnosu na zadatak t2’

Zadatak	Prioritet (relativan prioritet)	Tip zadatka
τ_1	10->7 (H->H)	H
τ_3	4 ->8->4 (L->H->L)	LHL
τ_4	9->2->3->2 (H->L->L->L)	HL
τ_5	3->1->6 (L->L->H)	LH

Maksimalno vreme blokiranja zadatka t2

- Maksimalno vreme blokiranja zadatka t_2 se može izračunati na sledeći način:

$B_2 = \max(C_{32}, C_{53}) + C41 = \max(5,10) + 8 = 18$ ms

Vreme odziva zadatka t2

- Korišćenjem proširenog testa vremena odziva zadatka dobija se sledeći rekurzivan razvoj:

$a_0 = C_1 + C_2 + B_2 = 6 + 20 + 18 = 44$ ms

$a_1 = B_2 + C_2 + \text{veći\_ceo}(a_0/T_1)*C_1 = 18 + 20 + 2(6) = 50$ ms

$a_2 = B_2 + C_2 + \text{veći\_ceo}(a_1/T_1)*C_1 = 18 + 20 + 2(6) = 50$ ms $\rightarrow$ kraj rekurzije jer je $a_2 = a_1$

- gde funkcija veći_ceo(x) zaokružuje x na prvi veći ceo broj. Pošto vreme odziva (50 ms) nije veće od krajnjeg roka zadatka (50 ms), zaključuje se da je zadatak t_2 rasporediv.

Analiza u računarskoj mreži

- Ova proširenja su nužna za distribuirane sisteme u realnom vremenu, koji se sastoje od više procesora međusobno povezanih računarskom mrežom.
- U sistemima u realnom vremenu se najčešće primenjuje FDDI (engl. token-ring) računarska mreža, kod koje je prenos paketa podataka deterministićan, tako da je moguće izračunati maksimalno kašnjenje paketa kroz mrežu.

Analiza od-kraja-do-kraja

- Tipičan scenario je npr. zadatak na telemetrijskoj stanici očitava senzor i filtrira podatak, koji zatim šalje preko mreže zadatku primarne obrade u dispečerskom centru.
- Da bi se proverilo da li zadatak primarne obrade probija svoj krajnji rok, potrebno je uvažiti vreme završetka zadatka koji očitava senzor i filtrira podatak, vreme prenosa podatka kroz mrežu i vreme završetka primarne obrade. Ovo se naziva analizom krajnjeg roka od-kraja-do-kraja.

Alati za podršku RM analizi

- Programski alati za podršku RM analizi, kao što je RAPID RMA (Tri-Pacific Corp., Alameda, CA), koriste analitički pristup za verifikaciju ograničenja realnog vremena, na osnovu teorije raspoređivanja.
- RAPID RMA analizira apstraktni model sistema.
- Projektant sistema razvija model sistema i koristi radno okruženje radi određivanja rasporedivosti zadataka, na osnovu primenjenog algoritma raspoređivanja.

NAMENA ALATA RAPID RMA

- Modeliranje zadataka, modeliranje resursa, aperiodičnih servera, sa višestrukim algoritmima raspoređivanja i višestrukim algoritmima dodele prioriteta, za analizu jednog ili više čvorova, analizu od-kraja-do-kraja i za interaktivno projektovanje.
- Između ostalih podržani su i sledeći algoritmi raspoređivanja: RM (engl. rate-monotonic), DM (engl. deadline-monotonic) i EDF (engl. earliest-deadline-first).

Interaktivno projektovanje sa alatom RAPID RMA

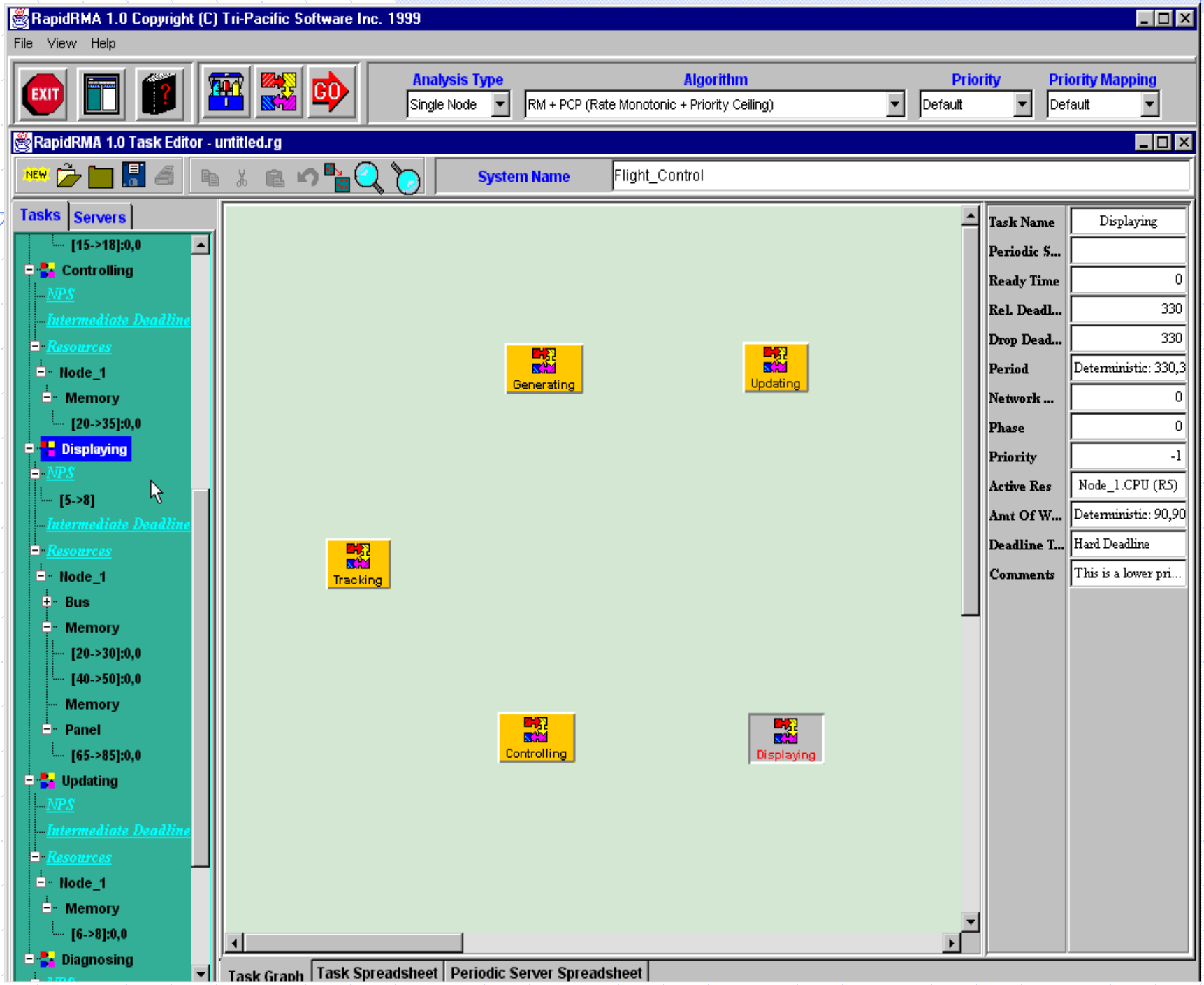
- Npr. moguće je izučavati efekte i odnose između različitih parametara sistema, koji neposredno utiču na rasporedivost celog skupa zadataka.
- Analizator ne samo da određuje da li je ceo skup zadataka rasporediv, već i identifikuje gde i zašto određeni zadaci probijaju krajnje rokove.
- Npr. alat može da ukaže na problem u modelu zadataka kao što je predugo zaključavanje nekog resursa.

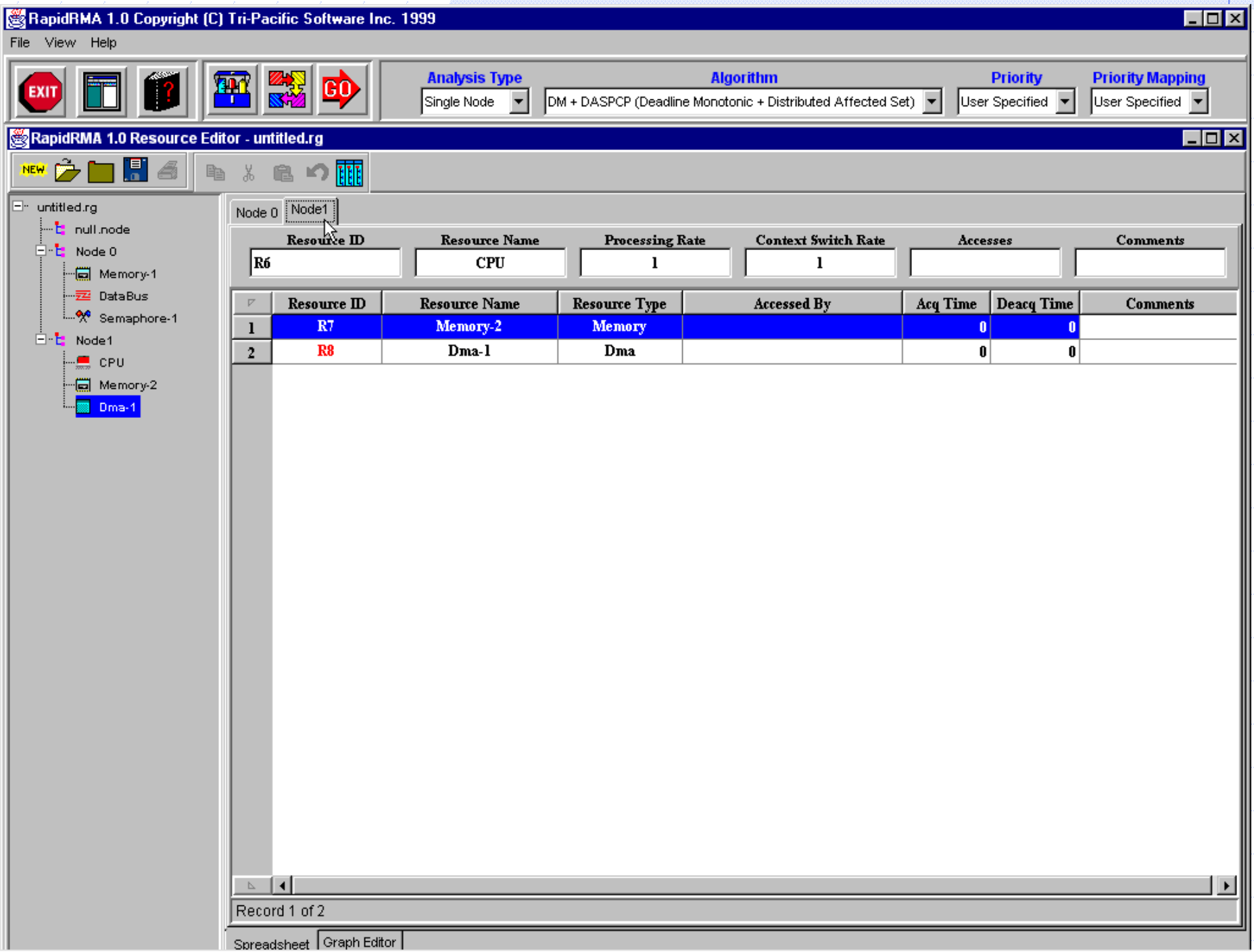
Šta-ako scenariji

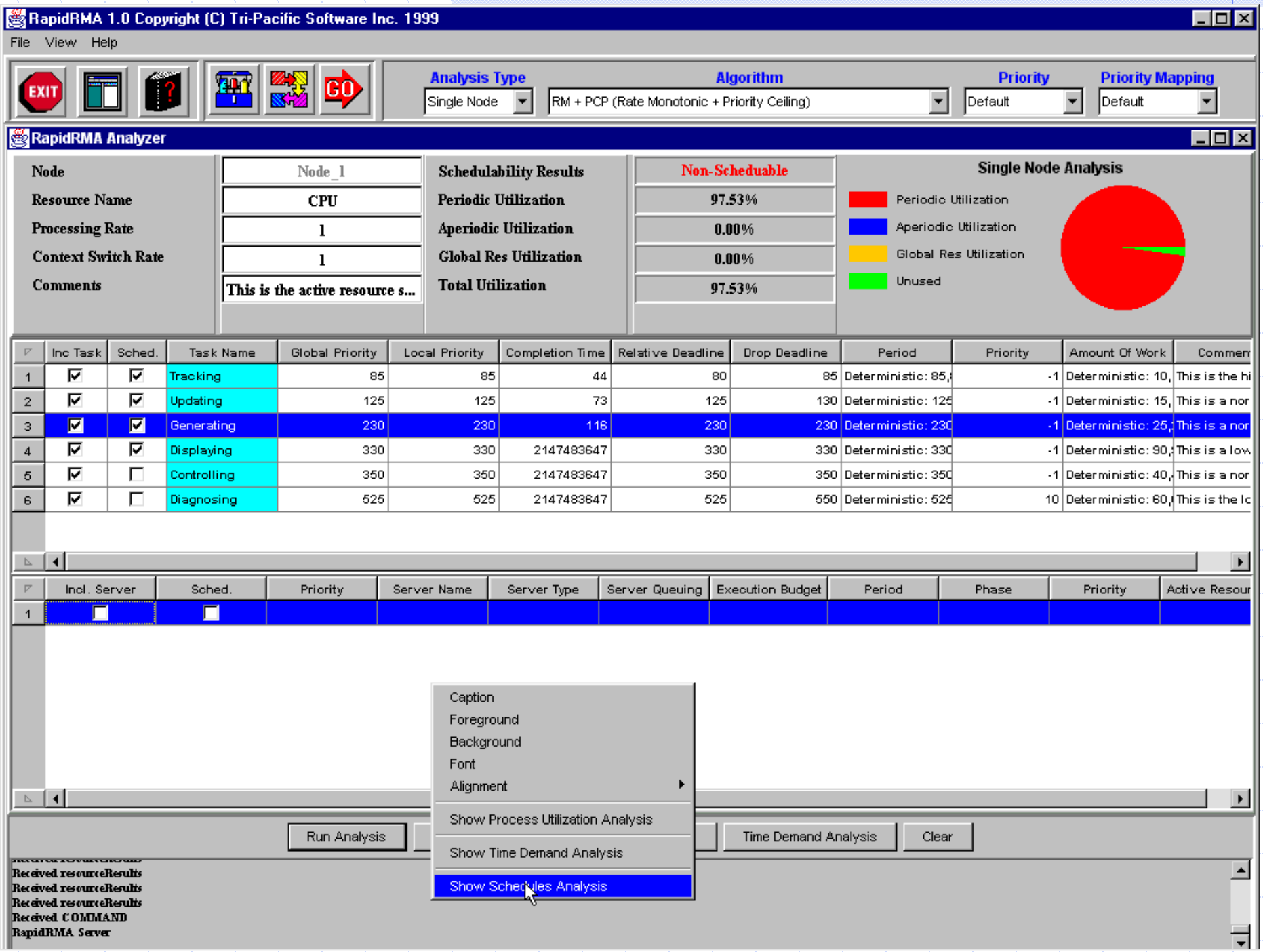
- Vrlo lako se mogu promeniti određeni parametri sistema radi poboljšanja rasporedivosti.
- Analizator takođe predviđa vremenske posledice promene parametara na nivou komponenata fizičke arhitekture i operativnog sistema.
- Pošto je alat zasnovan na analitičkom računu, rezultati su odmah raspoloživi, tako da se brzo može probati niz šta-ako scenarija.

Tri osnovna koraka u korišćenju RAPID RMA

- Formiranje modela zadataka
- Formiranje modela resursa
- Analiza rasporedivosti sistema


RapidRMA 1.0 Copyright (C) Tri-Pacific Software Inc. 1999
File View Help
EXIT
GO
Analysis Type
Single Node
Algorithm
RM + PCP (Rate Monotonic + Priority Ceiling)
Priority
Default
Priority Mapping
Default
RapidRMA 1.0 Task Editor - untitled.rg
NEW
System Name
Flight_Control
Tasks
Servers
[15->18]:0,0
Controlling
NPS
Intermediate Deadline
Resources
Node_1
Memory
[20->35]:0,0
Displaying
NPS
[5->8]
Intermediate Deadline
Resources
Node_1
Bus
Memory
[20->30]:0,0
[40->50]:0,0
Memory
Panel
[65->85]:0,0
Updating
NPS
Intermediate Deadline
Resources
Node_1
Memory
[6->8]:0,0
Diagnosing
Generating
Updating
Tracking
Controlling
Displaying
Task Name
Displaying
Periodic S...
Ready Time
0
Rel. Deadl...
330
Drop Dead...
330
Period
Deterministic: 330,3
Network ...
0
Phase
0
Priority
-1
Active Res
Node_1.CPU (RS)
Amt Of W...
Deterministic: 90,90
Deadline T...
Hard Deadline
Comments
This is a lower pri...
Task Graph
Task Spreadsheet
Periodic Server Spreadsheet


RapidRMA 1.0 Copyright (C) Tri-Pacific Software Inc. 1999
File View Help
EXIT
GO
Analysis Type
Single Node
Algorithm
DM + DASPCP (Deadline Monotonic + Distributed Affected Set)
Priority
User Specified
Priority Mapping
User Specified
RapidRMA 1.0 Resource Editor - untitled.rg
NEW
untitled.rg
null.node
Node 0
Memory-1
DataBus
Semaphore-1
Node1
CPU
Memory-2
Dma-1
Node 0
Node1
Resource ID
Resource Name
Processing Rate
Context Switch Rate
Accesses
Comments
R6
CPU
1
1
Resource ID
Resource Name
Resource Type
Accessed By
Acq Time
Deacq Time
Comments
1
R7
Memory-2
Memory
0
0
2
R8
Dma-1
Dma
0
0
Record 1 of 2
Spreadsheet
Graph Editor


RapidRMA 1.0 Copyright (C) Tri-Pacific Software Inc. 1999
File View Help
EXIT
GO
Analysis Type
Single Node
Algorithm
RM + PCP (Rate Monotonic + Priority Ceiling)
Priority
Default
Priority Mapping
Default
RapidRMA Analyzer
Node
Node_1
Resource Name
CPU
Processing Rate
1
Context Switch Rate
1
Comments
This is the active resource s...
Schedulability Results
Non-Scheduable
Periodic Utilization
97.53%
Aperiodic Utilization
0.00%
Global Res Utilization
0.00%
Total Utilization
97.53%
Single Node Analysis
Periodic Utilization
Aperiodic Utilization
Global Res Utilization
Unused
Inc Task
Sched.
Task Name
Global Priority
Local Priority
Completion Time
Relative Deadline
Drop Deadline
Period
Priority
Amount Of Work
Comment
1 Tracking 85 85 44 80 85 Deterministic: 85, -1 Deterministic: 10, This is the hi
2 Updating 125 125 73 125 130 Deterministic: 125 -1 Deterministic: 15, This is a nor
3 Generating 230 230 116 230 230 Deterministic: 230 -1 Deterministic: 25, This is a nor
4 Displaying 330 330 2147483647 330 330 Deterministic: 330 -1 Deterministic: 90, This is a low
5 Controlling 350 350 2147483647 350 350 Deterministic: 350 -1 Deterministic: 40, This is a nor
6 Diagnosing 525 525 2147483647 525 550 Deterministic: 525 10 Deterministic: 60, This is the lo
Incl. Server
Sched.
Priority
Server Name
Server Type
Server Queuing
Execution Budget
Period
Phase
Priority
Active Resour
1
Caption
Foreground
Background
Font
Alignment
Show Process Utilization Analysis
Show Time Demand Analysis
Show Schedules Analysis
Run Analysis
Time Demand Analysis
Clear
Received resourceResults
Received resourceResults
Received resourceResults
Received COMMAND
RapidRMA Server

Korišćenje RMA tokom životnog ciklusa

- RMA se može koristiti u svim fazama razvoja programske podrške radi dobijanja sve bolje procene parametara zadataka i arhitekture sistema. Tipične faze razvoja programske podrške u realnom vremenu su:
  - faza analize zahteva (engl. software requirements analysis),
  - faza preliminarnog projektovanja (engl. preliminary design),
  - faza detaljnog projektovanja (engl. detailed design), i
  - faza realizacije (engl. implementation).

Upoređenje tradicionalnog i RMA pristupa inženjeringu sistema