Nuevo esquema del sistema de colas de Miztli

Coordinación de Supercómputo
Dirección General de Tecnologías de Información y Comunicación
UNAM

Índice

1. Introducción

El nuevo esquema del sistema de colas cuenta con 6 colas, que se encargan de manejar los diferentes tipos de recursos de Miztli:

q_htc, q_hpc, q_residual
Manejan los nodos regulares. Son tres conjuntos de nodos, para un total de 372 nodos y 6184 cores.

q_gpu
Maneja los nodos con GPUs. Son 8 nodos con 2 GPUs Tesla M2090 cada uno.

q_32p_256g, q_16p_256g
Manejan 10 nodos con 256 Gigabytes de RAM.

Al haber pocas colas el esquema puede parecer simple, pero en realidad necesita más intervención del usuario en la definición de parámetros del job, más aún considerando que las colas q_htc, q_hpc y q_residual manejan nodos que tienen características diferentes en cuanto a tipo y número de

cores y a cantidad de memoria. Así pues, para hacer un mejor uso de sus recursos, es importante que no le queden dudas acerca de:

1. Para qué sirve cada una de las colas.
2. Cuáles son las colas que puede usar y cómo usarlas.
3. Cuáles son los límites de cada cola.
4. Cuándo es conveniente usar q_residual.
5. Por qué es importante consultar el uso de los recursos y el estado de la máquina, y cómo hacerlo.
6. Cómo enviar jobs de diferentes tipos (memoria compartida, MPI, etc). a los diferentes tipos de nodos regulares.

2. Descripción de las colas

Anteriormente, las colas para usar los nodos regulares se definían a partir de las características de los *jobs* (duración, número de procesadores). En el nuevo esquema, las colas se definen a partir de las características de los *proyectos de supercómputo* (recursos asignados, ritmo de utilización). Los proyectos se han dividido en dos grupos: HTC y HPC, a los que corresponden las colas q_htc y q_hpc, respectivamente.

2.1. q_htc

Los proyectos HTC son los que requieren correr jobs de forma continua, y tienen recursos suficientes para hacerlo. Los miembros de estos proyectos deben correr sus jobs usando la cola q_htc, cuyas características son:

- Solamente los usuarios con proyectos HTC pueden enviar trabajos a esta cola.
- Cada proyecto tiene un límite de cores que puede usar simultáneamente. Este límite está calculado de acuerdo a la cantidad de recursos que tiene asignado el proyecto.
- Se pueden enviar jobs con cualquier número de cores, hasta el límite del proyecto.
- Se puede enviar cualquier cantidad de jobs.
- La duración máxima de los jobs es de 240 horas.
- q_htc tiene la mayor prioridad, así que los jobs en espera y dentro del límite de esta cola son los primeros en ser despachados cuando se liberan cores.
- Los tiempos de espera de jobs dentro del límite de la cola son mínimos.

2.2. q_hpc

Los proyectos HPC son los que no requieren ejecutar jobs de forma continua o no tienen suficientes recursos asignados para hacerlo. Los miembros de estos proyectos deben correr sus jobs usando la cola q_hpc, cuyas características son:

- Solamente los usuarios con proyectos HPC pueden enviar trabajos a esta cola.

- Cada proyecto tiene un límite de cores que puede usar simultáneamente. Este límite se ha establecido de acuerdo al tipo de software que preponderántemente usan los miembros del proyecto.
- Se pueden enviar jobs con cualquier número de cores, hasta el límite del proyecto.
- Se puede enviar cualquier cantidad de jobs.
- La duración máxima de los jobs es de 72 horas.
- Esta cola es la segunda con mayor prioridad. Todos sus jobs en espera son despachados después de los jobs de q_htc y antes que todas las demás colas.
- Los trabajos en esta cola compiten por los recursos en el esquema "fairshare", es decir, existe una prioridad dinámica que disminuye conforme se usan más recursos.
- El tiempo de espera de los jobs dependerá del estado de la máquina, pero hay que recalcar que Miztli tiene recursos suficientes para ejecutar simultáneamente jobs de al menos 10 proyectos HPC.

2.3. q_residual

En periodos en los que Miztli no está saturada, es posible utilizar más cores de los que establecen los límites de las colas q_htc y q_hpc mediante el uso de q_residual, cuyas características son:

- Cualquier usuario puede usar esta cola.
- No tiene límites en el número de cores.
- La duración máxima de los jobs es de 72 horas.
- Es la cola con menor prioridad, de modo que puede despachar sus jobs solamente si no hay jobs por despachar en q_htc o q_hpc.
- Si todos los cores de los nodos regulares están ocupados, y cualquiera de las colas q_htc o q_hpc recibe un job dentro del límite, se eliminarán tantos jobs de q_residual como sea necesario para ejecutar el job recién recibido.

2.4. Nodos con GPUs

La cola q_gpu es la que se debe utilizar para tener acceso a los nodos con GPUs. Las características de esta cola son:

- Cualquier usuario puede utilizarla.
- Se pueden usar hasta 2 nodos (32 cores y 4 GPUs) simultáneamente.
- La duración máxima de los jobs es de 96 horas.
- Los jobs deben solicitar al menos 16 cores, para garantizar que no haya interferencias en el uso de las GPUs.

2.5. Nodos con 256 GB de RAM

Las colas q_32p_256g y q_16p_256g son las que deben utilizarse para tener acceso a los nodos con 256 Gigabytes de RAM. Las características de estas colas son:

- Cualquier usuario puede utilizarlas.
- El límite de cores por job es de 16 (16p) o 32 (32p).
- El límite de cores por grupo en la cola es de 32 en ambos casos.
- La duración máxima de los jobs es de 124 horas (16p) o de 72 horas (32p).
- Los jobs deben solicitar al menos 16 cores, para garantizar que no haya interferencias en el uso de la memoria.

3. Consideraciones adicionales para el uso de nodos regulares

Un usuario puede mandar tantos jobs como desee a la cola que le corresponde a su proyecto. El sistema de colas ejecutará los jobs hasta alcanzar el límite del proyecto, y los demás quedarán en espera. Es importante notar que los límites son por grupo, no por usuario.

Si se desea usar simultáneamente más cores de los que permiten los límites, se pueden enviar los jobs adicionales a la cola q_residual. Los jobs en esta cola iniciarán sólo si hay recursos disponibles.

Debe ser precavido en la selección de la cola para sus jobs. En general, deben evitarse los siguientes escenarios:

- Solicitar cores más allá de su límite en q_htc o q_hpc, si la máquina no está saturada. Esto sólo haría que sus jobs estén en espera siendo que la máquina tiene cores disponibles. Los cores adicionales que desee usar deberían solicitarse en q_residual.
- Solicitar cores en q_residual si todavía no ha alcanzado su límite en q_htc o q_hpc. Esto es porque los jobs en q_residual siempre tienen el riesgo de ser eliminados por el sistema en cuanto q_htc o q_hpc demanden cores y no los haya disponibles.

Estos escenarios son, en cierta medida, una contraindicación, pero confiamos en que con la práctica pronto encontrará criterios para decidir qué cola usar.

Cabe mencionar que no es necesario que modifique sus scripts para cambiar la cola de destino del job, pues esto se puede modificar desde la línea de comando. Por ejemplo, el script:

```
#!/bin/bash
#BSUB -q  q_htc
#BSUB -oo salida
#BSUB -eo error
#BSUB -n  64

mpirun ./a.out
```

se ejecutará con 64 procesadores en la cola q_htc con la siguiente instrucción:

```
 % bsub < script.lsf
```

pero se ejecutará en la cola q_residual con la siguiente instrucción:

```
 % bsub -q q_residual < script.lsf
```

De hecho cualquier parámetro del job se puede cambiar de esta forma. Por ejemplo, con la siguiente instrucción se ejecutará ese mismo script en la cola q_residual y con 128 cores:

```
% bsub -q q_residual -n 128 < script.lsf
```

4. Herramientas de consulta

El despacho de los trabajos depende de diversos factores como pueden ser el estado de los recursos, los límites establecidos o el uso de recursos de cada grupo de usuarios. El sistema de colas ofrece herramientas que permiten consultar estos datos para ayudarles a definir ciertos requerimientos de sus trabajos, con la finalidad de permitirles hacer un mejor uso de los recursos disponibles en determinado momento.

4.1. Grupo, límites, uso y estado

A cada proyecto aprobado por el Comité Académico de Supercómputo, le corresponde un grupo al que pertenecen el titular del proyecto y todas las cuentas adicionales que haya solicitado. El nombre asignado a cada grupo se forma tomando el login de la cuenta titular seguido de los caracteres "_g".
A continuación listamos algunas de la herramientas que pueden ser de utilidad para usar adecuadamente el sistema de colas:

1. Consultar datos referentes a límites, tipo de proyecto y las colas a las que un usuario tiene acceso:

   ```
   $ consulta
   ```

2. Como una herramienta para elegir el tipo de nodos que conviene solicitar al encolar sus trabajos, puede listar la cantidad de cores disponibles por grupo de nodos:

   ```
   $ cores
   ```

   El programa anterior, seguido del nombre de un grupo de nodos, muestra una breve descripción del grupo indicado:

   ```
   $ cores [GrupoDeNodos]
   ```

3. Listar el número total de procesadores en uso o asignados a un grupo de usuarios, para cada cola a la que tienen acceso:

   ```
   $ uso
   ```

4. Listar los trabajos encolados de un usuario o del grupo al que pertenece el usuario:

   ```
   $ bjobs [-u login]
   $ bjobs [-u grupo]
   ```

5. Adicionalmente, cuando se tienen trabajos en ejecución, es posible consultar los límites y la cantidad de recursos en uso por cada grupo, mediante:

   ```
   $ blimits
   ```

4.2. Características de las colas

La herramienta para consultar la configuración de las colas es bqueues. Mediante el uso de opciones y argumentos puede indicarse el tipo de información que se requiere, por ejemplo:

1. Mostrar un listado de las colas disponibles y el estado en el que se encuentran:

```
$ bqueues
```

2. Listar solamente las colas a las que un usuario o un grupo tiene acceso:

```
$ bqueues [-u login]
$ bqueues [-u grupo]
```

3. Al indicar el parámetro -l seguido del nombre de la cola puede consultar detalles sobre una cola en particular. Por ejemplo, el tiempo límite de ejecución, los usuarios que tienen acceso a la cola o los nodos a los que la cola tiene acceso:

```
$ bqueues [-l NombreDeCola]
```

5. Ejemplos de diferentes tipos de jobs

5.1. Descripción de los nodos y sus diferencias

La Fig. 1 muestra un esquema de la red de datos (infiniband) de Miztli, en el que se puede apreciar los diferentes tipos de nodos regulares y la forma en que están interconectados.

Los tipos de nodos regulares son:

- Nodos g1. Son los nodos regulares originales de Miztli. Sus procesadores son Intel Xeon 2670 v1. Cada nodo tiene 16 cores y 64 Gigabytes de RAM. Son 314 nodos.
- Nodos g2_a y g2_b. Son los nodos añadidos en 2016. Sus procesadores son Intel 2660 v3. Cada nodo tiene 20 cores y 128 Gigabytes de RAM. Son 28 y 30 nodos en cada grupo, respectivamente.

Los nodos g1 están conectados al Switch de 378 puertos, ya sea de forma directa o a través de un switch adicional (switch tipo I) que se conecta a través de 18 *hyperlinks*. La conectividad entre todos estos nodos es *simétrica*: el ancho de banda es el mismo entre cualquier par de nodos, y su valor nominal es de 40 Gbps.

Los nodos g2_a están conectados entre si mediante un switch (switch tipo II) y al resto de los nodos mediante 4 cables de fibra que van a un switch tipo I. Los 28 nodos que hay en el grupo tienen que repartirse el ancho de banda que proporcionan estas cuatro conexiones. De este modo, la conectividad de los nodos g2_a es *asimétrica*: entre nodos del mismo grupo su valor nominal es de 56 Gbps, pero hacia nodos de otros grupos es de 8 Gbps. Lo mismo se puede decir de los nodos g2_b.

Ejecutar un programa paralelo que utilice nodos de diferentes grupos sería ineficiente, dada la asimetría de g2_a y g2_b. Por esta razón, el sistema de colas de Miztli *está configurado para proporcionar todos los cores que se soliciten en un job solamente con nodos de un mismo grupo*. Esta es una restricción para cada job, pero el usuario no debe preocuparse por seleccionar el tipo de nodos, pues esta tarea la hace de forma automática el sistema de colas.

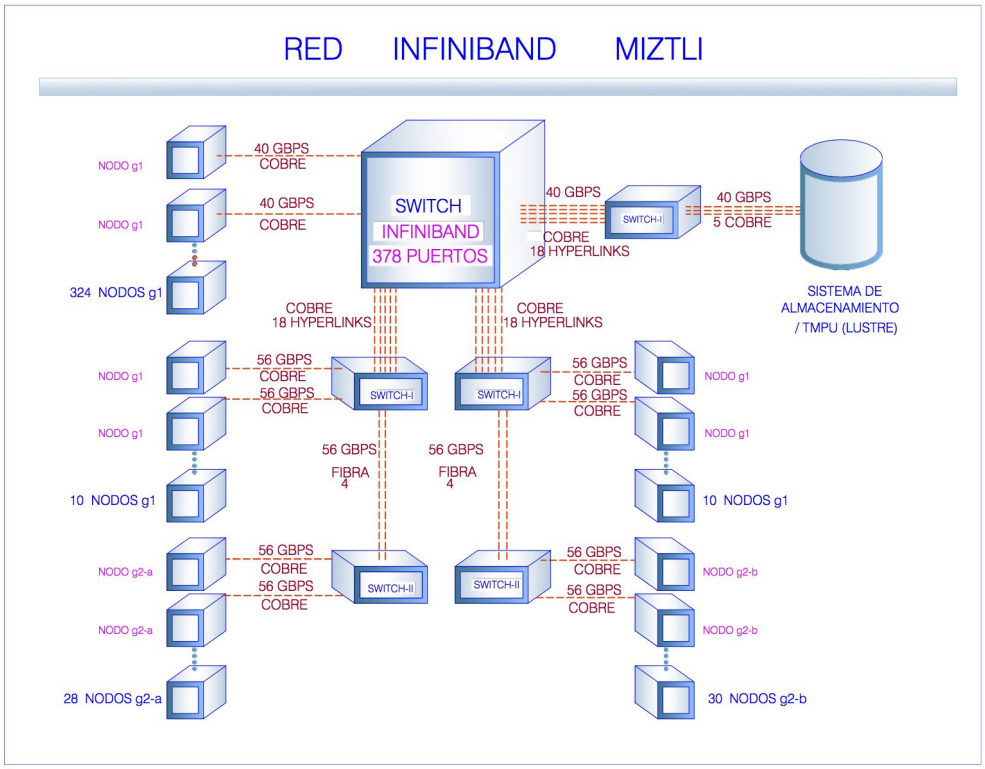

Figura 1: Esquema de la red Infiniband de Miztli.

En términos generales, el usuario solamente debe preocuparse por especificar el número de cores de su job, sin embargo, es importante que conozca y entienda los escenarios que se describen en las siguientes secciones.

5.2. Envío de jobs de memoria compartida

a) Un nodo parcial.

Como ya se mencionó, existen nodos tanto de 16 como de 20 cores. Si desea utilizar una cantidad de cores menor a 16, su job puede ser ejecutado en cualquier tipo de nodo. En este caso, puede usar como referencia los siguientes scripts:

```
#!/bin/bash

### SCRIPT PARA EJECUTAR JOBS DE PROGRAMAS OPENMP

#BSUB -q q_htc
#BSUB -n 8
#BSUB -R "span[hosts=1]"
#BSUB -oo salida
#BSUB -eo error
```

```
### LSB_DJOB_NUMPROC ES UNA VARIABLE QUE CONTIENE
### LA CANTIDAD DE CORES
### QUE HAN SIDO RESERVADOS PARA EJECUTAR EL JOB

export OMP_NUM_THREADS=$LSB_DJOB_NUMPROC

echo "Cantidad de threads $OMP_NUM_THREADS"
echo "Ejecucion de programa de OpenMP"
./app_openmp
```

```
#!/bin/bash

### SCRIPT PARA EJECUTAR JOBS DE GAUSSIAN

#BSUB -q  q_htc
#BSUB -n  8
#BSUB -R  "span[hosts=1]"
#BSUB -oo salida
#BSUB -eo error

module load gaussian/G09-E.01
input=test0001.com

### EL ARCHIVO DE ENTRADA DE GAUSSIAN SE MODIFICA
### PARA QUE LA DIRECTIVA
### %nprocshared CONTENGA LA CANTIDAD DE CORES
###  QUE HAN SIDO RESERVADOS PARA EL JOB

sed -i "s/\(%nprocshared=\).*/\1$LSB_DJOB_NUMPROC/g" $input

echo "Ejecucion de Gaussian "
g09 < $input
```

Es muy importante notar que para todos los programas de memoria compartida se tiene que utilizar siempre la opción:

```
#BSUB -R  "span[hosts=1]"
```

la cual concentra todos los cores en un sólo nodo; si no se utiliza está opción, el Sistema de Colas podrá reservar los cores en diferentes nodos de cálculo, provocando que la aplicación se ejecute de forma inadecuada o con un comportamiento no esperado en una aplicación de memoria compartida.

b) Usar todos los cores de un nodo.

Si no le importa que tipo de nodo usar, pero requiere usar todos los cores de un nodo, es necesario indicarle al sistema de colas que el número de cores solicitado depende del tipo de nodo asignado. Esto se puede realizar de la siguiente forma:

```
#!/bin/bash
#BSUB -q q_htc

### LA SIGUIENTE OPCION INDICA QUE SE PUEDEN USAR 16 O 20 CORES
```

```
#BSUB -n 16,20

#BSUB -R "span[hosts=1]"
#BSUB -oo salida
#BSUB -eo error

export OMP_NUM_THREADS=$LSB_DJOB_NUMPROC
echo "Cantidad de threads $OMP_NUM_THREADS"
echo "Ejecucion de programa de OpenMP"
./app_openmp
```

El sistema de colas buscará un nodo de 20 cores que esté completamente desocupado, y se lo asignará al job. Si no existe ninguno, buscará un nodo de 16 cores y se lo asignará al job.

c) Un tipo de nodo en específico.

Si requiere usar un tipo de nodo en específico, puede usar algunos de los siguientes ejemplos como base:

```
#!/bin/bash
#BSUB -q q_htc


#BSUB -n 20

### LA SIGUIENTE OPCION INDICA QUE EL NODO A USAR
### DEBE SER DE 20 CORES

#BSUB -m "g2_a g2_b"

#BSUB -R "span[hosts=1]"
#BSUB -oo salida
#BSUB -eo error

export OMP_NUM_THREADS=$LSB_DJOB_NUMPROC
echo "Cantidad de threads $OMP_NUM_THREADS"
echo "Ejecucion de programa de OpenMP"
./app_openmp
```

```
#!/bin/bash
#BSUB -q q_htc

#BSUB -n 16

### LA SIGUIENTE OPCION INDICA QUE EL NODO A USAR
### DEBE SER DE 16 CORES

#BSUB -m "g1"

#BSUB -R "span[hosts=1]"
#BSUB -oo salida
#BSUB -eo error
```

```
export OMP_NUM_THREADS=$LSB_DJOB_NUMPROC
echo "Cantidad de threads $OMP_NUM_THREADS"
echo "Ejecucion de programa de OpenMP"
./app_openmp
```

5.3. Envío de jobs de memoria distribuida

1. A cualquier tipo de nodo, con cualquier distribución de nodos.
   a) Forma tradicional.

   ```
   $ bsub -n NC < script.ls
   ```

   El trabajo iniciará si hay NC cores disponibles en cualquier grupo, sin importar la forma en que serán repartidos los procesos entre los nodos asignados.

   b) Número de cores variable.

   ```
   $ bsub -n nmin,nmax < script.ls
   ```

   El trabajo iniciará si al menos hay $nmin$ cores disponibles.

2. A un tipo de nodo, con una prioridad específica.
   a) Preferencia de un tipo de nodo.

   ```
   $ bsub -n NC -m "g2_a+2 g1+3 g2_b+1" < script.ls
   ```

   El trabajo comprobará si hay NC cores disponibles en el grupo $g1$ e iniciará si esta condición se cumple, de lo contrario realizará la misma verificación para $g2\_a$ y $g2\_b$, en el orden de prioridad especificado.

3. A un tipo de nodo específico.
   a) Eligiendo un tipo de nodo

   ```
   $ bsub -n NC -m "g2_b" < script.ls
   ```

   El trabajo iniciará si hay NC cores disponibles en un nodo del tipo $g2\_b$.

Referencias

Se recomienda consultar la página de manual de los comandos `bsub` y `bjob`.

Puede encontrar más información en:

Running jobs in IBM Platform LSF
https://www.ibm.com/support/knowledgecenter/SSETD4_9.1.3/lsf_kc_run_jobs.html

Créditos

Este es un material técnico realizado para la explicación del nuevo esquema del sistema de manejo de recursos de la supercomputadora Miztli, con la colaboración de:

Silvia Elizabeth Frausto Del Río. Diseño e implementación de la configuración de LSF para los requerimientos del nuevo esquema del sistema de colas. Concepción y programación de herramientas de consulta. Redacción de la sección "Herramientas de consulta".

Leobardo Itehua Rico. Concepción e implementación de pruebas para la ejecución de programas paralelos de memoria distribuida. Propuesta de casos, concepción de ejemplos y redacción de la sección "Envío de jobs de memoria distribuida".

Irving Carlos Álvarez Castillo. Concepción e implementación de pruebas para la ejecución de programas paralelos de memoria compartida. Propuesta de casos, concepción de ejemplos y redacción de la sección "Envío de jobs de memoria compartida".

José Luis Gordillo Ruiz. Propuesta del esquema de colas, estructura del presente documento y redacción de "Introducción", "Descripción de las colas" y "Consideraciones adicionales".

Camilo González González. Elaboración de la Figura 1. "Esquema de la Red Infiniband de Miztli".

Contactos

José Luis Gordillo Ruiz
Coordinador de Supercómputo, DGTIC-UNAM
Correo: jlgr@super.unam.mx
Teléfono: 5622-8529

Irving Carlos Álvarez Castillo
Paralelismo y Optimización
Coordinación de Supercómputo, DGTIC-UNAM
Correo: irving@super.unam.mx
Teléfono: 5622-8599

Leobardo Itehua Rico
Paralelismo y Optimización
Coordinación de Supercómputo, DGTIC-UNAM
Correo: itehua@super.unam.mx
Teléfono: 5622-8164

Silvia Elizabeth Frausto Del Río.
Administración de Supercómputo
Coordinación de Supercómputo, DGTIC-UNAM
Correo: silvia@super.unam.mx
Teléfono: 5622-8599