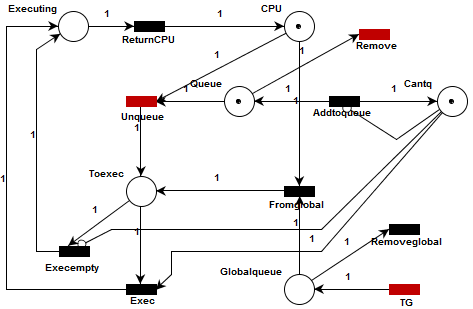
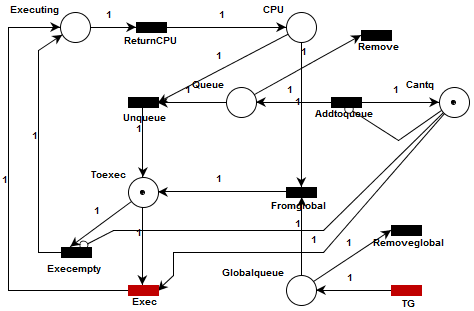


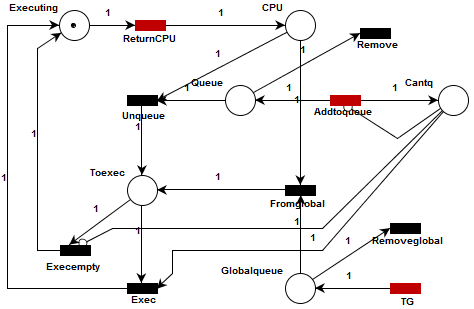
Como una situación inicial, para un determinado CPU se encuentran sensibilizadas la transición de encolado global y, para el caso en que el Cantq sea 0, la transición local de encolado al CPU.



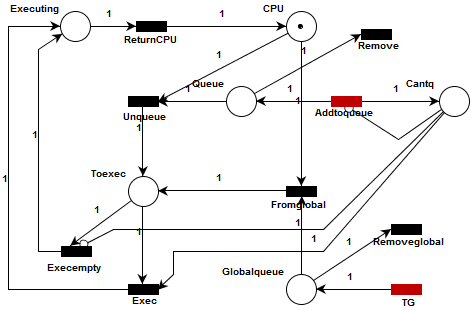
Cuando el CPU encola un hilo en su cola local, el mismo pasa a sensibilizar las transiciones de remoción de la cola y, para cuando el CPU se encuentre disponible, la transición de desencolado. También se puede ver que el encolado a la cola local deja de estar sensibilizado porque aumento en uno la plaza Cantq.



Con el CPU disponible, se desencola el hilo y pasa a estado listo para ser ejecutado. Dado que el Cantq indica que el CPU no puede encolar, la transición de ejecución que premia al CPU actual pasa a estar sensibilizada.



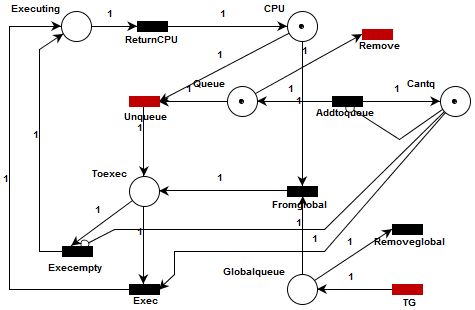
Cuando el hilo pasa a ejecución, se dispara la transición de ejecución y se indica mediante un token (executing) su situación actual. Esto permite que solamente cuando un hilo pasa a ejecución se habilite la transición para retornar el CPU. A su vez se puede ver que el CPU puede encolar nuevamente ya que el mismo fue premiado retirándole un token de la plaza Cantq al haber ejecutado un hilo y demostrar que se encuentra activo.



Finalizada la ejecución, se retorna el CPU y se vuelve a la situación inicial.

Durante la secuencia anterior el hilo pudo haber tomado dos caminos alternativos de acuerdo a la situación de la red.

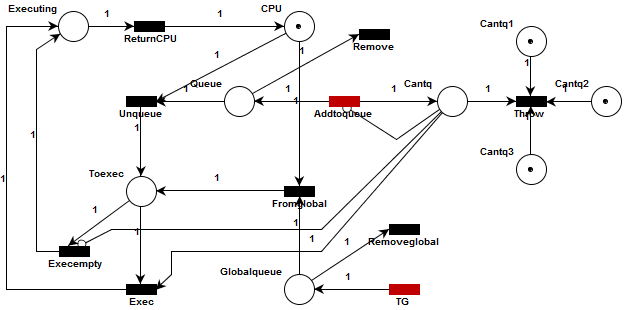
Remoción de la cola local:



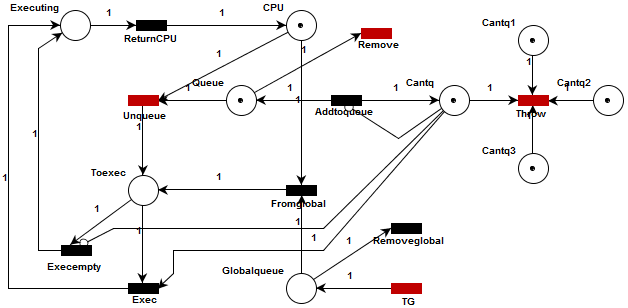
Con el hilo en la cola, el mismo puede ser sacado de la misma para ser encolado nuevamente en una cola de CPU diferente.

Ejecución sin premiar al CPU:

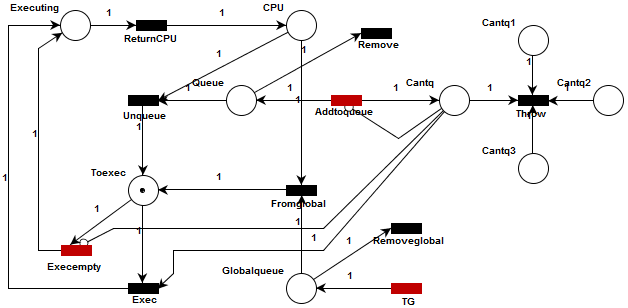
Para esto se va a extender la situación inicial incorporando la transición Throw y los otros CPUs.



Suponiendo una situación inicial donde el resto de los CPUs tienen un hilo en su cola, al encolar en el CPU actual se habilitará la transición Throw.

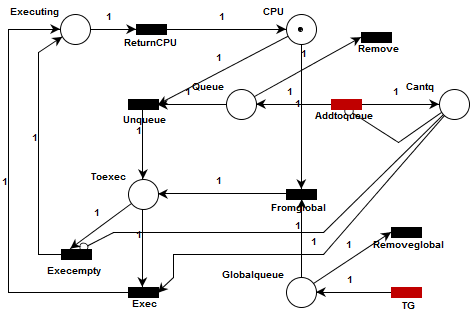


Al encolar, se habilita la transición automática de descarte para premiar a todos los CPUs por haber encolado de forma pareja. En esta situación podemos avanzar descartando los token de los Cantq y desencolar el hilo en el CPU actual.

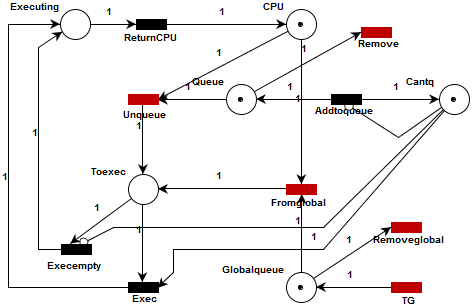


Al haber descartado los token, los CPUs que tengan su Cantq en 0 pasarán a estar nuevamente habilitados para encolar. A su vez, el hilo desencolado pasara a ejecución sin premiar al CPU ya que tal premio no es necesario. De esta manera, se sensibiliza el camino de ejecución alternativo.

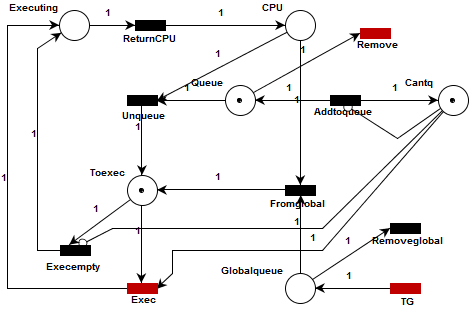
Si ahora volvemos a la situación inicial, también podemos llevar a cabo la secuencia de un hilo que se encola en la cola global, lo toma un CPU y lo manda a ejecución.



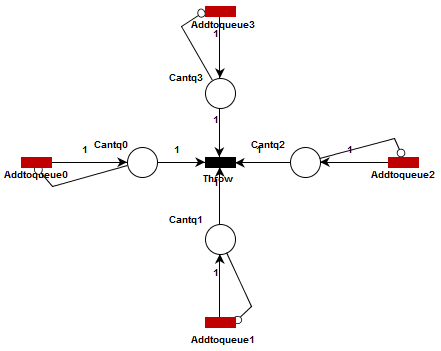
En este caso, podemos suponer que el CPU encola un hilo en su cola local y que a su vez otro hilo es encolado en la cola global.



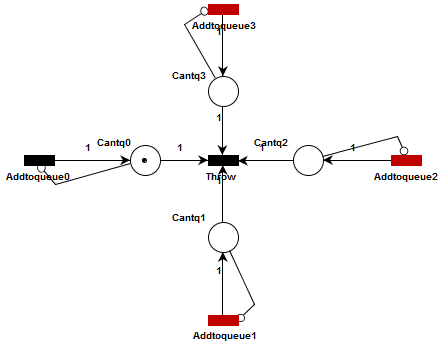
Podemos ver ahora que, habiendo hilos encolados tanto en la cola local como en la global y estando presente el CPU, pasan a estar sensibilizadas las transiciones de desencolado tanto de la cola local cola la preveniente desde la global. También están sensibilizadas las transiciones de remoción. A la hora de seleccionar un hilo para ejecutar, se fijara en ambas colas y elegirá el de mayor prioridad.



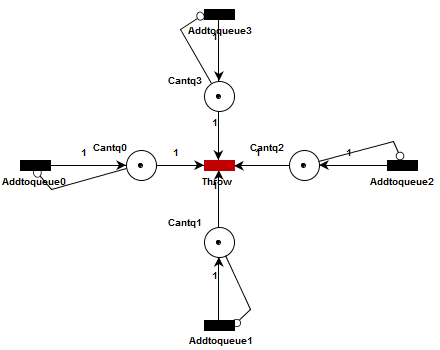
Suponiendo que seleccionó el hilo de la global, la transición de desencolado local queda deshabilitada al no estar presente el CPU y pasa a estar habilitada la de ejecución para el hilo tomado.



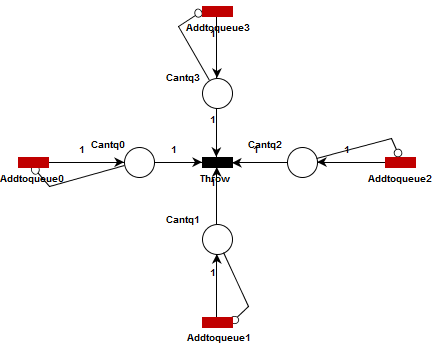
En esta situación, se mostrará el funcionamiento del encolado en los diferentes CPUs. Inicialmente, y cuando todos los CPUs están habilitados, todas las colas de encolado están sensibilizadas para encolar los hilos que vayan llegando.



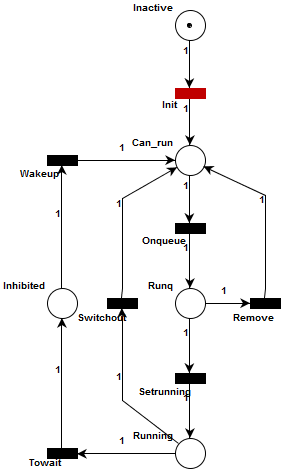
El hilo actual es encolado en el CPU0, lo cual deshabilita a ese CPU para encolar. De esta manera, al llegar otro hilo solo los otros podrán encolarlo. Así se logra una repartición de hilos pareja entre las colas de los CPUs.



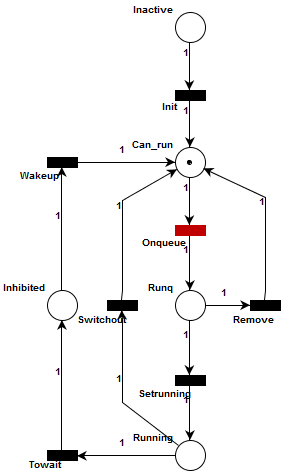
En este momento todos los CPUs han encolado un hilo, por lo que la transición de descarte se habilita para permitir que os CPUs puedan volver a encolar.



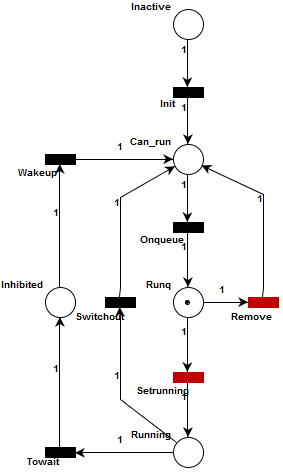
Se puede ver como se vuelve a la situación inicial, por lo que se logró una repartición equitativa entre todos los CPUs de los hilos pendientes a ser encolados.



Cada hilo posee una red propia que modela todos los estados posibles en lo que puede encontrarse. Inicialmente, al inicializar un hilo el mismo se encuentra en estado inactivo esperando ser seteado para poder correr.

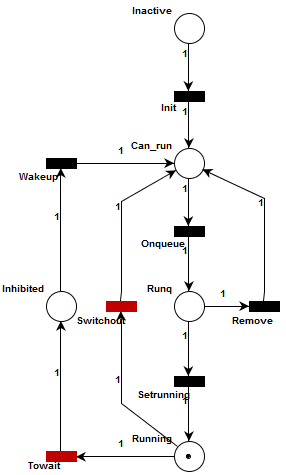


Una vez inicializado, el hilo es pasado por el sistema operativo al estado en que puede correr. El mismo queda en espera a que el scheduler lo agregue a una cola para esperar su turno de ejecución.



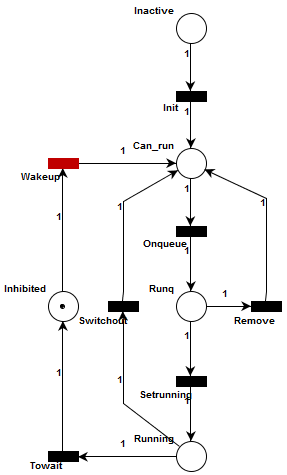
Una vez ubicado en alguna cola, el hilo puede seguir dos caminos:

* Que el CPU lo elija para ejecutarlo, esto va a darse cuando su prioridad sea mayor al resto de los hilos esperando en la cola.
* Que el scheduler decida sacarlo de su cola actual y ubicarlo en una nueva.

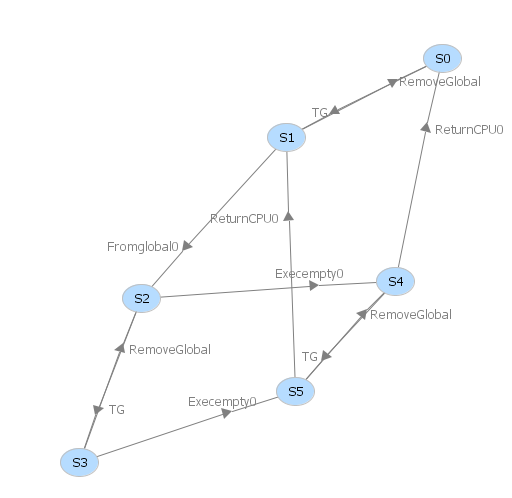


Cuando el hilo toma el CPU y pasa a ejecución, se presentan 2 situaciones:

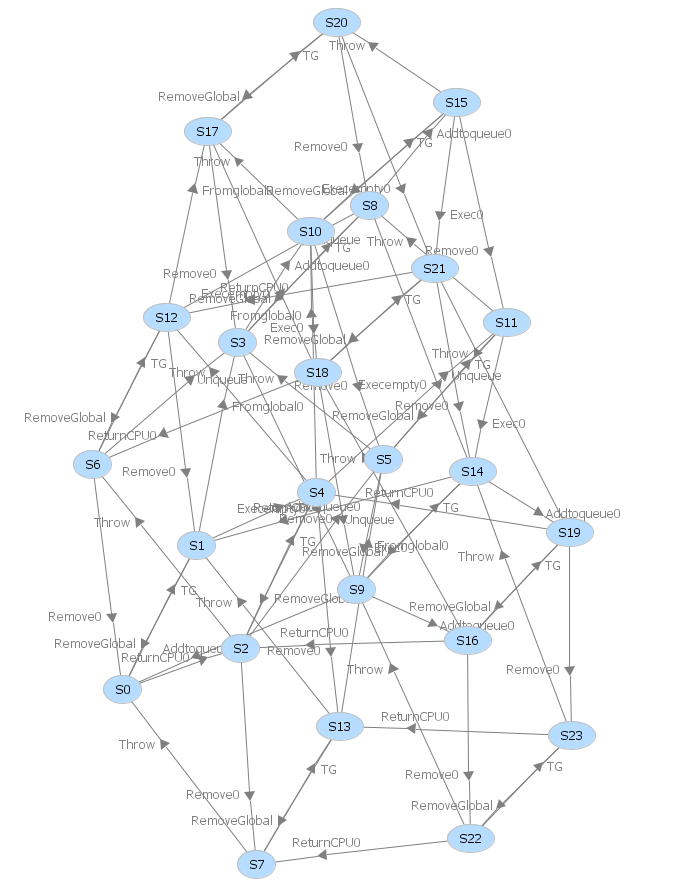
* Retorno de CPU involuntario: este caso se da cuando el hilo consume si time slice de ejecución. Llegado este momento abandonará en CPU y pasara nuevamente al estado esperando a ser encolado por medio de la transición switch out.
* Retorno de CPU voluntario: este caso se da cuando se produce una interrupción que hace que el hilo abandone su ejecución antes de finalizar su tiempo, pasando al estado de inhibición por medio de la transición to wait.



Para el caso del retorno voluntario, el hilo queda esperando en el estado inhibido hasta que el sistema operativo lo despierte. Llegado ese momento, el hilo pasa l=al estado de poder ser encolado nuevamente por medio de la transición de despertar.

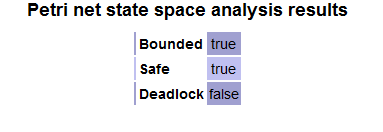


Dibujo para un CPU con smp\_not\_started

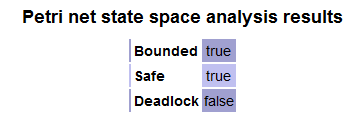


Dibujo para un CPU con smp\_started

## 

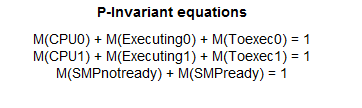


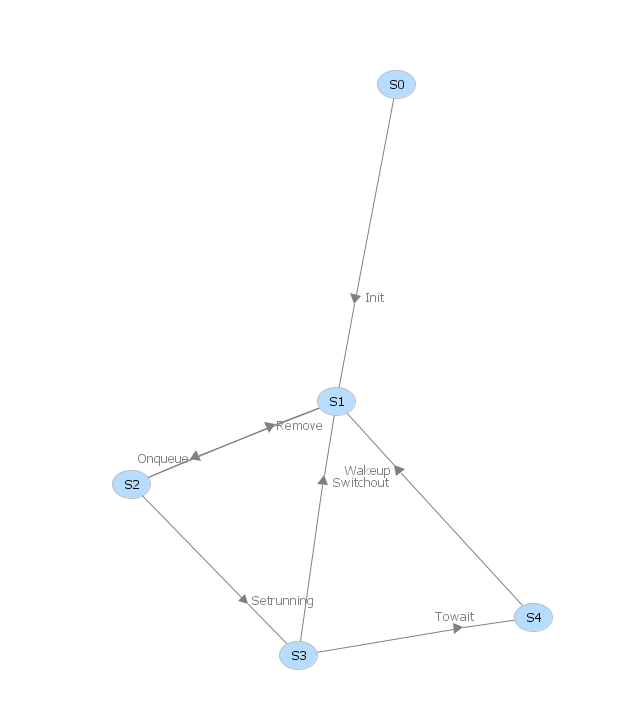
Para un CPU con smp\_started colas limitado a 1



Para 2 CPUs con colas limitado a 1

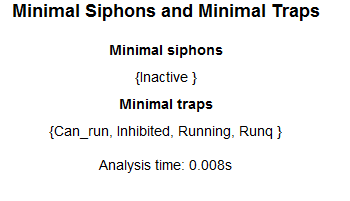
Invariantes:

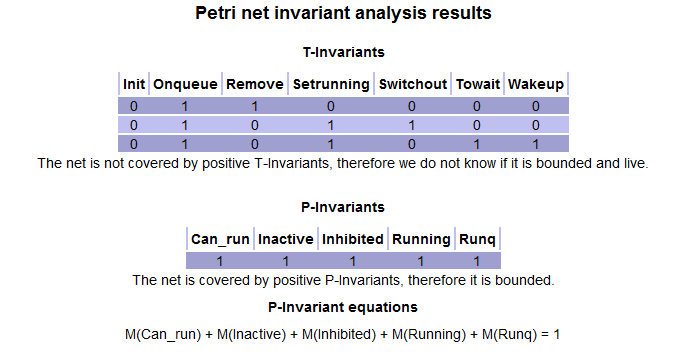




Para red de petri del thread

Sifones y trampas:





Invariantes RDP del thread