

UNA NUEVA FAMILIA DE TOPOLOGÍAS INDIRECTAS, EFICIENTES Y TOLERANTES A FALLOS

-RESUMEN-

Actualmente, los grandes sistemas de cómputo paralelo están adoptando la arquitectura de *cluster* como base de su construcción (lista Top500). Estos *clusters* están siendo contruidos con miles de nodos de procesamiento, los cuales se conectan a través de una red de interconexión de altas prestaciones.

En estos sistemas, el rendimiento, la tolerancia a fallos y el coste de la red juegan un factor clave en el diseño de todo el sistema. Los niveles de cómputo requeridos solo pueden ser alcanzados incrementando el número de nodos que lo componen. Sin embargo, a medida que el sistema crece también lo hace la cantidad de componentes de la red, y con ello la probabilidad de un fallo en la misma. Dado que la disponibilidad de estos sistemas es una preocupación, los mecanismos de tolerancia a fallos son implementados regularmente basados en el aumento y replicación de componentes, afectando de forma directa a su coste.

En este campo, las topologías indirectas a menudo son elegidas en el diseño de *clusters* de alto rendimiento. Entre ellas, la más utilizada es el fat-tree, la cual es una topología bidireccional multietapa que provee un buen rendimiento de red y un buen nivel de tolerancia a fallos, pero a un alto coste. Para reducir su coste, se propuso RUFT, una topología unidireccional multietapa que obtiene un rendimiento de red similar al fat-tree, utilizando menos recursos de hardware (aproximadamente la mitad). Sin embargo, el punto débil de RUFT es que no ofrece ningún tipo de tolerancia a fallos.

En este trabajo, nos enfocamos en diseñar una topología indirecta que ofrezca un alto rendimiento de red y sea tolerante a fallos, a la vez que mantiene un bajo coste del hardware.

En particular, proponemos una nueva familia de topologías indirectas con diferentes propiedades en términos de coste, rendimiento y tolerancia a fallos. Estas nuevas topologías son capaces de alcanzar un rendimiento similar o mejor al ofrecido por el fat-tree, además de ofrecer un buen nivel de tolerancia a fallos y, a diferencia de la mayoría de topologías disponibles, también son capaces de tolerar fallos en los enlaces que conectan con los nodos de procesamiento.

Nuestra primera contribución es RUFT-PL, una topología que duplica los enlaces de inyección, red y eyección, siguiendo el mismo patrón de conexión utilizado por RUFT para interconectar todos los elementos de la red. Esta topología obtiene un alto rendimiento de red y un ligero grado de tolerancia a fallos, usando los mismos recursos de hardware que el fat-tree.

Como segunda contribución, proponemos la topología FT-RUFT-212. Esta topología incrementa el rendimiento de red con respecto al fat-tree, ofreciendo además un buen nivel de tolerancia a fallos a un bajo coste de diseño, gracias al sistema de inyección/eyección propuesto que implementan los nodos de procesamiento.

La tercera contribución, FT-RUFT-222, es una topología que aprovecha las mejores propiedades de las dos propuestas anteriores. En particular, esta topología implementa la inyección/eyección utilizada por FT-RUFT-212 y los dobles enlaces de red de RUFT-PL para conectar los conmutadores. Esta propuesta ofrece un alto rendimiento de red y de tolerancia a fallos, utilizando los mismos recursos de hardware requeridos por el fat-tree.

Nuestra última contribución es FT-RUFT-XL, una topología que rediseña tanto la inyección/eyección como la conexión entre los conmutadores. Esta topología incrementa notablemente el nivel de tolerancia a fallos ofrecido por las demás propuestas, ofreciendo también un alto rendimiento de red. Además, a diferencia de muchas topologías unidireccionales, ésta permite que los paquetes tomen diferentes rutas en cada etapa de la red, acercándolos siempre a su destino en cada salto.