

NUMA -2- (Fallback Node)

📅 2016-06-03 (<http://jake.dothome.co.kr/numa-fallback-node/>) 👤 Moon Young-il

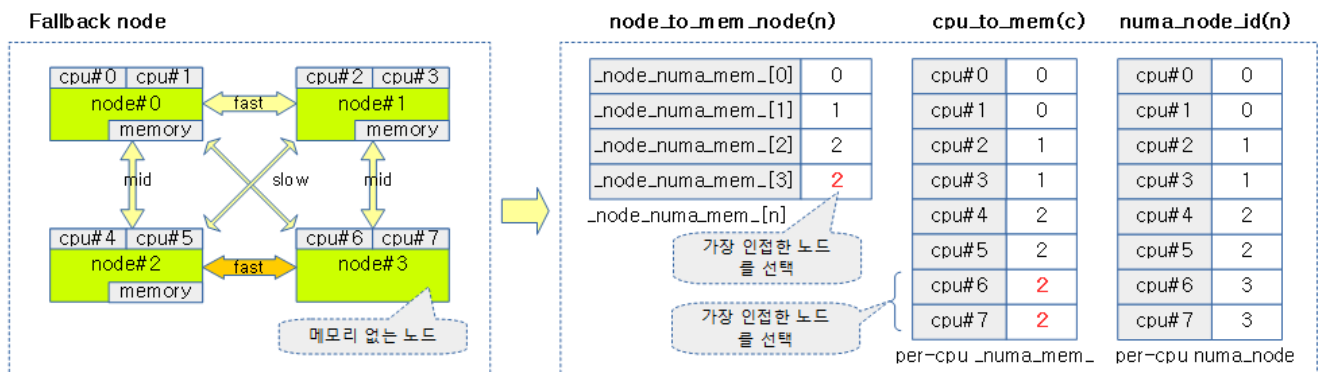
(<http://jake.dothome.co.kr/author/admin/>) 📁 Linux Kernel (<http://jake.dothome.co.kr/category/linux/>)

<kernel v5.10>

In systems with the NUMA architecture, there is an API that can be used to use adjacent memory banks if there is no memory bank on a particular node.

NUMA -2- (Fallback node)

When attempting to allocate memory on the current node, if memory does not exist on your node, the node chooses the nearest (fastest) node with memory as the fallback node.



(<http://jake.dothome.co.kr/wp-content/uploads/2016/06/fallback-node-1b.png>)

- node_numa_mem_[3]: Node#3 has no memory, so it contains the nearest node id 2 that has memory.
- per-cpu _numa_mem_: Node #6가 7, which corresponds to CPU#3 and CPU#3, but has no memory, so it contains the nearest node ID 2 that has memory.
- per-cpu numa_node: Node #6 corresponding to CPU#7 and CPU#3. (With or without memory)

numa_node_id()

include/linux/topology.h

```

01 #ifdef CONFIG_USE_PERCPU_NUMA_NODE_ID
02 #ifndef numa_node_id
03 /* Returns the number of the current Node */
04 static inline int numa_node_id(void)
05 {
06     return raw_cpu_read(numa_node);
07 }
08 #endif
09 #else
10 /* Returns the number of the current Node. */
11 #ifndef numa_node_id

```

```

12 | static inline int numa_node_id(void)
13 | {
14 |     return cpu_to_node(raw_smp_processor_id());
15 | }
16 | #endif
17 | #endif

```

Returns the node ID to which the CPU currently belongs.

- If you use the CONFIG_USE_PERCPU_NUMA_NODE_ID kernel option, you can get the numa node ID by reading the numa_node value in a per-cpu way, and if you don't use this option, you can get the node ID from the current CPU number.
- It's not uncommon to use NUMA on a 32-bit arm, but if you do, using the per-cpu approach allows for faster processing because the node's calculations use TPIDRPRW registers instead of memory.
- Architectures such as x86, PowerPC, and IA64 are also ready for quick implementation.

numa_mem_id()

include/linux/topology.h

```

01 | #ifdef CONFIG_HAVE_MEMORYLESS_NODES
02 | #ifndef numa_mem_id
03 |     /* Returns the number of the nearest Node with memory */
04 |     static inline int numa_mem_id(void)
05 |     {
06 |         return raw_cpu_read(_numa_mem_);
07 |     }
08 | #endif
09 | #else
10 | #ifndef numa_mem_id
11 |     /* Returns the number of the nearest Node with memory */
12 |     static inline int numa_mem_id(void)
13 |     {
14 |         return numa_node_id();
15 |     }
16 | #endif
17 | #endif

```

Returns the node ID to which the CPU currently belongs, and if this node has no memory, it knows the ID of the nearest node in memory.

- If you use the CONFIG_HAVE_MEMORYLESS_NODES kernel option, it returns the node ID contained in the per-cpu _numa_mem_, which returns the ID of the nearest node with memory on the current node if the node to which the CPU belongs does not have memory, and if you don't use the kernel option, it simply returns the node ID to which the current CPU belongs.
- Consultation:
 - Enable memoryless node on x86 platforms (<https://lwn.net/Articles/605155/>) | LWN.net
 - Enable memoryless node support for x86 (<https://lwn.net/Articles/654702/>) | LWN.net

node_to_mem_node()

include/linux/topology.h

```

01 | #ifdef CONFIG_HAVE_MEMORYLESS_NODES

```

```

02 | #ifndef node_to_mem_node
03 | static inline int node_to_mem_node(int node)
04 | {
05 |     return _node_numa_mem_[node];
06 | }
07 | #endif
08 | #else
09 | #ifndef node_to_mem_node
10 | static inline int node_to_mem_node(int node)
11 | {
12 |     return node;
13 | }
14 | #endif
15 | #endif

```

If the specified node has memory, it returns the specified node ID as is, and if there is no memory, it knows the nearest node ID with memory.

- If you use the CONFIG_HAVE_MEMORYLESS_NODES kernel option, it returns an array of `_node_numa_mem_[]` values. The `_node_numa_mem_[]` array contains the same value as the nodeid used as an array argument, but in the case of a node with no memory, it contains the nearest node id with memory.

Related Global Variables

per-cpu numa_node variable

mm/page_alloc.c

```

1 | #ifdef CONFIG_USE_PERCPU_NUMA_NODE_ID
2 | DEFINE_PER_CPU(int, numa_node);
3 | #endif

```

It contains the node ID to which each CPU belongs.

_node_numa_mem Variables

mm/page_alloc.c

```

1 | #ifdef CONFIG_HAVE_MEMORYLESS_NODES
2 | int _node_numa_mem_[MAX_NUMNODES];
3 | #endif

```

It contains the node ID corresponding to each node in the array, but if that node is a node with no memory, it contains the ID of the adjacent node with memory as the alternate node.

per-cpu _numa_mem_ variables

mm/page_alloc.c

```

1 | #ifdef CONFIG_HAVE_MEMORYLESS_NODES
2 | /*
3 |  * N.B., Do NOT reference the '_numa_mem_' per cpu variable directly.
4 |  * It will not be defined when CONFIG_HAVE_MEMORYLESS_NODES is not defin
   | ed.

```

```
5 | * Use the accessor functions set_numa_mem(), numa_mem_id() and cpu_to_m  
   | em()  
6 | * defined in <linux/topology.h>.  
7 | */  
8 | DEFINE_PER_CPU(int, _numa_mem_);           /* Kernel "local memory"  
   | node */  
9 | #endif
```

It contains the node ID to which the CPU belongs, but if the node is a node with no memory, it contains the ID of a neighboring node with memory as a replacement node.

consultation

- NUMA -1- (ARM64 initialization) (<http://jake.dothome.co.kr/numa-1/>) | Qc
- NUMA -2- (Fallback Node) | Sentence C – Current post
- NODE Bitmap (API) (<http://jake.dothome.co.kr/node-api>) | Qc

LEAVE A COMMENT

Your email will not be published. Required fields are marked with *

Comments

name *

email *

Website

WRITE A COMMENT

◀ Slub Memory Allocator -3- (Create Cache) (<http://jake.dothome.co.kr/slub-cache-create/>)

Slub Memory Allocator -12- (debugging slub) ▶ (<http://jake.dothome.co.kr/slub-debug/>)

Munc Blog (2015 ~ 2024)