MP2: Solution Outline

MyKV works in a system of 4 machines (Machine 1 through 4) and implements a distributed version of a perfect binary search tree with 6 levels. Each node of the binary search tree has 2 children. All leaves are at the bottom level. So, there are 32 leaves and 63 total nodes in this data structure.

The goal of this MP is to do both storage and lookup load balancing, i.e. to embed the nodes in the machines in a way so that: i) storage is evenly distributed across the machines, and ii) the lookup traffic is evenly distributed across the machines.

Storage load balancing is straight forward. As only the leaves are responsible for storing the keys, having each machine store $32/4 = 8$ of the leaves satisfies the storage load balancing requirement.

For lookup load balancing, note that the root node (level = 1) will receive all the requests. Therefore, load on that node = 1. Each of the nodes on level 2 will receive and forward $\frac{1}{2}$ of the requests. So, the load on each of the level 2 nodes = $\frac{1}{2}$. Table presents the look up load on the nodes at each level of the binary tree.

Level	Lookup load per node	# of nodes
1	1	1
2	$\frac{1}{2}$	2
3	$\frac{1}{4}$	4
4	$\frac{1}{8}$	8
5	$\frac{1}{16}$	16

Table 1: Lookup load on the nodes

To embed the nodes into the machines we have to ensure that the load on each of the machines is equal. There are multiple ways to come up with such an embedding. One possible solution is to have each of the machines host each of the first 4 levels of the binary tree (node 1 is hosted at machine 1, nodes 2,3 are hosted at machine 2, and so on) and divide the nodes at level 5 equally into the 4 machines. This configuration is presented in Figure 1.

Note that, the summation of the lookup loads on the nodes of a specific level = 1. Therefore, the lookup load on each of the machines is = $1 + \frac{1}{4} = \frac{5}{4}$.

Other embeddings are also possible/acceptable.

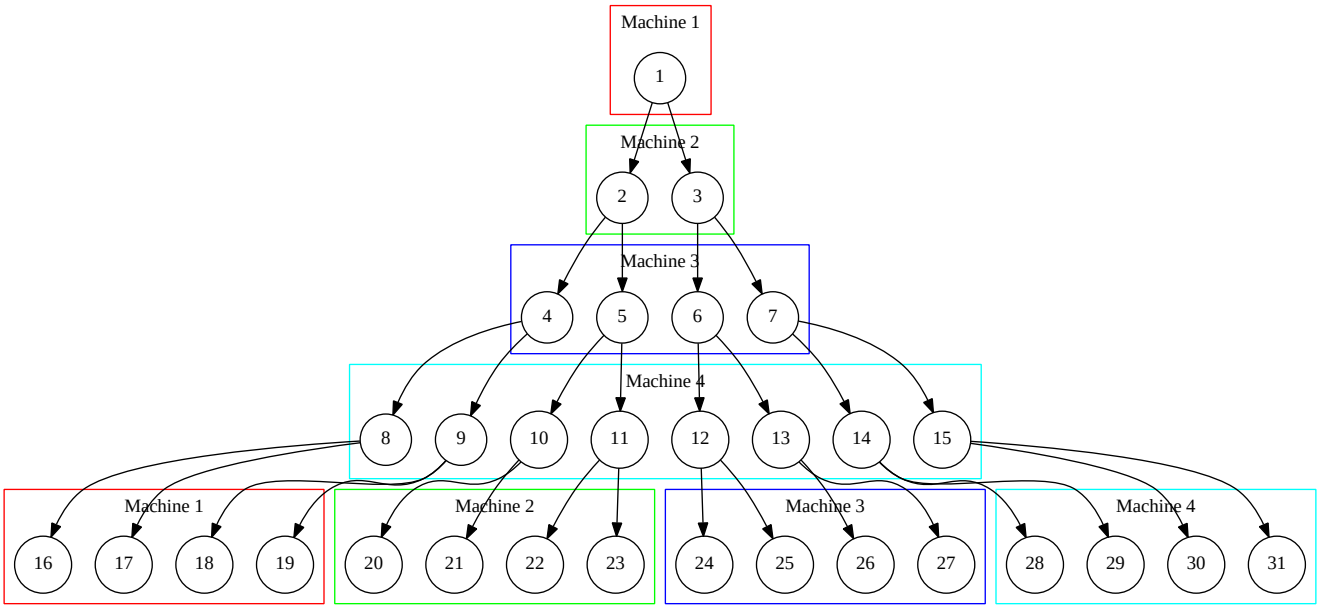

Figure 1: Embedding of the nodes to the machines