A LIGHT-WEIGHT VIRTUAL MACHINE MONITOR FOR BLUE GENE/P

JAN STOESS, UDO STEINBERG VOLKMAR UHLIG, JOHNATHAN APPAVOO, AMOS WATERLAND, JENS KEHNE

CARLOS MARTÍN FLORES GONZÁLEZ

mfloresg@computer.org

ESCUELA DE INGENIERÍA EN COMPUTACIÓN
INSTITUTO TECNOLÓGICO DE COSTA RICA

SISTEMAS OPERATIVOS AVANZADOS
PROFESOR: FRANCISCO TORRES, PH.D
20 DE MARZO, 2017

BLUE GENE/P EN EL ARGONNE NATIONAL LABORATORY. TOMADO DE WIKIPEDIA.

AGENDA

- Problema
- L4 + VMM
- El sistema
- Microkernel
- VMM
- Resultados iniciales
- Conclusiones

Blue Gene/P en Brookhaven National Laboratory. Wikipedia

PROBLEMA

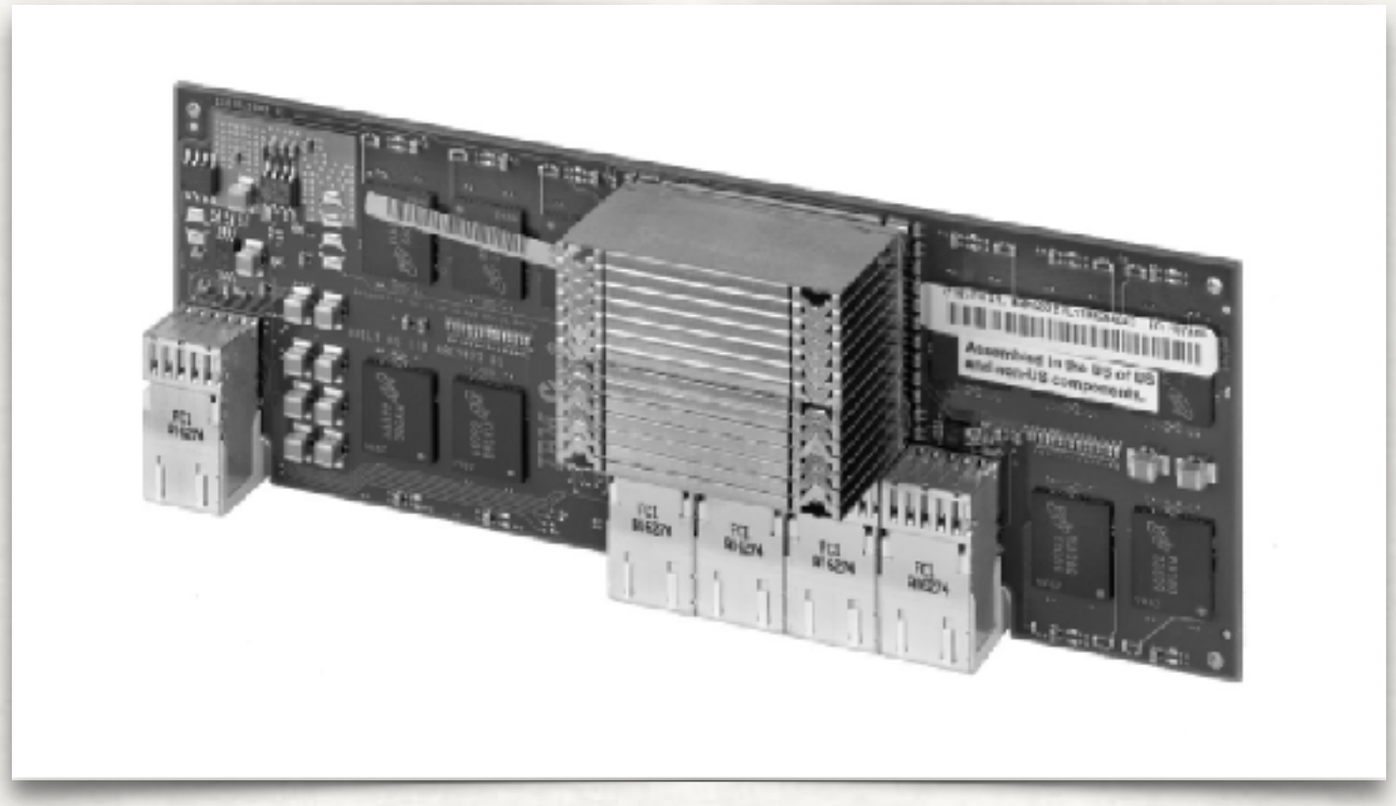
- IBM provee un kernel light para la Blue Gene/P llamado Compute Node Kernel (CNK)
- CNK facilita el desarrollo de aplicaciones para supercomputadoras (POSIX)
- CNK no es totalmente compatible con POSIX
- Las aplicaciones de hoy en día empiezan a escalar a sistemas Exascale de dimensiones globales.
- El soporte restringido de interfaces estandarizadas hace que esto se convierta en un cuello de botella

UN MONITOR DE MÁQUINA VIRTUAL BASADO EN MICRO-KERNEL

- Un micro-kernel provee un pequeño conjunto de primitivas de sistema operativo
- Se construye un VMM de nivel de usuario que virtualiza la plataforma BG/P y permite que sistemas operativos(SO) arbitrarios se ejecuten.
- Provee compatibilidad con el hardware de la BG/P
- Reducen la funcionalidad del kernel a gestión básica de recursos y comunicación
- Prototipo basado en el microkernel L4

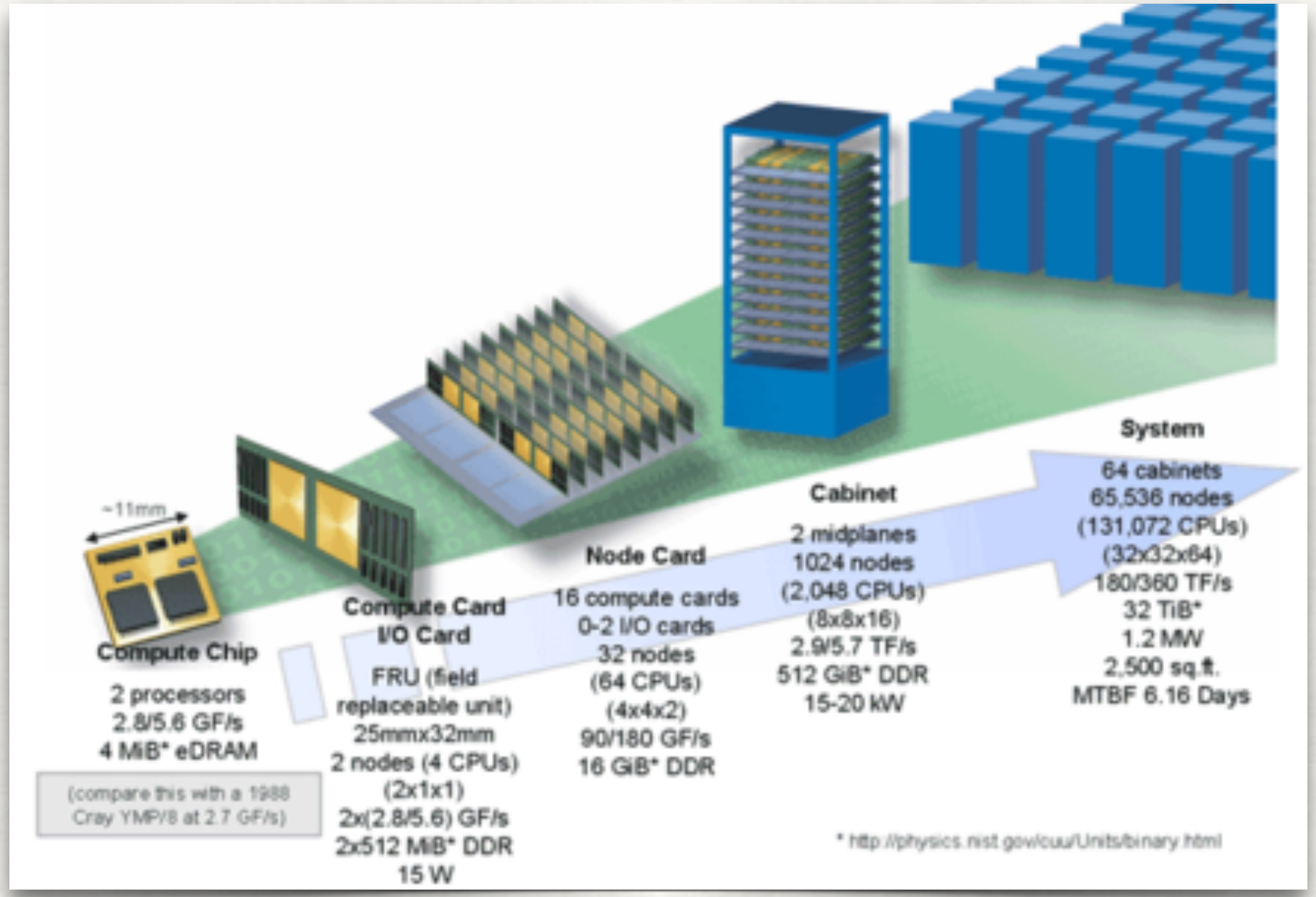
EL SISTEMA

- El bloque básico de un BG/P es un computer node
- PowerPC quadcore, cinco interfaces de red, un controlador DDR2 y 2 o 4GB de RAM integrado todo en un chip

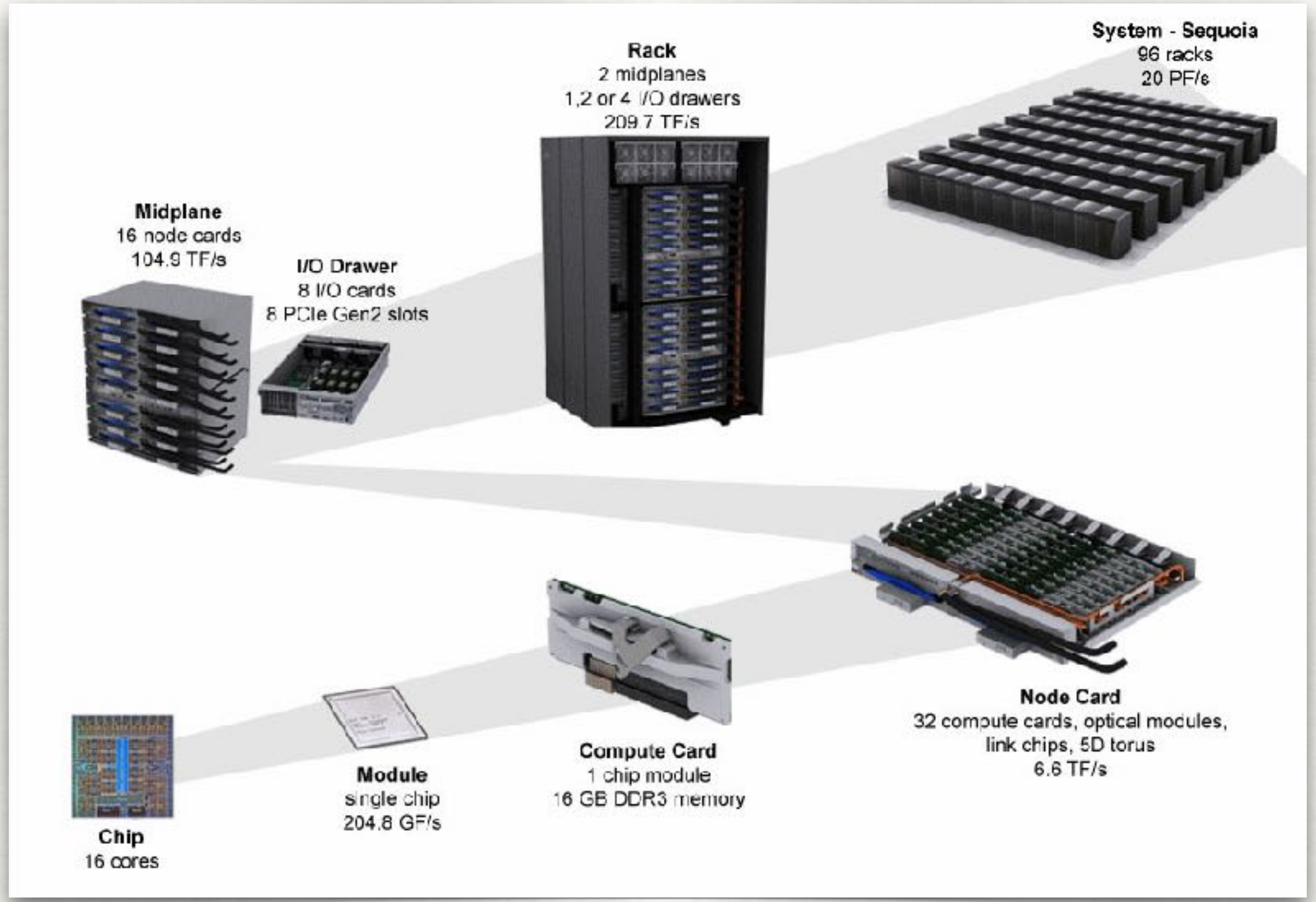


BG/P Compute Node Card. Stoess et al.

EL SISTEMA - BG/P HARDWARE



EL SISTEMA - BG/Q HARDWARE



Fuente: https://computing.llnl.gov/tutorials/bgq/

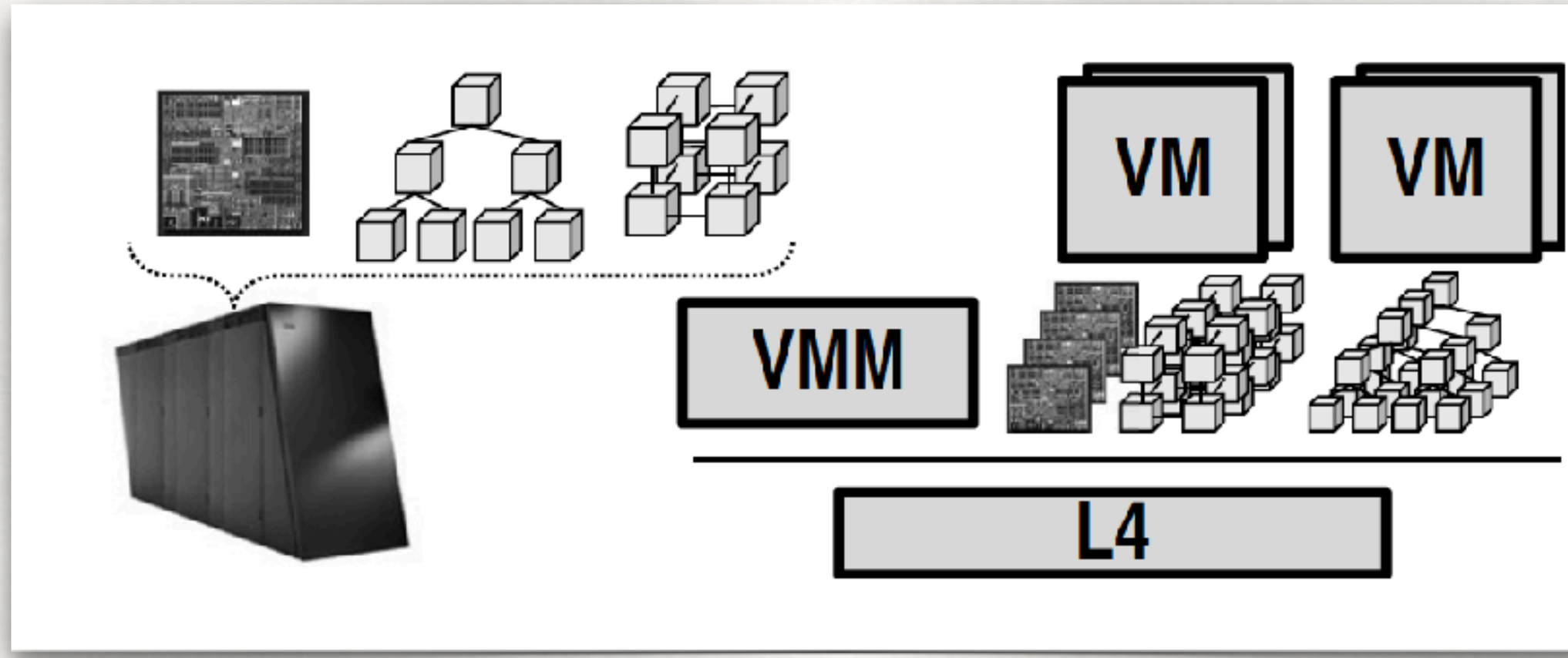
BLUE GENE/P EN EL ARGONNE NATIONAL LABORATORY. TOMADO DE WIKIPEDIA.

EL SISTEMA (3)

L4 + VMM

- Usa L4Ka::Pistachio para forzar seguridad en la ejecución
- VMM real está implementada como una aplicación de nivel de usuario por fuera del kernel
- L4 actúa como un sistema de mensajería segura propagando instrucciones sensitivas del guest al VMM. VMM a su vez, decodifica la instrucción y la emula apropiadamente y luego responde con un fault reply message que le dice a L4 que actualice el contexto de la VM guest y luego retome la ejecución.

EL SISTEMA - ARQUITECTURA BÁSICA



Un VMM basado en micro-kernel virtualizando
núcleos e interconexiones. Stoess et al.

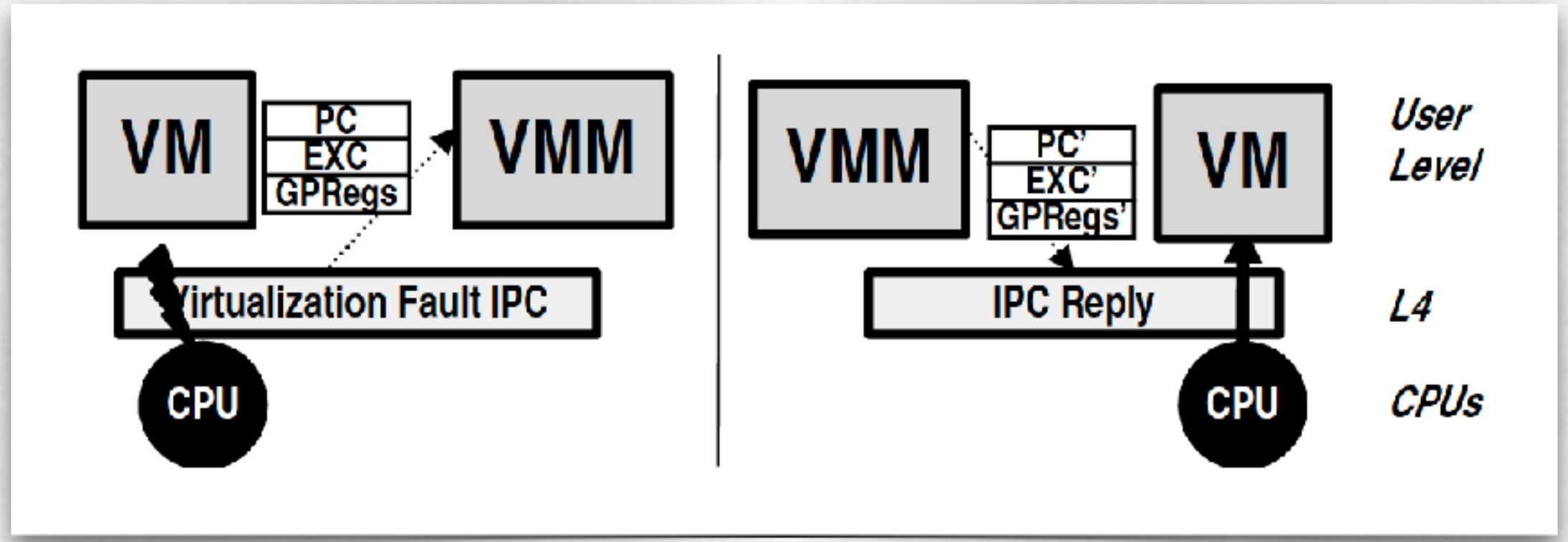
L4 MICROKERNEL

VIRTUAL POWERPC PROCESSOR

- L4 virtualiza núcleos(cores) mapeando cada CPU (vCPU) virtual en un thread dedicado
- L4 como tal no emula todas las instrucciones sensitivas por si mismo. A menos que la instrucción esté relacionada con TLB virtual o que pueda ser manejada rápidamente
- Se depende en los IPCs de L4 para implementar el protocolo de virtualización: **guest-trap → emulación de VMM → reinicio del guest**
- L4 sintetiza un mensaje de IPC fault en nombre del guest a un manejador de excepciones de vCPU designado

L4 MICROKERNEL (2)

VIRTUAL POWERPC PROCESSOR



Las “salidas” del vCPU son propagadas al VMM como mensajes IPC, el VMM responde enviando un reply IPC para reanudar al guest. Stoess et al.

Volver

L4 MICROKERNEL (3): VIRTUALIZED MEMORY MANAGEMENT

GUEST-VIRTUAL → GUEST-PHYSICAL | GUEST PHYSICAL → HOST-PHYSICAL

- **Virtual Physical Memory:** L4 trata el espacio de direcciones físicas de la VM guest como un espacio normal de direcciones y delega el establecimiento de traducciones a un paginador de nivel de usuario (el VMM).

- **Virtual TLB:** L4 provee la noción de TLB virtual. Gestión: por medio de instrucciones internas de L4. Instrucciones de emulación internas de L4 guardan las entradas dentro de un vTLB por VM.

- **Virtual Address Space Protection:** Un VMM virtualiza el motor de traducción del TLB y sus características de protección.

MICROKERNEL (4)

INTERRUPT VIRTUALIZATION

- GB/P provee un controlador personalizado de interrupciones llamado Blue Gene Interrupt Controller(BIC)

- El VMM usa el soporte en interrupciones que provee L4 para recibir y reconocer interrupciones para los dispositivos BG/P. Para inyectar interrupciones virtuales en el guest, el VMM modifica el estado de vCPU ya sea por medio de :

  - un llamado de sistema de cambio de estado de L4

  - llevando la actualización del estado a la virtualization fault reply, en el caso de que el VM guest ya estuviera esperando por el VMM cuando una interrupción ocurre. **Link**.

USER-LEVEL VMM

TRADUCE INVOCACIONES DEL API DE VIRTUALIZACIÓN EN INVOCACIONES DE API DE LA ARQUITECTURA DE L4.

- **Emulating Sensitive Instructions**: con la llegada de un virtualization fault IPC el VMM decodifica la instrucción y sus parámetros los cuales están almacenados dentro del IPC que fue enviado desde L4 en nombre de la VM que lanzó la trampa.

- **Virtual Physical Memory:**

- Cuando el guest sufre un TLB miss:

  - L4 envía un page fault IPC conteniendo la instrucción que falla, la dirección y otros estados de TLB (virtuales) necesarios

  - El VMM responde con un mensaje IPC que hace que L4 inserte el mapeo correspondiente es su base de datos de vTLB y en el hardware.

USER-LEVEL VMM

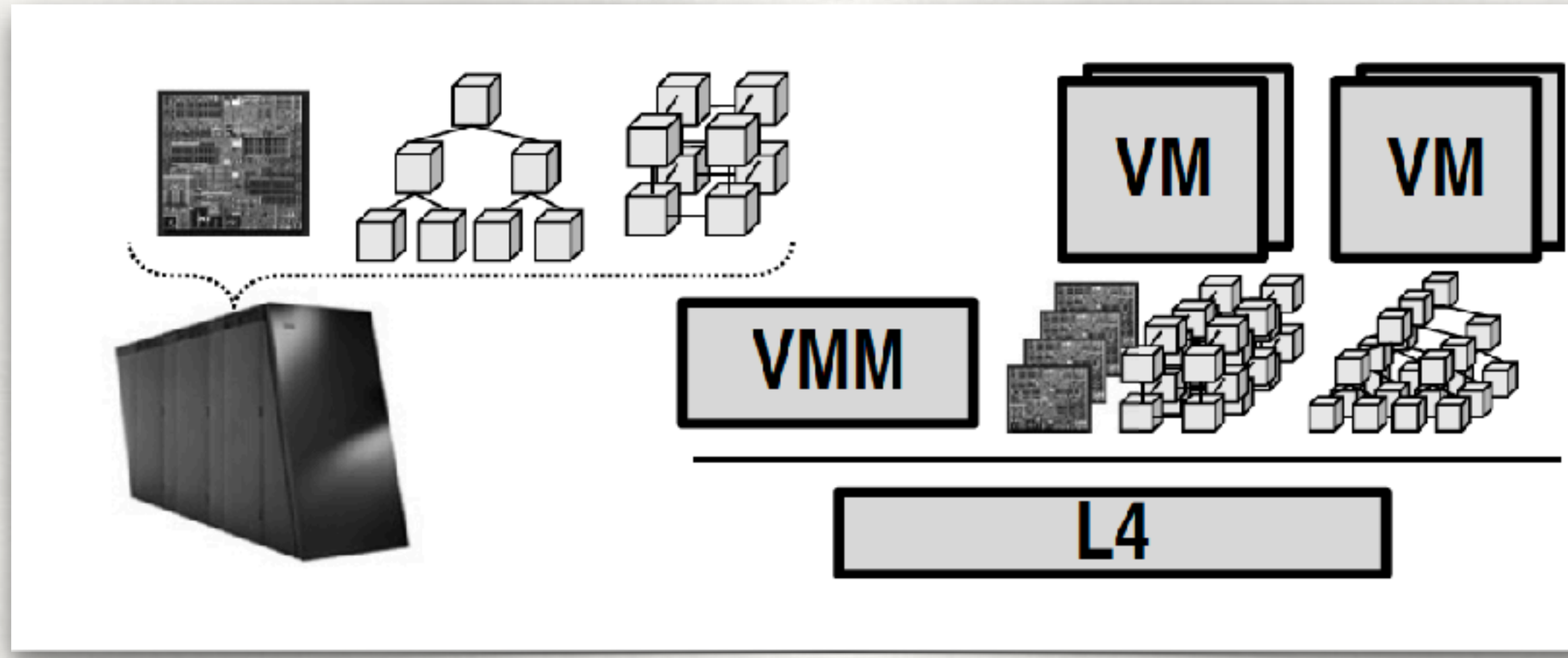
DEVICE VIRTUALIZATION

- **Collective Network:** es un árbol binario sobreconectado. Un medio uno-a-todos para operaciones de transmisión o reducción

- **Torus:** Cada compute node es parte de una red toroidal 3D. Provee dos interfaces de transmisión, una normal basada en buffer y otra basada en accesos de memoria directo remoto

- El VMM emula los registros de control (DCR) usado para configurar el dispositivos. Las instrucciones correspondientes son sensitivas y directamente atrapadas por L4 y el VMM.

EL SISTEMA - ARQUITECTURA BÁSICA



Un VMM basado en micro-kernel virtualizando
núcleos e interconexiones. Stoess et al.

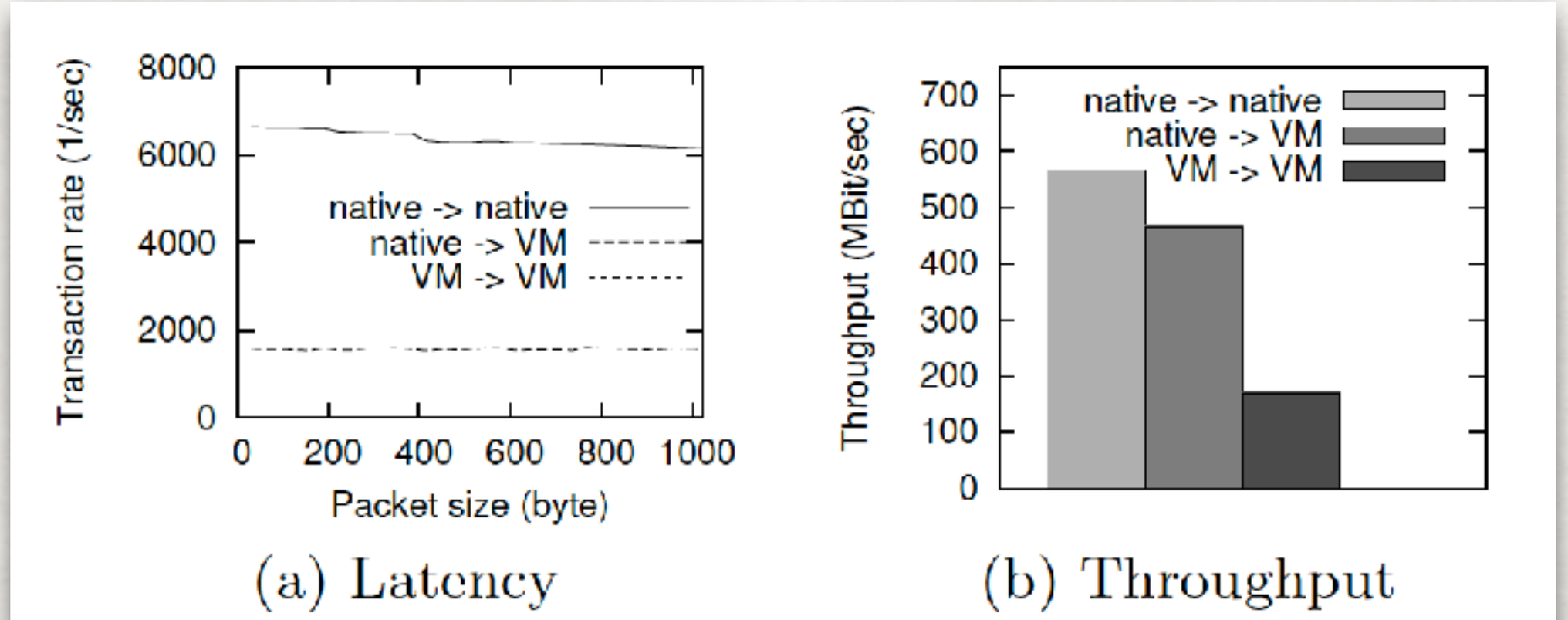
RESULTADOS INICIALES

- El VMM basado en L4 soporta la ejecución de SO guests en GB/P
- Permite que uno o más instancias de Kittyhawk Linux corran como VM, sin modificación alguna
- **Primer experimento**: compilación de un proyecto de código pequeño en la versión virtualizada de Kittyhawk Linux + Herramienta de debug de L4 para encontrar rutas de código de VM frecuentemente ejecutadas.
- Se notó que el número de IPCs es relativamente bajo, lo que significa que L4 maneja la mayoría de las salidas del guest internamente.
- Alto número de intentos fallidos de TLB y de instrucciones relacionadas con TLB, lo que indica que el subsistema de memoria virtualizada es un cuello de botella en la implementación.

RESULTADOS INICIALES (2)

- En el segundo experimento, se midió el throughput y la latencia entre dos compute nodes. Paquetes fueron enviados al torus interconnect por medio del modulo controlador de Ethernet de Kittyhawk Linux.

- En los resultados se mostró que la capa de virtualización plantea un overhead significativo al rendimiento de la red Ethernet.



CONCLUSIONES

- El CNK de la BG/P es un kernel no totalmente compatible con POSIX. Se compromete la portabilidad de las aplicaciones

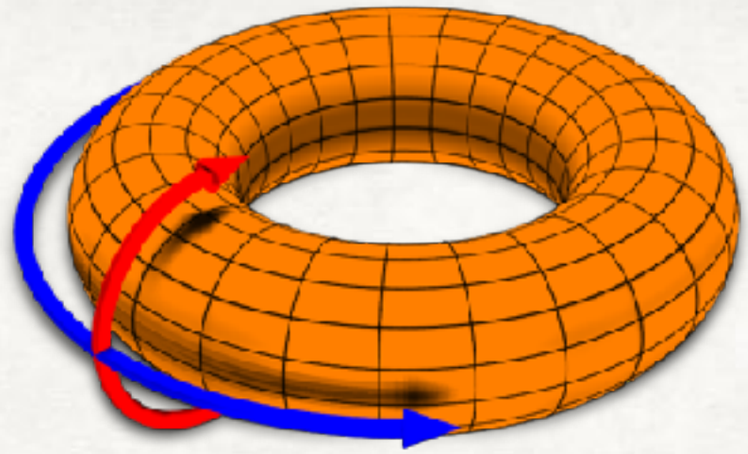
- Un sistema operativo híbrido se propone e implementa para brindar compatibilidad con el hardware de BG/P:
  L4: el microkernel
  VMM de nivel de usuario: capa que virtualiza BG/P

- En resultados iniciales se pudo correr SOs sin modificaciones sobre el VMM

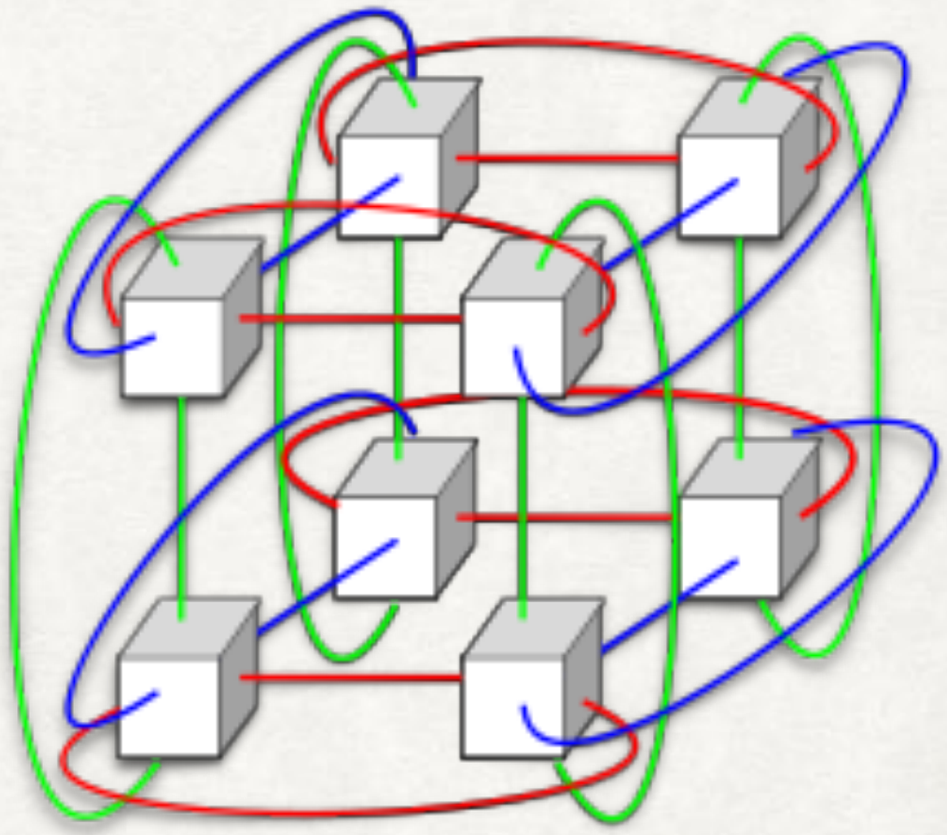
- Función por sobre rendimiento: La solución presenta overhead en TLB misses y en el rendimiento de la red Ethernet.

MUCHAS GRACIAS

TOROIDE: “superficie de revolución generada por una circunferencia que gira alrededor de una recta exterior coplanaria (en su plano y que no la corta) o, llanamente, la curva tridimensional que resulta de hacer girar una circunferencia alrededor de un eje que no la corta. ”.



TORUS Interconnect: “A torus interconnect is a switch-less network topology for connecting processing nodes in a parallel computer system”.



Volver