



Universidad Nacional de Córdoba
Facultad de Ciencias Exactas, Físicas y Naturales

Programación Concurrente

Trabajo Práctico n° 3:

Modelado de un sistema de un procesador con dos núcleos

Grupo: “No Java, no party”

Fecha: 24/06/2019

Profesores :

- ☐ Dr. Ing. Micolini, Orlando
- ☐ Ing. Ventre, Luis
- ☐ Ing. Ludemann, Mauricio

Integrantes :

- ☐ Agustinoy, Jeremías
- ☐ Ferrero, Alejandro
- ☐ Viccini, Patricio

Introducción

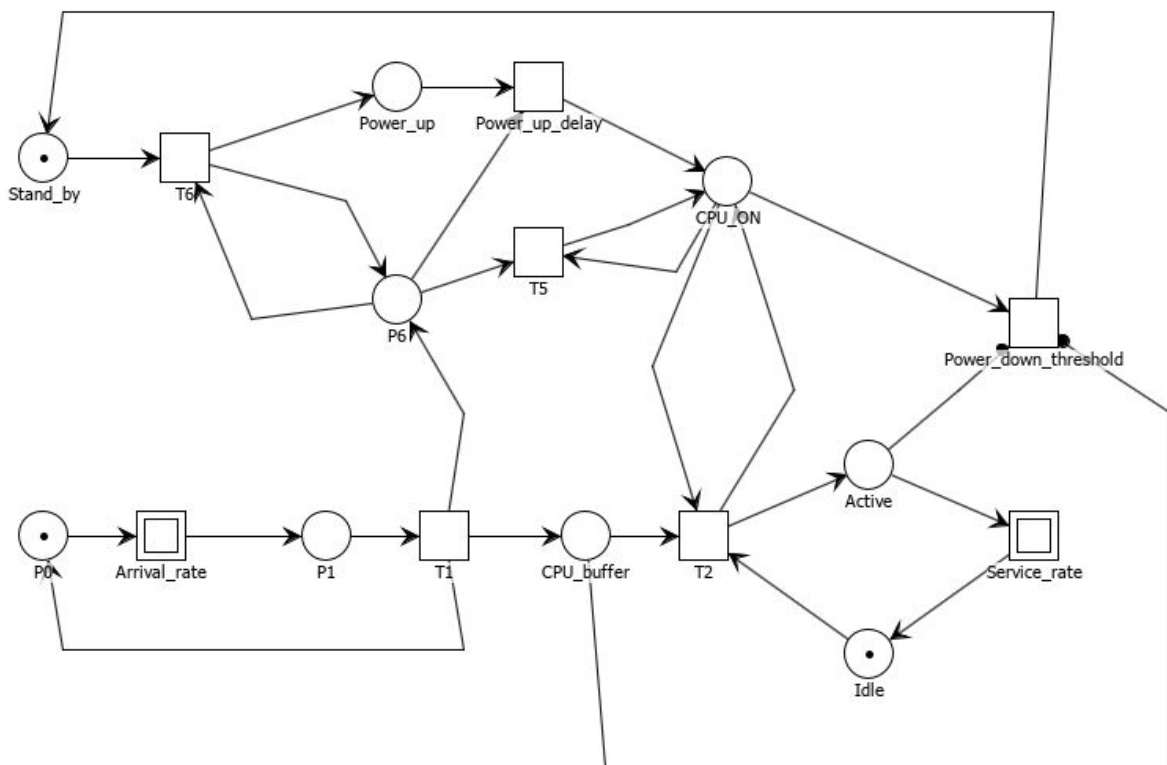
El trabajo consistió en extender un modelo de un procesador mononúcleo a un procesador de dos núcleos. Se buscó que el modelo resultante tenga las mismas propiedades que el original. La lógica del modelo que obtuvimos lo representamos de la misma forma que el del cual partimos, mediante una **red de petri temporal**.

Luego se buscó generar un programa en java que gestione la simulación de la red.

También según la consigna se debía

Red de Petri

La Red de Petri original que modela el sistema de un procesador con un núcleo es:



La cual posee las siguientes propiedades:

Petri Net Classification

Types of Petri net

State Machine	false
Marked Graph	false
Free Choice Net	false
Extended FCN	false
Simple Net	false
Extended SN	false

Mathematical properties

Bounded	false
Safe	false
Deadlock	false

Petri Net Invariant Analysis

T-Invariants

Arrival_rate	Power_down_threshold	Power_up_delay	Service_rate	T1	T2	T5	T6
1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	0

The net is covered by positive T-Invariants, therefore it might be bounded and live.

P-Invariants

Active	CPU_buffer	CPU_ON	Idle	P0	P1	P6	Power_up	Stand_by
1	0	0	1	0	0	0	0	0
0	0	0	0	1	1	0	0	0
0	0	1	0	0	0	0	1	1

The net is not covered by positive P-Invariants, therefore we do not know if it is bounded.

P-Invariant equations

$$M(\text{Active}) + M(\text{Idle}) = 1$$

$$M(P0) + M(P1) = 1$$

$$M(\text{CPU_ON}) + M(\text{Power_up}) + M(\text{Stand_by}) = 1$$

Forwards incidence matrix I^+

	Arrival_rate	Power_down_threshold	Power_up_delay	Service_rate	T1	T2	T5	T6
Active	0	0	0	0	0	1	0	0
CPU_buffer	0	0	0	0	1	0	0	0
CPU_ON	0	0	1	0	0	1	1	0
Idle	0	0	0	1	0	0	0	0
P0	0	0	0	0	1	0	0	0
P1	1	0	0	0	0	0	0	0
P6	0	0	0	0	1	0	0	1
Power_up	0	0	0	0	0	0	0	1
Stand_by	0	1	0	0	0	0	0	0

Backwards incidence matrix I^-

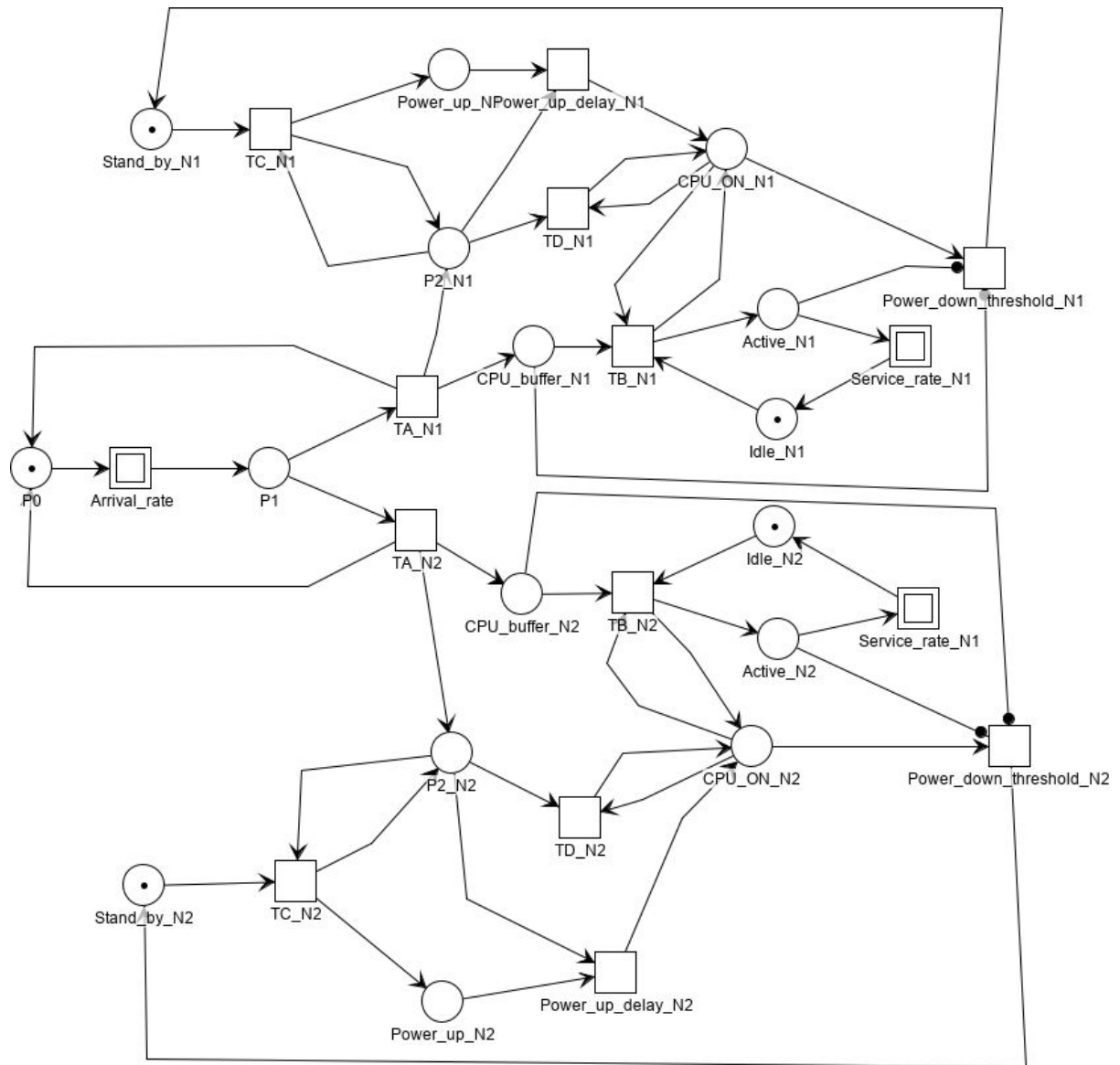
	Arrival_rate	Power_down_threshold	Power_up_delay	Service_rate	T1	T2	T5	T6
Active	0	0	0	1	0	0	0	0
CPU_buffer	0	0	0	0	0	1	0	0
CPU_ON	0	1	0	0	0	1	1	0
Idle	0	0	0	0	0	1	0	0
P0	1	0	0	0	0	0	0	0
P1	0	0	0	0	1	0	0	0
P6	0	0	1	0	0	0	1	1
Power_up	0	0	1	0	0	0	0	0
Stand_by	0	0	0	0	0	0	0	1

Combined incidence matrix I

	Arrival_rate	Power_down_threshold	Power_up_delay	Service_rate	T1	T2	T5	T6
Active	0	0	0	-1	0	1	0	0
CPU_buffer	0	0	0	0	1	-1	0	0
CPU_ON	0	-1	1	0	0	0	0	0
Idle	0	0	0	1	0	-1	0	0
P0	-1	0	0	0	1	0	0	0
P1	1	0	0	0	-1	0	0	0
P6	0	0	-1	0	1	0	-1	0
Power_up	0	0	-1	0	0	0	0	1
Stand_by	0	1	0	0	0	0	0	-1

Inhibition matrix H								
	Arrival_rate	Power_down_threshold	Power_up_delay	Service_rate	T1	T2	T5	T6
Active	0	1	0	0	0	0	0	0
CPU_buffer	0	1	0	0	0	0	0	0
CPU_ON	0	0	0	0	0	0	0	0
Idle	0	0	0	0	0	0	0	0
P0	0	0	0	0	0	0	0	0
P1	0	0	0	0	0	0	0	0
P6	0	0	0	0	0	0	0	0
Power_up	0	0	0	0	0	0	0	0
Stand_by	0	0	0	0	0	0	0	0

La Red de Petri resultante que modela el sistema de un procesador con dos núcleos es:



La cual posee las siguiente propiedades:

Resultados de la clasificación de la RdP

Types of Petri net

State Machine	false
Marked Graph	false
Free Choice Net	false
Extended FCN	false
Simple Net	false
Extended SN	false

Mathematical properties

Bounded	false
Safe	false
Deadlock	false

P - invariantes

P-Invariants

Active_N1	CPU_buffer_N1	CPU_ON_N1	Idle_N1	P0	P1	P2_N1	Power_up_N1	Stand_by_N1	CPU_ON_N2	P2_N2	Stand_by_N2	Power_up_N2	Active_N2	Idle_N2	CPU_buffer_N2
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Ecuaciones de los P - invariantes

$$M(\text{Active_N1}) + M(\text{Idle_N1}) = 1$$

$$M(P0) + M(P1) = 1$$

$$M(\text{CPU_ON_N1}) + M(\text{Power_up_N1}) + M(\text{Stand_by_N1}) = 1$$

$$M(\text{CPU_ON_N2}) + M(\text{Stand_by_N2}) + M(\text{Power_up_N2}) = 1$$

$$M(\text{Active_N2}) + M(\text{Idle_N2}) = 1$$

Matriz de Incidencia I+

Forwards incidence matrix I^+

	Arrival_rate	Power_down_ threshold_N1	Power_up_del ay_N1	Service_rate_ N1	TA_N1	TB_N1	TD_N1	TC_N1	Power_down_ threshold_N2	Power_up_del ay_N2	TD_N2	TC_N2	Service_rate_ N1	TB_N2	TA_N2
Active_N1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
CPU_buffer_N1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
CPU_ON_N1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0
Idle_N1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
P0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
P1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P2_N1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
Power_up_N1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Stand_by_N1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
CPU_ON_N2	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
P2_N2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Stand_by_N2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Power_up_N2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Active_N2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Idle_N2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
CPU_buffer_N2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Matriz de Incidencia I-

Backwards incidence matrix I^-

	Arrival_rate	Power_down_th reshold_N1	Power_up_delay _N1	Service_rate_N 1	TA_N1	TB_N1	TD_N1	TC_N1	Power_down_th reshold_N2	Power_up_delay _N2	TD_N2	TC_N2	Service_rate_N 1	TB_N2	TA_N2
Active_N1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
CPU_buffer_N1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
CPU_ON_N1	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0
Idle_N1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
P0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
P2_N1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0
Power_up_N1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Stand_by_N1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
CPU_ON_N2	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0
P2_N2	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
Stand_by_N2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Power_up_N2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Active_N2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Idle_N2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
CPU_buffer_N2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Matriz de Incidencia Combinada I

Combined incidence matrix /

	Arrival_rate	Power_down_th reshold_N1	Power_up_delay _N1	Service_rate_N 1	TA_N1	TB_N1	TD_N1	TC_N1	Power_down_th reshold_N2	Power_up_delay _N2	TD_N2	TC_N2	Service_rate_N 1	TB_N2	TA_N2
Active_N1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0
CPU_buffer_N1	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0
CPU_ON_N1	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0
Idle_N1	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0
P0	-1	0	0	0	1	0	0	0	0	0	0	0	0	0	1
P1	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	-1
P2_N1	0	0	-1	0	1	0	-1	0	0	0	0	0	0	0	0
Power_up_N1	0	0	-1	0	0	0	0	1	0	0	0	0	0	0	0
Stand_by_N1	0	1	0	0	0	0	0	-1	0	0	0	0	0	0	0
CPU_ON_N2	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	0
P2_N2	0	0	0	0	0	0	0	0	0	-1	-1	0	0	0	1
Stand_by_N2	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0
Power_up_N2	0	0	0	0	0	0	0	0	0	-1	0	1	0	0	0
Active_N2	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0
Idle_N2	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0
CPU_buffer_N2	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1

Matriz de Inhibición H

Inhibition matrix H																
	Arrival_rate	Power_down_th reshold_N1	Power_up_delay _N1	Service_rate_N 1	TA_N1	TB_N1	TD_N1	TC_N1	Power_down_th reshold_N2	Power_up_delay _N2	TD_N2	TC_N2	Service_rate_N 1	TB_N2	TA_N2	
Active_N1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
CPU_buffer_N1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
CPU_ON_N1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Idle_N1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
P0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
P1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
P2_N1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Power_up_N1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Stand_by_N1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
CPU_ON_N2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
P2_N2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Stand_by_N2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Power_up_N2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Active_N2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
Idle_N2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
CPU_buffer_N2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	

Marcado inicial

Marking																
	Active_N1	CPU_buffer_N1	CPU_ON_N1	Idle_N1	P0	P1	P2_N1	Power_up_N1	Stand_by_N1	CPU_ON_N2	P2_N2	Stand_by_N2	Power_up_N2	Active_N2	Idle_N2	CPU_buffer_N2
Initial	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0
Current	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0

Transiciones habilitadas

Enabled transitions																
	Arrival_rate	Power_down_th reshold_N1	Power_up_delay _N1	Service_rate_N 1	TA_N1	TB_N1	TD_N1	TC_N1	Power_down_th reshold_N2	Power_up_delay _N2	TD_N2	TC_N2	Service_rate_N 1	TB_N2	TA_N2	
Enabled	yes	no	no	no	no	no	no	no	no	no	no	no	no	no	no	

Siphons and Traps

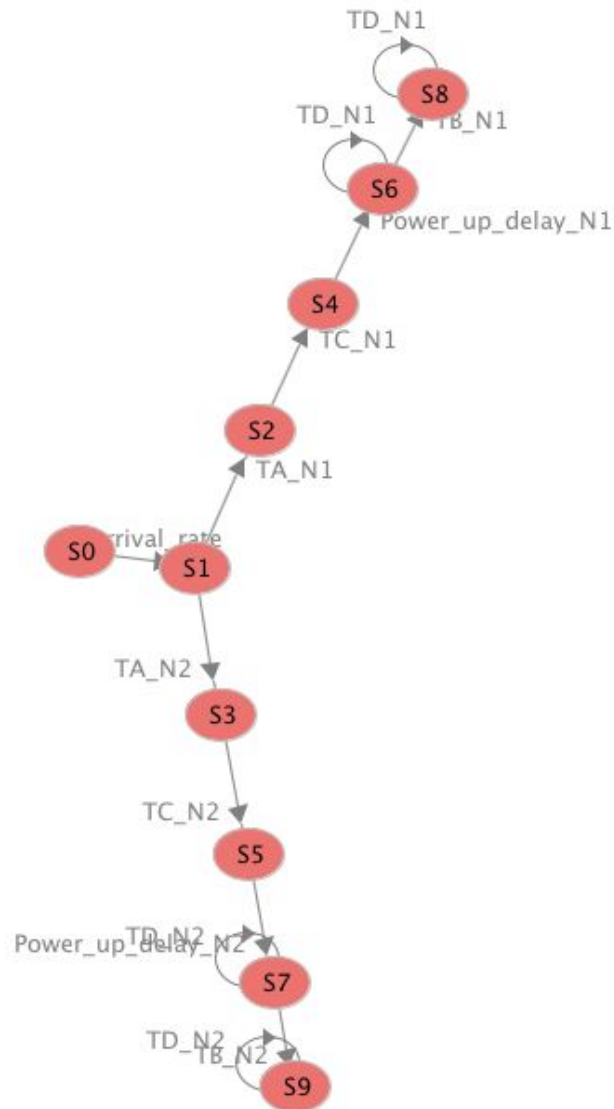
Minimal siphons

{CPU_ON_N2, Stand_by_N2, Power_up_N2 }
 {P0, P1 }
 {Active_N2, Idle_N2 }
 {CPU_ON_N1, Power_up_N1, Stand_by_N1 }
 {Active_N1, Idle_N1 }

Minimal traps

{CPU_ON_N2, Stand_by_N2, Power_up_N2 }
 {Active_N2, Idle_N2 }
 {P0, P1 }
 {CPU_ON_N1, Power_up_N1, Stand_by_N1 }
 {Active_N1, Idle_N1 }
 {CPU_ON_N1, P2_N1, Stand_by_N1 }
 {CPU_ON_N2, P2_N2, Stand_by_N2 }

Grafo de cobertura



Software

Los problemas de concurrencia se manejaron en el código mediante una clase **Monitor** y una clase **Políticas** que controlan el disparo de las transiciones de la red.

Las transiciones temporizadas se manejaron mediante la clase **TransicionConTiempo**, la misma lleva el tiempo de sensibilizado y dice si se está dentro de la ventana de tiempo del disparo, también si está antes o después.

La sensibilización de las transiciones se manejó mediante la **ecuación de estado generalizada** (se adjunta link del paper), esto para poder tener en cuenta los brazos inhibidores y las transiciones temporizadas.

Encontramos que la cantidad óptima de hilos corriendo es 5, además de main, ya que la red sólo permite que se disparen un máximo de 5 transiciones concurrentemente.

Entonces, para realizar los disparos de la red se realizaron 5 objetos que heredan de **Thread** y su función es tratar de disparar transiciones de zonas específicas de la red donde encontramos trazas que volvían a su estado anterior luego de cierta cantidad de disparos. Estos objetos además representan partes del procesador, tenemos:

Obj_Creador:

Es el que crea las tareas que llegan al procesador y las coloca en los buffers.

Dispara las transiciones *Arrival_rate*, *TA_N1*, *TA_N2*

Obj_CPU:

Es el que enciende y apaga el núcleo 1 del procesador.

Dispara las transiciones *TC_N1*, *TD_N1*, *Powe_up_delay_N1*, *Power_down_threshold_N1*

Obj_CPU2:

Es el que enciende y apaga el núcleo 2 del procesador.

Dispara las transiciones *TC_N2*, *TD_N2*, *Powe_up_delay_N2*, *Power_down_threshold_N2*

Obj_Nucleo1:

Es el que realiza las tareas del núcleo 1.

Dispara las transiciones *TB_N1*, *Service_rate_N1*

Obj_Nucleo2:

Es el que realiza las tareas del núcleo 2.

Dispara las transiciones *TB_N2*, *Service_rate_N2*

Resolución del conflicto en la entrada a los Buffers

Este conflicto consiste en que el generador de tareas quiere poner una tarea en alguno de los buffers disparando las transiciones *TA_N1* o *TA_N2*, en principio puede disparar cualquiera pero solo puede disparar una, desensibilizando la otra en el proceso.

La solución fue implementar una **política dinámica** dentro de la clase Políticas que decida cual disparar en función de la cantidad de tokens en cada buffer, buscando que el valor en ambos buffers sea el mismo.

Análisis de casos - Conclusiones

Se realizaron las siguientes ejecuciones con 1000 tareas completadas (para cada caso):

- a) Ambos núcleos con el mismo tiempo de "service_rate".
- b) Un núcleo con el doble de tiempo de "service_rate" que el otro.
- c) Un núcleo con el triple de tiempo de "service_rate" que el otro.

Se considera 50 mseg de *Arrival_rate* (tiempo entre arribos) y se obtuvieron los siguientes resultados:

a) Ambos núcleos con el mismo tiempo de "service_rate"(50 mseg cada uno).

Duración total del programa: 0 min y 59 seg.

Duración total en miliseg: 58574

Tareas completadas en N1: 0

Tareas completadas en N2: 1000

b) Un núcleo con el doble de tiempo de "service_rate" que el otro(50 mseg en núcleo 1 y 100 mseg en núcleo 2).

Duración total del programa: 0 min y 57 seg.

Duración total en miliseg: 56430

Tareas completadas en N1: 474

Tareas completadas en N2: 526

c) Un núcleo con el triple de tiempo de “service_rate” que el otro (50 mseg en núcleo 1 y 150 mseg en núcleo 2).

Duración total del programa: 0 min y 58 seg.

Duración total en miliseg: 58529

Tareas completadas en N1: 637

Tareas completadas en N2: 363

Enlaces - bibliografía

Link del paper de la Ecuación de estado generalizada:

[https://www.researchgate.net/publication/328253053_Ecuacion_de_estado_generalizada_p
ara_redes_de_Petri_no_autonomas_y_con_distintos_tipos_de_arcos](https://www.researchgate.net/publication/328253053_Ecuacion_de_estado_generalizada_para_redes_de_Petri_no_autonomas_y_con_distintos_tipos_de_arcos)