Réseaux de serveurs et de stockage, offrant basse consommation et faible latence, sur base RapidIO

Mohammad Akhter, Architecte en chef Integrated Device Technology

Récemment, les serveurs lame, les micro-serveurs et les micros-systèmes de stockage suscitent un intérêt croissant en tant qu'alternatives aux systèmes en rack traditionnels, pour les centres de calcul de type "cloud" (architecture à serveurs virtuels délocalisés). Les serveurs et systèmes de stockage d'une infrastructure "cloud" typique doivent permettre une évolution facile, et une virtualisation efficace, tandis que les processeurs multi-cœurs performants à faible latence qui les équipent doivent s'interconnecter pour constituer un système fiable et consommant le moins possible, tout en garantissant une bonne réactivité (faible latence) à l'utilisateur.

Les architectures à lames, équipées de cartes de commutation en châssis et d'un minimum de câbles de liaison, offrent un coût inférieur et une fiabilité accrue par rapport aux systèmes en rack. D'un autre côté, les micro-serveurs et microsystèmes de stockage, offrent un encombrement et une consommation inférieurs par rapport aux lames. En fonction de l'architecture et de la disponibilité d'E/S (Entrée/Sortie) natives sur le processeur, la latence des micro-serveurs et microsystèmes de stockage peut être sensiblement réduite tout en offrant une fiabilité et une évolutivité supérieures, et une virtualisation sécurisée.

Typiquement, les architectures serveur ou de stockage, font appel à des interconnexions de type Ethernet, PCIe ou InfiniBand. Bien qu'Ethernet (avec TCP/IP) soit majoritairement utilisé pour les réseaux, il peut aussi servir à transporter du trafic mixte (stockage, gestion réseau, et calcul) en utilisant un Ethernet dit "convergé" (Converged Enhanced Ethernet, ou Ethernet convergent amélioré). Néanmoins, la latence et la consommation de tels systèmes à trafic mixte, sont assez élevées du fait de piles de protocole gourmandes en calcul. Le QoS (qualité de service) est également impacté du fait des méthodes de gestion de congestion par trames, et de l'incapacité d'Ethernet à interrompre la transmission de trames longues.

D'un autre côté, les protocoles PCIe et InfiniBand, sont surtout utilisés pour le trafic calcul et stockage. Privé d'une gestion native de messages et ne pouvant gérer qu'une seule arborescence, PCIe est limité en évolutivité. Les systèmes serveur et de stockage basés sur des dispositifs InfiniBand quant à eux, manquent d'un support natif de processeur, et souffrent d'un coût supérieur et d'une latence de bout en bout plus élevée.

1. Exemple d'utilisation "Cloud"

Parmi d'autres, les architectures lame et micro-serveur peuvent être intéressantes pour des applications comme l'analyse de grands volumes de données, avec des contraintes de temps, en mode "cloud". Une telle application doit pouvoir supporter des flux de trafic mixtes correspondant au réseau, au stockage et au calcul. Pour assurer un meilleur QoS, ce genre d'applications doit également être capable de traiter un grand volume de données ou de lire un flux de données temps-réel en un temps limité. L'environnement Apache Hadoop plus HBase est typiquement utilisé pour l'analyse de gros volumes de données.

L'environnement Hadoop-HBase comprend trois composants : MapReduce (calcul), HBase (mémoire), et un tissu d'interconnexion. Dans ce contexte, les données et les tâches associées au moteur MapReduce sont chargées et exécutées en parallèle, sur une batterie de serveurs connectés à un réseau extensible et interconnecté de manière performante, en fonction du volume de données. Les fonctions Map et Reduce de l'environnement assurent les calculs, tandis que HBase assure un accès rapide et aléatoire aux grands noeuds de stockage distribués sur le réseau. Pour qu'un grand nombre de moteurs de mapping fonctionnent en parallèle, le grand volume de données est d'abord partitionné en plusieurs sous-ensembles plus petits. Les moteurs de mapping trient ensuite les données selon un format intermédiaire (associé à une "clé") et les transfèrent à la mémoire la plus proche du noeud de calcul.

Après le chargement et le mappage des données, les données en sortie des noeuds de stockage sont transférées aux différents serveurs réducteurs ("Reduce") en fonction de la valeur de la "clé". Les serveurs maîtres assurent la distribution des données aux serveurs réducteurs, selon la proximité des données intermédiaires. Une fois le traitement réducteur terminé, les données sont recombinées pour produire les résultats finaux qui sont enregistrés dans un ou plusieurs fichiers de sortie dans le serveur de stockage, via le réseau.

Dans un tel environnement, pour achever le traitement (calcul et accès au stockage) en un temps donné, l'infrastructure logicielle peut profiter d'architectures serveur et de stockage, associées à des interconnexions efficaces. Par exemple, le modèle MapReduce peut tirer profit d'une infrastructure à faible latence pour charger et exécuter des tâches en parallèle sur un nombre variable de noeuds de calcul et de stockage. Avec un matériel performant à tolérance de panne, correction d'erreur et synchronisation à faible latence, un tel environnement permet aussi de simplifier le logiciel assurant la gestion et la fiabilité du système (MapReduce plus HBase). En résumé, le traitement de données dans un tel environnement impose les contraintes suivantes sur les interconnexions du centre de calcul.

Etapes de traitement	Contraintes d'interconnexion
Partition des tâches, Distribution, et suivi des données	Interconnexion fiable, extensible et réactive pour contrôler, suivre et gérer les tâches et les données avec le contrôle de flux approprié
Chargement des données	Transfert de données à faible latence, prévisible et fiable entre les nœuds de stockage et de traitement
Réplication des données	Interconnexion fiable avec multicast pour supporter la réplication de données initiale et périodique
Traitement "Mapping"	Redistribution des données à faible latence vers les serveurs réducteurs, selon la "clé"
Traitement "Reduce"	Transfert de données à faible latence pour produire les résultats finaux et les écrire en mémoire
Tolérance de panne et fiabilité	Réseau à tolérance de panne pour simplifier le logiciel assurant la fiabilité
Trafic mixte ("Converged")	Maintenir un excellent QoS pour le trafic de stockage, de calcul et de gestion réseau.

Le protocole RapidIO, abordé dans la section suivante, répond à toutes ces contraintes ainsi qu'à beaucoup d'autres relatives aux applications serveur et de stockage en mode "cloud".

2. Présentation de RapidIO

Le protocole haute performance RapidIO a été introduit en 2002, en tant que norme ouverte de transmission de données par paquets. Depuis lors, des millions de dispositifs RapidIO ont été livrés dans le monde par un grand nombre d'OEM et de fournisseurs de silicium pour répondre aux exigences d'interconnexion de stations de base radio 3G/4G, de serveurs vidéo, ou d'applications de transmission militaires, de traitement embarqué ou de calcul haute performance. Certaines de ces applications ont tiré profit de l'architecture extensible à faible latence de RapidIO, tandis que d'autres ont bénéficié de sa garantie déterministe de livraison, de sa tolérance aux pannes et de sa fiabilité, pour connecter un grand nombre de CPU multi-coeurs.

Le protocole et le format des paquets RapidIO sont spécifiés dans une structure hiérarchique à trois couches. Le protocole supporte les liaisons courtes, moyennes et longues, éventuellement à travers plusieurs cartes. La norme supporte aussi bien les liaisons de type fibre que câble.

En termes de gestion et de contrôle, le protocole RapidIO supporte tous les mécanismes de construction système, d'interopérabilité, de multidiffusion, de gestion d'erreur et de récupération de panne. Grâce à l'architecture simplifiée des couches, il est possible d'implanter les piles de protocole au niveau matériel, en limitant la complexité du logiciel, le coût global du système et la consommation. Le Tableau 2 résume les principales caractéristiques de RapidIO applicables aux architectures lame et micro-serveur.

Tableau 1 Principales caractéristiques du protocole RapidIO applicables aux serveurs lame et aux micro-serveurs

Attributs	Caractéristiques du protocole RapidIO
Latence	➔ Plus faible latence garantie d'application à application avec livraison déterministe des paquets (latence de commutation 100 ns environ). ➔ Des symboles de contrôle assurent un contrôle et une synchronisation à très faible latence
Coût et consommation	➔ Commutateurs et NIC (cartes réseau) les moins chers (~2W par NIC, ~10W par commutateur)
"Scalabilité" (extensibilité)	➔ Facilement extensible à plusieurs milliers de nœuds et plusieurs millions de flux
Qualité de service (QoS)	➔ Excellent QoS grâce aux en-têtes efficaces, aux formats des paquets, au contrôle de flux, et à la gestion de congestion : plus faible latence, symboles de contrôle courts, VOQ-BP et gestion de trafic de données
Tolérance de panne **Fiabilité**	➔ Détection rapide et isolement des pannes ➔ ➔ Livraison ordonnée et faible grâce au matériel ; peu de logiciel supplémentaire.
Débit et topologie	Bande passante E/S adaptée aux performances du processeur grâce à une meilleure utilisation du tissu d'interconnexion et à la vitesse du protocole ➔ x1, x2, x4, x8, x16 voies ➔ ➔ Jusqu'à 100 GBaud par port (Gen2). ➔ ➔ Plus de 160 GBaud par port (Gen3 10xN), voire 400 GBaud par port (25xN roadmap). ➔ ➔ Toutes topologies : tore, maille, arbre, hypercube, etc…
Trafic mixte	Support des trafics calcul, réseau et stockage avec un excellent QoS, trafic réseau et stockage avec QoS élevé grâce à un format de paquet efficace et à un contrôle de flux

Attributs	Caractéristiques du protocole RapidIO
	symbolique compact. Supporte VCs.
Réplication des données	➔ Supporte unicast, multicast et broadcast grâce au tissu de commutation RapidIO et aux fonctions multicast de RapidIO
Sécurité	➔ Paires de files d'attente sécurisées, sécurité au niveau port par cryptage, et sécurité au niveau protocole par filtrage
Virtualisation	➔ Piles de protocole simplifiée implantées en matériel, simplifiant la virtualisation E/S en permettant la partage direct des ressources E/S de l'application. Réseau à faible latence et haut débit pour une migration de VM plus facile

3. RapidIO pour serveurs lame et micro serveurs

Pour tirer le meilleur profit des performances de RapidIO, il est possible de concevoir des lames ou des micro-serveurs x86 à l'aide de cartes réseau et de commutateurs conformes RapidIO (Figure 1). On peut aussi construire des micro-serveurs (Figure 2) sur base RapidIO avec des processeurs ARM ou d'autres processeurs basse consommation3. Grâce à RapidIO, il est possible d'offrir la plus faible latence système entre applications au sein de batteries importantes de serveurs et de systèmes de stockage. Avec un système d'interconnexion à haut-débit et faible latence, les applications peuvent être exécutées et partitionnées sur les différents noeuds de calcul et de mémoire, de manière transparente. En outre, avec une tolérance de panne fiable, il est possible d'améliorer la disponibilité système et de réduire la charge CPU liée à la synchronisation, à la commande et à la surveillance du système.

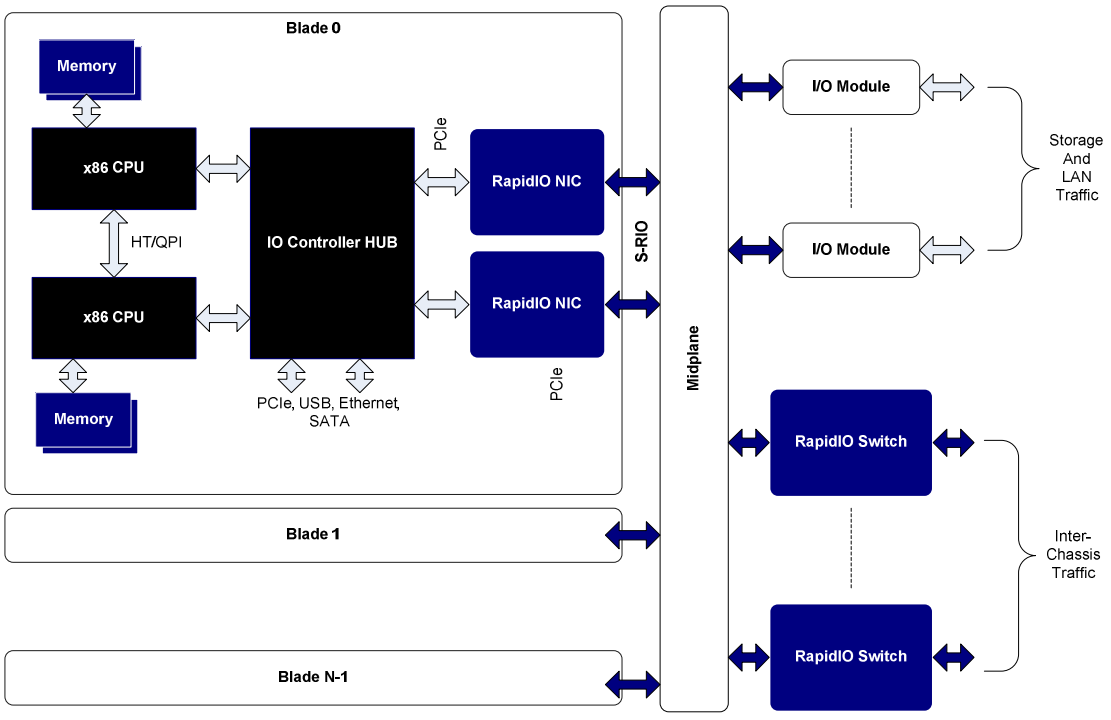


Figure 1 Les architectures serveur et de stockage à lames x86 et dispositifs RapidIO offrent un coût inférieur et une meilleure fiabilité que les systèmes en rack

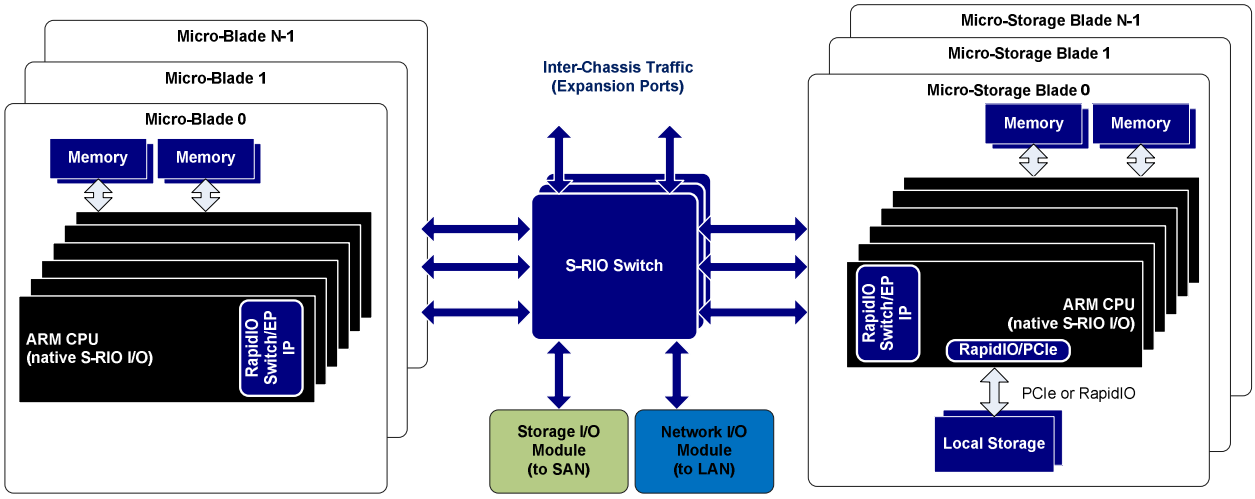


Figure 2 Les architectures micro-serveurs et de stockage à processeur ARM ou autre processeur basse consommation et RapidIO natif, offrent une latence inférieure à celle des systèmes en rack

Résumé

Les dispositifs RapidIO peuvent être utilisés dans les centres de calcul pour différentes applications comme l'analyse à latence critique de grands volumes de données, la recherche web, l'analyse, le jeu en ligne multi-joueurs, et la mise en cache distribuée. Les systèmes basés sur le protocole RapidIO à faible latence peuvent par nature être étendus à un grand nombre de noeuds et de files d'attente tout en maintenant un excellent QoS, une bonne tolérance aux pannes et une très bonne fiabilité. Les architectures RapidIO garantissent une livraison déterministe en bon ordre et une utilisation optimum du tissu d'interconnexion en adaptant les E/S aux performances du processeur grâce à un contrôle de flux performant et au format des paquets. En résumé, RapidIO constitue une excellente solution évolutive pour les réseaux de serveurs et de stockage compacts à faible latence de nouvelle génération.