Simulador online de red Omega y Baseline

Materia: Arquitectura de computadoras y técnicas digitales.

Integrantes: - Guerrero, Adrian Pablo.
- Oses, Cristian.

Introducción:

Como trabajo final de la materia, se nos encargó la realización de un simulador online de una red Omega o Baseline. Dicho simulador debía implementar una red de interconexión bloqueante bidireccional con switches crossbar de 2x2 y conexiones entre etapas compatibles con redes Omega y Baseline. Además, se deberían poder configurar los siguientes parámetros:

- Número de procesadores.
- Procesadores Activos (Inician requerimientos a memoria, que deben ser respondidos por la misma).
- Periodicidad de los requerimientos (Única vez, periódico, al azar).
- Configuración de dirección de memoria (dirección única, direcciones al azar, lista de direcciones).

A su vez, se deberían tener en cuenta los distintos conflictos que pudieran llegar a ocurrir en cada unos de los switches, con respecto al envío de mensajes desde los procesadores hacia la memoria.

Un requerimiento imprescindible para el trabajo era que corriera vía web, de manera que cualquier alumno pueda probarlo sin necesidad de instalación, y solo accediendo desde la página principal de la materia.

Desarrollo:

Al momento de comenzar a desarrollar el simulador, la primera toma de decisión fue acerca de la tecnología que se iba a utilizar para su desarrollo. Como hasta el momento no habíamos tenido la oportunidad de desarrollar software que corriera vía web, se estudiaron las opciones que estaban a nuestra disposición, y finalmente se optó por desarrollarlo con *VueJs* (https://vuejs.org/), un framework progresivo que transcribe Javascript.

Una vez decidido esto, se debió analizar qué estructura principal iba a tener nuestro simulador, es decir, que componentes o inputs se iban a colocar en la pantalla principal del simulador, que opciones distintas de selección se le daría al usuario, cuales se deberían completar manualmente, entre otras.

Para esto, se debió explotar al máximo dos cualidades principales que presenta el framework de *VueJs*:

- Reactividad: Vue presenta un sistema de reactividad que utiliza objetos JavaScript simples y una representación optimizada. Cada componente realiza un seguimiento de sus dependencias reactivas durante su renderizado, por lo que el sistema sabe con precisión cuándo volver a renderizar y qué componentes volver a renderizar.
- Componentes: Los componentes Vue extienden elementos HTML básicos para encapsular código reutilizable. En un nivel alto, los componentes son elementos personalizados a los que el compilador de Vue asocia el comportamiento. En Vue, un componente es esencialmente una instancia de Vue con opciones predefinidas. Para nuestro simulador en particular cada crossbar, conexión, procesador,

slot de memoria, etapa, puerto, memoria corresponde a un componente Vue.

Como se mencionó anteriormente, en nuestro sistema de interconexión se podrán graficar completamente redes Baseline, u Omega, cada una con la cantidad de procesadores que se desee. Para esto, el usuario deberá seleccionar un número de procesadores y una red, lo cual inmediatamente dibujara el cuerpo de la red, que nos permitirá seleccionar todos los requerimientos o características propias del procesador, de la red propiamente dicha, y del envío de mensajes que se consideren necesarios. Dichas configuraciones serán brindadas de manera selectiva o para completar manualmente, cada una de ellas dependiente de la opción inmediatamente precedente.

Una vez seleccionadas las características de la red y de los envíos de mensajes, se contará con dos botones, "*Paso siguiente*" y "*Reset*". El primero se encargará de mostrarnos cada paso de la simulación (cabe destacar que cada click en el botón corresponde a cada movimiento o paso del mensaje dentro de la red, desde su salida del procesador hasta su llegada a memoria, y no a cada ciclo de reloj). El segundo botón, se encarga de resetear completamente el simulador, llevándonos a la pantalla principal la cual nos permitirá realizar una nueva simulación, completamente indiferente de la anterior.

En cuanto a los conflictos que pueden aparecer dentro de la red, se observó el caso de que dos mensajes pueden arribar al switch de manera simultánea, por lo que deberíamos optar por cierto criterio para poder mostrar por

pantalla dicho conflicto, y realizar una solución que permitiría continuar con el proceso de simulación. Dicha solución, fue estudiada y realizada de la siguiente manera:

- Cuando llega un único mensaje al switch, continúa su proceso sin producir ningún tipo de conflicto.
- Si arriban dos mensajes de manera simultánea, ocupando ambas entradas del switch, el mensaje que ocupa el puerto más alto pasa a la salida del switch, y el mensaje que se encontraba en la entrada inferior del switch pasa al buffer alojado dentro del procesador. Cabe destacar, que previo al comienzo de la simulación, el usuario debe completar un input que hace referencia al tamaño deseado de dicho buffer.
- Cada vez que saco un mensaje alojado en el buffer, si existe otro mensaje en uno de los puertos del switch, este pasa a almacenarse dentro del buffer para poder liberar una entrada del switch.

Conclusión:

Como estudiantes avanzados de la carrera de Ingeniería en Sistemas, creemos que la utilización de este tipo de software conlleva a que los alumnos obtengan un enfoque aún más profundo de una unidad o tema en particular que se dicte en la materia, permitiéndoles realizar una práctica más detallada en el tema, con componentes que se asemejan a los existentes en la vida real.

En nuestro caso particular, la realización de este trabajo nos dio la oportunidad de crear software que corre via web lo cual no habíamos tenido la oportunidad de realizar hasta el momento, realizando un estudio de las tecnologías similares a Javascript que existen hasta el momento, y cual de ellas se adaptaba mejor a nuestro objetivo en particular. Además, la posibilidad de estudiar en profundidad los distintos tipo de redes de interconexión multietapa, y las características propias que poseen cada una.