NUMA resource associativity

Associativity represents the groupings of the various platform resources into domains of substantially similar mean performance relative to resources outside of that domain. Resources subsets of a given domain that exhibit better performance relative to each other than relative to other resources subsets are represented as being members of a sub-grouping domain. This performance characteristic is presented in terms of NUMA node distance within the Linux kernel. From the platform view, these groups are also referred to as domains.

PAPR interface currently supports different ways of communicating these resource grouping details to the OS. These are referred to as Form 0, Form 1 and Form2 associativity grouping. Form 0 is the oldest format and is now considered deprecated.

Hypervisor indicates the type/form of associativity used via "ibm,architecture-vec-5 property". Bit 0 of byte 5 in the "ibm,architecture-vec-5" property indicates usage of Form 0 or Form 1. A value of 1 indicates the usage of Form 1 associativity. For Form 2 associativity bit 2 of byte 5 in the "ibm,architecture-vec-5" property is used.

Form 0

Form 0 associativity supports only two NUMA distances (LOCAL and REMOTE).

Form 1

With Form 1 a combination of ibm,associativity-reference-points, and ibm,associativity device tree properties are used to determine the NUMA distance between resource groups/domains.

The “ibm,associativity” property contains a list of one or more numbers (domainID) representing the resource’s platform grouping domains.

The “ibm,associativity-reference-points” property contains a list of one or more numbers (domainID index) that represents the 1 based ordinal in the associativity lists. The list of domainID indexes represents an increasing hierarchy of resource grouping.

ex: { primary domainID index, secondary domainID index, tertiary domainID index.. }

Linux kernel uses the domainID at the primary domainID index as the NUMA node id. Linux kernel computes NUMA distance between two domains by recursively comparing if they belong to the same higher-level domains. For mismatch at every higher level of the resource group, the kernel doubles the NUMA distance between the comparing domains.

Form 2

Form 2 associativity format adds separate device tree properties representing NUMA node distance thereby making the node distance computation flexible. Form 2 also allows flexible primary domain numbering. With numa distance computation now detached from the index value in "ibm,associativity-reference-points" property, Form 2 allows a large number of primary domain ids at the same domainID index representing resource groups of different performance/latency characteristics.

Hypervisor indicates the usage of FORM2 associativity using bit 2 of byte 5 in the "ibm,architecture-vec-5" property.

"ibm,numa-lookup-index-table" property contains a list of one or more numbers representing the domainIDs present in the system. The offset of the domainID in this property is used as an index while computing numa distance information via "ibm,numa-distance-table".

prop-encoded-array: The number N of the domainIDs encoded as with encode-int, followed by N domainID encoded as with encode-int

For ex: "ibm,numa-lookup-index-table" = {4, 0, 8, 250, 252}. The offset of domainID 8 (2) is used when computing the distance of domain 8 from other domains present in the system. For the rest of this document, this offset will be referred to as domain distance offset.

"ibm,numa-distance-table" property contains a list of one or more numbers representing the NUMA distance between resource groups/domains present in the system.

prop-encoded-array: The number N of the distance values encoded as with encode-int, followed by N distance values encoded as with encode-bytes. The max distance value we could encode is 255. The number N must be equal to the square of m where m is the number of domainIDs in the numa-lookup-index-table.

For ex: ibm,numa-lookup-index-table = <3 0 8 40>; ibm,numa-distace-table = <9>, /bits/ 8 < 10 20 80 20 10 160 80 160 10>;

```
   | 0    8   40
---|------------
   |
0  | 10   20  80
   |
8  | 20   10  160
   |
```

```
   40| 80   160  10
```

A possible "ibm,associativity" property for resources in node 0, 8 and 40

{ 3, 6, 7, 0 } { 3, 6, 9, 8 } { 3, 6, 7, 40}

With "ibm,associativity-reference-points" { 0x3 }

"ibm,lookup-index-table" helps in having a compact representation of distance matrix. Since domainID can be sparse, the matrix of distances can also be effectively sparse. With "ibm,lookup-index-table" we can achieve a compact representation of distance information.