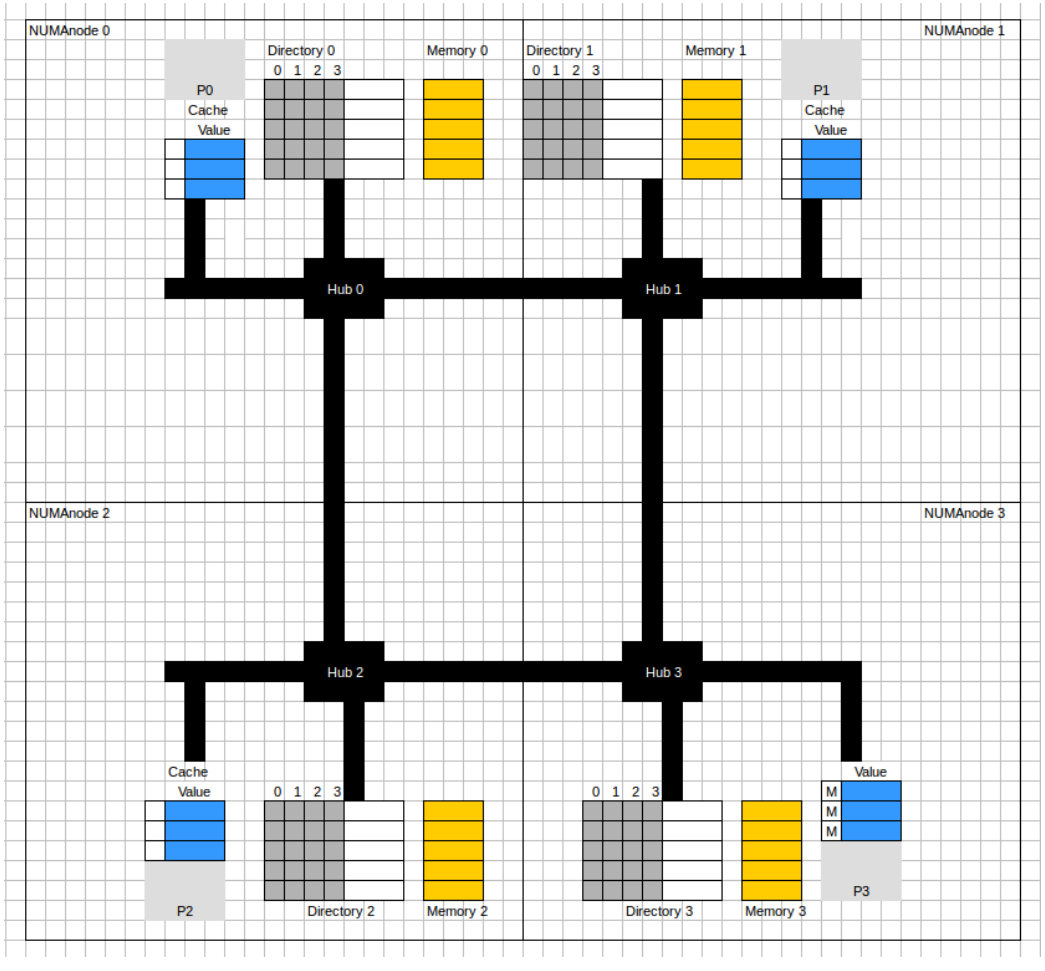
# NUMA Quiz 1

In a NUMA (Non-Uniform Memory Access time) multiprocessor there are two or more identical (NUMA) nodes, each one with a processor and its complete memory hierarchy, including a portion of main memory. The overall memory of the system is physically distributed among all the nodes but logically shared by all of them (i.e., the processor in any node can access to its main memory and also to any memory location in any other node through the interconnection network).

Which of the following statements are true?



A given physical memory address can only be stored in the memory of a single node, although multiple copies of the line containing that address may be temporarily stored in the cache memories of other nodes.

A given physical memory address can only be s ode, although multiple copies of the line containing stored in the cache memories of other nodes.

There are copies of a memory line but a data can not be in two places of main memory – It would be a mess!!!

In a NUMA system, instructions different from the conventional load and store are required to access to variables stored in other nodes.

A given physical memory address can only be stored in the memory of a single node, although multiple copies of the line containing that address may be temporarily stored in the cache memories of other nodes.

In a NUMA system, n the conventional load and store are required to access r nodes.

NO! In a NUMA system Memory is shared and Load/Store is enough

The way data is distributed among the different nodes of a NUMA multiprocessor system ...

Trieu-ne una:

- ... does not have any impact in the performance of the parallel application.
- ... is determined by the operating system based on a given data allocation policy (for example, first touch).
- ... dynamically changes with the objective of balancing the number of local accesses that are performed by the processors in the different NUMA nodes.
- ... is statically determined by the compiler, based on the accesses that are performed by the tasks in the parallel program.

A given physical memory address can only be stored in the memory of a single node, although multiple copies of the line containing that address may be temporarily stored in the cache memories of other nodes.

In a NUMA system, instructions different from the conventional load and store are required to access to variables stored in other nodes.

The way data is distributed among the different nodes of a NUMA multiprocessor system ...

Of course, it has, do you remember all the protocol? We will see an example later

Triu-ne una:

- ... does not have any impact in the performance of the parallel application.
- ... is determined by the operating system based on a given data allocation policy (for example, first touch).
- ... dynamically changes with the objective of balancing the number of local accesses that are performed by the processors in the different NUMA nodes.
- ... is statically determined by the compiler, based on the accesses that are performed by the tasks in the parallel program.

A given physical memory address can only be stored in the memory of a single node, although multiple copies of the line containing that address may be temporarily stored in the cache memories of other nodes.

In a NUMA system, instructions different from the conventional load and store are required to access to variables stored in other nodes.

The way data is distributed among the different nodes of a NUMA multiprocessor system ...

Triu-ne una:

- ... does not have any impact in the performance of the parallel application.
- ... is determined by the operating system based on a given data allocation policy (for example, first touch).
- ... dynamically changes with the objective of balancing the number of local accesses that are performed by the processors in the different NUMA nodes.
- ... is statically determined by the compiler, based on the accesses that are performed by the tasks in the parallel program.

OS determines where the Data is allocated and it is NOT moved

A given physical memory address can only be stored in the memory of a single node, although multiple copies of the line containing that address may be temporarily stored in the cache memories of other nodes.

In a NUMA system, instructions different from the conventional load and store are required to access to variables stored in other nodes.

The way data is distributed among the different nodes of a NUMA multiprocessor system ...

Triu-ne una:

- ... does not have any impact in the performance of the parallel application.
- ... is determined by the operating system based on a given data allocation policy (for example, first touch).
- ... dynamically changes with the objective of balancing the number of local accesses that are performed by the processors in the different NUMA nodes.
- ... is statically determined by the compiler, based on the accesses that are perfo el program.

First, data is not moved. Second, compiler cannot figure out that

A given physical memory address can only be stored in the memory of a single node, although multiple copies of the line containing that address may be temporarily stored in the cache memories of other nodes.

In a NUMA system, instructions different from the conventional load and store are required to access to variables stored in other nodes.

The way data is distributed among the different nodes of a NUMA multiprocessor system ...

Triu-ne una:

... does not have any impact in the performance of the

OS determines where the Data is allocated

... is determined by the operating system based on a given data allocation policy (for example, first touch).

... dynamically changes with the objective of balancing the number of local accesses that are performed by the processors in the different NUMA nodes.

... is statically determined by the compiler, based on the accesses that are performed by the tasks in the parallel program.

Assume that the coherence in a NUMA multiprocessor is based on a directory attached to the main memory in each node. The directory structure present in each node provides ...

Trieu-ne una:

- ... coherence information for the memory lines that are stored in the cache memory of the same node.
- ... information that allows a processor to find [illegible] nodes.
- ... information to keep coherent all possible copies [illegible] the memory of that node.
- ... information that allows a processor in that node to find the nearest node where to find a given memory address, in order to minimize the memory access time.

NO! Provides coherece information for Memory lines placed on the NUMA node, decided by the OS

Assume that the coherence in a NUMA multiprocessor is based on a directory attached to the main memory in each node. The directory structure present in each node provides ...

Trieu-ne una:

- ... coherence information for the memory lines that are stored in the cache memory of the same node.
- ... information that allows a processor to find the data that is allocated in other nodes.
- ... information to keep coherent all possible copi[illegible] the memory of that node.
- ... information that allows a processor in that node to [illegible] to find a given memory address, in order to minimize the [illegible]

IT DOESN'T HAVE ANY INFORMATION ABOUT WHICH MEMORY LINES ARE IN OTHER NUMAnodes

Assume that the coherence in a NUMA multiprocessor is based on a directory attached to the main memory in each node. The directory structure present in each node provides ...

Trieu-ne una:

- ... coherence information for the memory lines that are stored in the cache memory of the same node.
- ... information that allows a processor to find the data that is allocated in other nodes.
- ... information to keep coherent all possible copies in cache of the lines stored in the memory of that node.
- ... information that allows a processor in that node to find the nearest node where to find a given memory address, in order to minimize the memory access time.

IT DOESN'T HAVE ANY INFORMATION ABOUT WHICH MEMORY LINES ARE IN OTHER NUMAnodes

Assume that the coherence in a NUMA multiprocessor is based on a directory attached to the main memory in each node. The directory structure present in each node provides ...

Trieu-ne una:

- ... coherence information for the memory lines that are stored in the cache memory of the same node.
- ... information that allows a processor to find the data that is allocated in other nodes.
- ... information to keep coherent all possible copies in cache of the lines stored in the memory of that node.
- ... information that allows a processor in tha to find a given memory address, in order to minimiz

YES!, IT KEEPS INFORMATION ABOUT STATE OF THE MEMORY LINE (M,S,U) AND WHICH NUMAnode'S **HAVE COPIES**

Assume that the coherence in a NUMA multiprocessor is based on a directory attached to the main memory in each node. The directory structure present in each node provides ...

Trieu-ne una:

... coherence information for the memory lines that are stored in the cache memory of the same node.

... information that allows a processor to find the data that is allocated in other nodes.

... information to keep coherent all possible copies in cache of the lines stored in the memory of that node.

... information that allows a processor in th where to find a given memory address, in order to mi

YES!
A VECTOR OF BITS IS USED TO CONTROL COPIES IN THE NUMAnodes (1 bit x NUMAnode) and also BITS TO HAVE THE STATUS

In a NUMA multiprocessor system, with directory-based coherence protocol, the number of bits in each entry of the directory depends on the number of nodes in the system, with one or several additional bits to keep the state of the associated line.

The number of entries in the directory of a NUMA node ...

Trieu-ne una:

- ... is the total number of cache lines in the overall NUMA system, helping to identify which caches have a copy of a memory line.
- ... is determined by the maximum number of copies that are allowed for each line in main memory.
- ... is the number of lines that are stored in the main memory associated to it.
- ... depends on the number of NUMA nodes in the system in order to implement the list of nodes with remote copies.

The number of entries in the directory of a NUMA node ...

Trieu-ne una:

- ... is the total number of cache lines in the overall NUMA system, helping to identify which caches have a copy of a memory line.
- ... is determined by the maximum number of copies that are allowed for each line in main memory.
- ... is the number of lines that are stored in the main memory associated to it.
- ... depends on the number of NUMA nodes in the system in order to implement the list of nodes with remote copies.

THERE ARE AS MANY ENTRIES AS NUMBER OF LINES OF MEMORY SINCE WE KEEP THE INFROMATION PER MEMORY LINE

Assume a NUMA multiprocessor architecture with 1024 nodes, each node with a single processor and 24 GB of main memory, with directory-based MSU coherence protocol; memory lines are 128 bytes wide. In that system, which is the percentage of the whole main memory (**including both data and directory**) that is used by the directory to store all the information related to coherence?

Trieu-ne una:

- With the information provided one can not compute the number. You should have provided the size of the cache memory in each node to be able to compute the requested percentage.
- close to 200%
- close to 50%
- close to 100%

The number of entries in the directory of a NUMA node ...

Trieu-ne una:

- ... is the total number of cache lines in the overall NUMA system, helping to identify which caches have a copy of a memory line.
- ... is determined by the maximum number of copies that are allowed for each line in main memory.
- ... is the number of lines that are stored in the main memory associated to it.
- ... depends on the number of NUMA nodes in the system in order to implement the list of nodes with remote copies.

Assume a NUMA multiprocessor architecture with 1024 nodes, each node with a single processor and 24 GB of main memory, with directory-based MSU coherence protocol; memory lines are 128 bytes wide. In that system, which is the percentage of the whole main memory (**including both data and directory**) that is used by the directory to store all the information related to coherence?

Trieu-ne una:

- With the information provided one can not compute the number. You should have provided the size of the cache memory in each node to be able to compute the requested percentage.
- close to 200%
- close to 50%
- close to 100%

System:
- 24GB=24*2^30 bytes
- 128 bytes/line
- 1024 nodes

Directory entry
- 2 bits status/line
- 1024 bits presence/line

Total bytes per line?:
T=(1024+2)/8+128

% = 100*(1024+2)/8 / (T)
%= 50,1

False sharing cannot occur across nodes in a NUMA multiprocessor architecture because the directory structure attached to main memory provides information about the location of each variable in the same line.

In a NUMA multiprocessor architecture, false sharing implies the simultaneous existence of at least two copies of the same cache line in M state in the associated directory entry.

Two different processors in a cache-coherent multiprocessor architecture (either UMA or NUMA) continuously executing a count++ instruction originate a false sharing situation.

False sharing cannot occur across nodes in a NUMA multiprocessor architecture because the directory structure attached to main memory provides information about the location of each variable in the same line.

THE COHERENCE IS MAINTAINED AT MEMORY LINE GRANULARITY!

In a NUMA multiprocessor architecture, false sharing implies the simultaneous existence of at least two copies of the same cache line in M state in the associated directory entry.

Two different processors in a cache-coherent multiprocessor architecture (either UMA or NUMA) continuously executing a count++ instruction originate a false sharing situation.

False sharing cannot occur across nodes in a NUMA multiprocessor architecture because the directory structure attached to main memory provides information about the location of each variable in the same line.

THAT WOULD MEAN THAT THERE IS DATA NOT COHERENT!
IT IS NOT GOOD TO HAVE TWO M !!!

In a NUMA multiprocessor architecture, false sharing implies the simultaneous existence of at least two copies of the same cache line in M state in the associated directory entry.

Two different processors in a cache-coherent multiprocessor architecture (either UMA or NUMA) continuously executing a count++ instruction originate a false sharing situation.

False sharing cannot occur across nodes in a NUMA multiprocessor architecture because the directory structure attached to main memory provides information about the location of each variable in the same line.

In a NUMA multiprocessor architecture, false sharing implies the simultaneous existence of at least two copies of the same cache line in M state in the associated directory entry.

NO! THIS IS TRUE SHARING!

Two different processors in a cache-coherent multiprocessor architecture (either UMA or NUMA) continuously executing a count++ instruction originate a false sharing situation.