

## SIMULACIÓN DE PROYECTOS MEDIANTE COLAS Y SERVIDORES

ENRIQUE G. BAQUELA

Grupo de Investigación en Simulación y Optimización Industrial (GISOI)-Facultad Regional  
San Nicolás-Universidad Tecnológica Nacional - ARGENTINA  
[ebaquela@frsn.utn.edu.ar](mailto:ebaquela@frsn.utn.edu.ar)

*Fecha Recepción: Diciembre 2010 - Fecha Aceptación: Agosto 2012*

### RESUMEN

En este trabajo se propone un marco genérico para simular la ejecución de proyectos (o secuencia de tareas en general) con recursos acotados. En base al mismo se pueden evaluar distintas políticas de configuración y priorización de tareas de manera sencilla.

**PALABRAS CLAVE:** Simulación – Programación – Proyecto – Eventos discretos – Cola – Servidor - Recurso

### ABSTRACT

In this paper, a generic framework to simulate the execution of projects (or sequence of tasks in general) having limited resources is proposed. Based on it, diverse policies of configuration and prioritization can be easily evaluated.

**KEYWORDS:** Simulation - Schedule - Project – Discrete events – Queue – Server - Resource

### 1. INTRODUCCIÓN

En general, la programación de proyectos suele prestar atención a la programación de la ejecución de las tareas y posteriormente a la adaptación del volumen de recursos necesarios. Sin embargo, una vez que el proyecto está en marcha, no se puede modificar el volumen de recursos tan fácilmente, por lo cual, a fines de la programación, se vuelven más importantes los recursos críticos que el camino crítico. Esto también puede ser cierto en la etapa de planificación “pre-inicio” del proyecto cuando los recursos que dispone el ejecutor del proyecto son muy acotados. Carece un poco de lógica, en estos casos, asignar a las tareas del proyecto tiempos arbitrarios y luego, a través de estándares, calcular el nivel de recursos necesarios. Es aquí donde la teoría de colas y la simulación por eventos discretos pueden prestarnos gran ayuda.

Por lo tanto, se presenta a continuación un modelo de ejecución de proyectos focalizado en la determinación de la criticidad de los recursos vía simulación por eventos discretos.

## **2. DESARROLLO**

### **2.1 Conceptos:**

#### 2.1.1 Modelado del proyecto:

La base de la metodología aquí presentada consiste en abstraer la ejecución de un proyecto a un proceso estocástico modelizable en términos de colas y servidores. Para lo cual, previamente, es necesario definir cuales serán los servidores y cuales los clientes de nuestro proceso:

- Servidores
  - Unidades operativas
- Clientes o entidades transitorias
  - Tareas
  - Recursos genéricos

En forma simplificada, cada unidad operativa (cuadrilla, recursos con capacidad limitada, etc.) se comporta como un servidor, denominándose aquí "Servidores de Proceso". Las tareas se comportan como los clientes de los servidores anteriores, siendo su procesamiento la abstracción de la ejecución de la misma (resultando esto en que el tiempo de procesamiento de cada servidor de procesamiento es variable respecto de la tarea).

Los recursos que no califican como unidades operativas son considerados como clientes de segundo grado, es decir, necesarios para que un servidor de proceso pueda servir a una tarea, pero no procesables por si mismos.

#### 2.1.2 Simulación de la ejecución del proyecto:

Para poder simular la ejecución del proyecto, es necesario recurrir al siguiente modelo de colas y servidores:

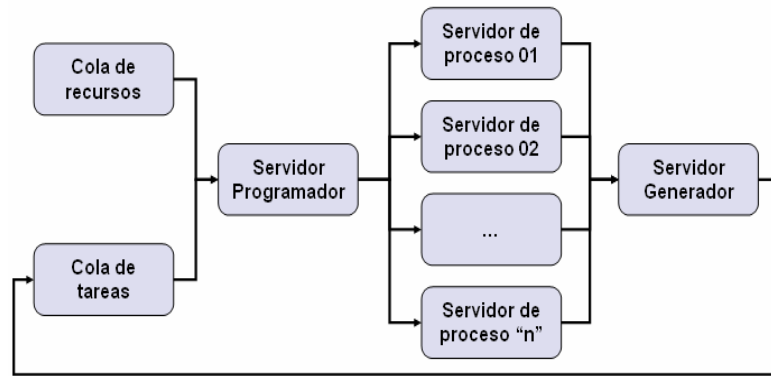


FIGURA 1

En este esquema, las tareas son entidades transitorias con una propiedad "Tipo" que la define como tarea y no como recurso, y un estado denominado "Ejecutado" que vale "NO" antes de su procesamiento y "SI" posteriormente a él. Al inicio de la simulación, las tareas que no tienen tareas predecesoras están disponibles en la cola de tareas.

Respecto de los recursos, muchos van a estar disponibles al inicio de la simulación, otros van a estar disponibles según un calendario de disponibilidad y un tercer grupo arribará según los eventos que se vayan presentando en la simulación.

Los servidores de procesamiento simulan la ejecución de la tarea por una unidad operativa concreta. El tiempo de ciclo para ese procesamiento puntual es equivalente a la duración de la ejecución de la tarea. El producto que entregan los servidores consiste en la misma tarea con su estado "Ejecutada" con el valor "SI".

Además de los servidores de procesamiento, se agregan al sistema dos servidores extras, el "Servidor Generador" y el "Servidor Programador". El primero es el encargado de asegurar el cumplimiento de las relaciones de precedencia y sucesión entre las tareas del proyecto. Cada vez que recibe una tarea con estado finalizado por parte de un servidor de procesamiento, verifica en una tabla interna de condiciones cuales de las tareas no generadas están habilitadas para su ejecución debido a sus condiciones de precedencia. Si encuentra tareas que ya cumplen todas las condiciones, las genera y las envía a la cola de tareas.

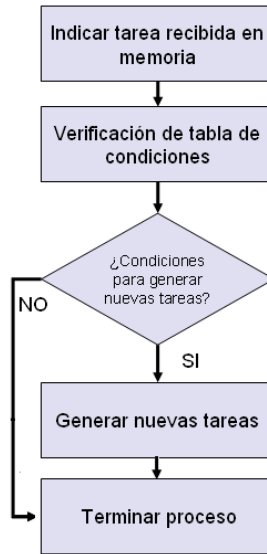


FIGURA 2

El servidor programador es el encargado de asignar tareas y recursos a servidores de procesamiento. Cada vez que un servidor procesador se desocupa, envía una señal al procesador, el cual verifica si alguna tarea se puede procesar, es decir, si alguno de los servidores libres puede procesar esa tarea y, además, se dispone de la cantidad de recursos suficientes. En dicho caso, la tarea, con los recursos asociados, se asocia al servidor.

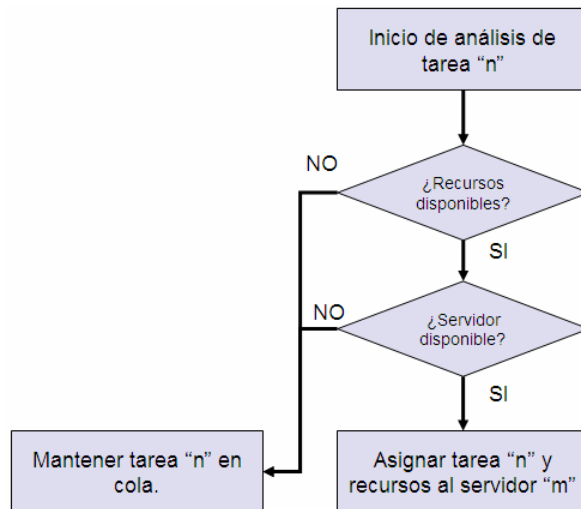


FIGURA 3

Obviamente, la lógica de asignación de tareas a servidores puede parametrizarse de manera tal de reflejar el proceso de asignación en forma lo mas real posible. En principio, el servidor programador debería elegir el procesador en función de alguna propiedad que presente el mix tarea-servidor, como por ejemplo, el tiempo de ejecución, o la fiabilidad en la realización de la misma.

La cola de tareas funciona en forma atípica. Matemáticamente, obedece más a una definición de conjunto que de cola, ya que no respetan ningún criterio de priorización para su envío a los servidores de procesamiento, sino que es el servidor procesador quien las selecciona para procesarlas. Esta selección puede estar basada en un factor de importancia de las tareas, pero también en criterios externos, como por ejemplo que no se puedan ejecutar en una determinada época del año. Estas lógicas podrían programarse en el servidor programador o ser un atributo dinámico de las tareas.

De manera similar funciona la cola de recursos, los mismos pueden dejar de estar disponible durante algún periodo de tiempo, o bien puede ser perecederos.

El inicio y el fin de la simulación están regulados por el servidor generador.

#### 2.1.3 Entradas del sistema simulador:

Además de modelar el conjunto de tareas, recursos y unidades operativas, y asociar la disponibilidad de los últimos a un eje temporal, es necesario ingresar al sistema la información relativa al procesamiento de las tareas. Esta información incluye:

- Tareas que puede procesar cada servidor
- Tiempos de procesamiento de cada tarea en servidor
- Necesidad de recursos para procesar cada tarea
- Red de precedencia y sucesión de tareas

### **2.2 Aceleración de proyectos**

Dentro de las técnicas convencionales de programación de proyectos existe una denominada CPM-Cost que trata acerca de la aceleración de tareas puntuales incrementando los factores productivos de los mismos. Este concepto puede incluirse en la simulación mediante la asociación de un factor de "potencia" a los servidores de procesamiento. Dicho factor establece una relación funcional con el tiempo de procesamiento de todas las tareas procesables por dicho servidor. Dicha potencia puede utilizarse para representar:

- Tamaño de cuadrilla o grupo de trabajo.
- Incremento monetario de factores de producción.
- Aumento de la flota de equipos.
- Mejora en la productividad de la fuerza de trabajo.
- Aumento de la cantidad de horas laborales.

El tratamiento de esta potencia puede ser estático o dinámico. Es estático si se define una potencia fija al inicio de la simulación, pero dinámico si se permite que la misma sea variable en el tiempo.

### **2.3 Aleatoriedad en la duración del proyecto**

La aleatoriedad de la duración de un proyecto puede deberse a cualquiera de las siguientes causas:

- Aleatoriedad en la duración de las tareas.
  - Aleatoriedad de los tiempos de procesamiento frente a una potencia de servicio fija.
  - Aleatoriedad en la potencia del servidor.
- Aleatoriedad en los tiempos de inicio de las tareas.
- Factores externos.

En cualquiera de los casos, el tratamiento de estos factores en una simulación es mucho más realista que mediante las metodologías analíticas comúnmente utilizadas. En general, se suele aleatorizar un proyecto mediante la utilización de técnicas de Montecarlo sobre las duraciones de cada tarea. Al definir nuestro sistema como un conjunto de servidores procesadores de tareas podemos programar eventos aleatorios más complejos.

### **2.4 Salidas del sistema**

Simulando la ejecución del proyecto, podemos obtener, entre otras, las siguientes salidas del mismo:

- Calendario probabilístico de inicio y fin de cada tarea del proyecto.
- Camino crítico probabilística
- Recursos críticos
- Cadena crítica probabilística
- Cuellos de botella en servidores y recursos
- Utilización de las unidades operativas
- Análisis de periodos críticos en la utilización de las unidades operativas y recursos
- Flujo de fondos de proyecto simulado.

### **2.5 Utilización del modelo**

El modelo se puede utilizar para, entre otras, las siguientes tareas:

- Evaluación de mix entre duración del proyecto y costo para distintas configuraciones de unidades operativas.
- Análisis de alternativas de aceleración mediante modificación de mix de unidades operativas disponibles.
- Análisis del impacto de cambios en la disponibilidad de unidades operativas y recursos.
- Adecuación del proyecto en función de la disponibilidad de fondos.
- Optimización de la asignación de tareas a las unidades operativas.
- Predicción estocástica de la duración del proyecto.

## 2.6 Resumen de los beneficios de la metodología

Esta metodología brinda una estructura conceptualmente simple pero que permite abarcar toda la complejidad de la dinámica de la ejecución de un proyecto, permitiendo realizar previsiones y análisis del tipo “¿Qué pasa si...?” muchos mas realistas que la metodología Pert/CPM habitual. Además, si bien requiere mayores esfuerzos de planificación, permite conseguir una comprensión mucho mayor del proyecto a ejecutar / en ejecución.

Es especialmente útil durante la reparación de programas, donde ya se tiene un conocimiento previo del sistema a simular, pero se debe modificar la estrategia de ejecución “in situ” debido a los imprevistos acaecidos. También permite analizar y estimar en forma mas realista el flujo de fondos del proyectos, al agregar mas detalle al modelo de lo que suele incluir una simulación de flujo de fondos por montecarlo (el método más utilizado para estas actividades).

## 2.7 Líneas de investigación potenciales

El paso siguiente al desarrollo de un modelo de simulación de la ejecución de un proyecto es integrar el mismo a un motor de optimización, que permita determinar la asignación óptima de tareas a recursos, para minimizar la duración del proyecto o los costos del mismo.

A través de la aplicación de metaheurísticas, combinadas con este enfoque de modelado, es posible obtener soluciones que permitan mejorar la performance del equipo productivo.

## 3. CONCLUSIÓN

Si bien es pasible de muchas ampliaciones, el modelo desarrollado está basado en una estructura conceptual sencilla que hace fácil el entender y aplicarlo en situaciones reales y ámbitos de enseñanza. Sin embargo no abandona, en aras de dicha sencillez, su robustez y versatilidad. Bien aplicado, puede ser una herramienta muy útil para los gerentes de proyecto.