Automatically bind swap device to numa node

If the system has more than one swap device and swap device has the node information, we can make use of this information to decide which swap device to use in get_swap_pages() to get better performance.

How to use this feature

Swap device has priority and that decides the order of it to be used. To make use of automatically binding, there is no need to manipulate priority settings for swap devices. e.g. on a 2 node machine, assume 2 swap devices swapA and swapB, with swapA attached to node 0 and swapB attached to node 1, are going to be swapped on. Simply swapping them on by doing:

```
# swapon /dev/swapA
# swapon /dev/swapB
```

Then node 0 will use the two swap devices in the order of swapA then swapB and node 1 will use the two swap devices in the order of swapB then swapA. Note that the order of them being swapped on doesn't matter.

A more complex example on a 4 node machine. Assume 6 swap devices are going to be swapped on: swapA and swapB are attached to node 0, swapC is attached to node 1, swapD and swapE are attached to node 2 and swapF is attached to node3. The way to swap them on is the same as above:

```
# swapon /dev/swapA
# swapon /dev/swapB
# swapon /dev/swapC
# swapon /dev/swapD
# swapon /dev/swapE
# swapon /dev/swapF
```

Then node 0 will use them in the order of:

```
swapA/swapB -> swapC -> swapD -> swapE -> swapF
```

swapA and swapB will be used in a round robin mode before any other swap device.

node 1 will use them in the order of:

```
swapC -> swapA -> swapB -> swapD -> swapE -> swapF
```

node 2 will use them in the order of:

```
swapD/swapE -> swapA -> swapB -> swapC -> swapF
```

Similaly, swapD and swapE will be used in a round robin mode before any other swap devices.

node 3 will use them in the order of:

```
swapF -> swapA -> swapB -> swapC -> swapD -> swapE
```

Implementation details

The current code uses a priority based list, swap_avail_list, to decide which swap device to use and if multiple swap devices share the same priority, they are used round robin. This change here replaces the single global swap_avail_list with a per-numa-node list, i.e. for each numa node, it sees its own priority based list of available swap devices. Swap device's priority can be promoted on its matching node's swap_avail_list.

The current swap device's priority is set as: user can set a >=0 value, or the system will pick one starting from -1 then downwards. The priority value in the swap_avail_list is the negated value of the swap device's due to plist being sorted from low to high. The new policy doesn't change the semantics for priority >=0 cases, the previous starting from -1 then downwards now becomes starting from -2 then downwards and -1 is reserved as the promoted value. So if multiple swap devices are attached to the same node, they will all be promoted to priority -1 on that node's plist and will be used round robin before any other swap devices.