A cluster oriented model for dynamically balanced DHTs

José Rufino,[*] Albano Alves, José Exposto
Polytechnic Institute of Bragança
5300-854 Bragança, Portugal
{rufino,albano,exp}@ipb.pt

António Pina
University of Minho
4710-057 Braga, Portugal
pina@di.uminho.pt

Abstract

In this paper, we refine previous work on a model for a Distributed Hash Table (DHT) with support to dynamic balancement across a set of heterogeneous cluster nodes. We present new high-level entities, invariants and algorithms developed to increase the level of parallelism and globally reduce memory utilization.

In opposition to a global distribution mechanism, that relies on complete knowledge about the current distribution of the hash table, we adopt a local approach, based on the division of the DHT into separated regions, that possess only partial knowledge of the global hash table.

Simulation results confirm the hypothesis that the increasing of parallelism has as counterpart the degradation of the quality of the balancement achieved with the global approach. However, when compared with Consistent Hashing and our global approach, the same results clarify the relative merits of the extension, showing that, when properly parameterized, the model is still competitive, both in terms of the quality of the distribution and scalability.

1 Introduction

When designing Distributed Data Structures (DDSs) the possible heterogeneity of the nodes that will host data is often an issue not considered, specially if the target environment is the cluster, typically made of homogeneous nodes (essentially for administrative reasons). There are, however, valid reasons for heterogeneity in a cluster: **a)** economical reasons may impose the coexistence of machines from different generations; **b)** some tasks require specialized nodes.

Our first effort toward the definition of a model for a cluster oriented DDS, with support for heterogeneous nodes, was presented in [7]. The model proposed a novel approach to the design of a Distributed Hash Table (DHT), that supports the following major features: **a)** the share of a DHT handled by each cluster node is as a function of the amount of the computational resources it enrolls in the DHT; **b)** the enrollment level of each cluster node in a DHT is allowed to change dynamically; **c)** cluster nodes may dynamically join or leave the DHT.

The approach followed may be classified as *global*, once the balancement of the DHT requires the involvement of the totality of the cluster nodes enrolled in it, and each node must preserve global knowledge about the current distribution of the hash table.

In this paper, we extend our model with new entities, invariants and mechanisms that allow for the building of DHTs with improved performance and scalability. In contrast to the global perspective of the previous work, we name the new approach as *local*, once it only requires partial knowledge about the distribution of the hash table; this is accomplished by having the DHT subdivided into new high-level structures, that may evolve independently in time, with minimum coordination between cluster nodes, thus allowing to disperse and reduce the load of the overall cluster.

Simulation results clarify the relative merits of the extension demonstrating that by using the local approach there is a compromise between the quality of the dynamic balancement of a DHT and the desired parallelism and scalability.

The remaining of the paper is organized as follows. Section 2 revises our previous work, section 3 introduces the local approach, section 4 presents its evaluation, section 5 discusses related work and section 6 concludes.

2 Base model

In this section, we review the main concepts of our base model for a cluster oriented DHT, once they provide the foundations for the work we present in this paper.

2.1 Entities

In the global approach, the model comprises a set of entities. Their structural organization is shown in figure 1.

[*] Supported by PRODEP III, through the grant 5.3/N/199.006/00, and SAPIENS, through the grant 41739/CHS/2001.

0-7695-2132-0/04/$17.00 (C) 2004 IEEE

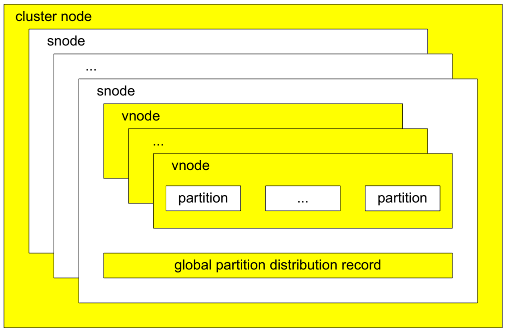

Figure 1. Entities of the model (global approach).

2.1.1 Software nodes

A DHT is primarily organized as a set of *software nodes* (or *snodes*), which are active software entities that manage parts of the DHT. A cluster node may host several snodes, each one specific to a different DHT.

2.1.2 Virtual nodes

The model provides *coarse–grain* and *fine–grain* balancement. Coarse-grain balancement is based on the definition of a certain number of *virtual nodes* (or *vnodes*) per snode, that translate the *enrollment level* of the snode in a DHT.

The enrollment level of a snode in a DHT primarily depends on the amount of local resources bound to the DHT. Such amount is not necessarily static: for instance, a snode may start by reserving a certain amount of secondary storage for a DHT; later, that amount may change in result of on–line disk repartitioning or hot–swapping mechanisms. In addition, the enrollment level should be a function of the relative performance between the cluster nodes, which may be assessed previously to the running of the DHT.

2.1.3 Partitions

Fine-grain balancement is based on the definition of a certain number of *partitions* per vnode. This number is allowed to fluctuate between well defined bounds, during the creation or deletion of vnodes. A *partition* is a contiguous subset of the range of the hash function.

2.1.4 Global Partition Distribution Record

Every snode hosts a copy of the *global partition distribution record* (GPDR). The GPDR is a table that registers the number of partitions per each vnode of the DHT.

2.2 Invariants

Let h be a hash function of range $R_h = \{i \in \mathbb{N}_0 : 0 \leq i < 2^{B_h}\}$, where B_h is the (fixed) number of bits of any hash index i. In the global approach, the previous entities and their relationships conform to the following invariants:

$\mathcal{G}_1$: R_h is fully divided into non–overlapping partitions;

$\mathcal{G}_2$: the overall number of partitions, P, is always a power of 2;

$\mathcal{G}_3$: every partition has the same size $S = 2^{B_h}/P$;

$\mathcal{G}_4$: for any vnode v, its number of partitions, P_v, is bounded: $P_{min} \leq P_v \leq P_{max}$, where P_{min} is a (fixed) power of 2 and $P_{max} = 2.P_{min}$;

$\mathcal{G}_5$: when the overall number of vnodes, V, is a power of 2, any vnode will have P_{min} partitions.

2.3 Goal

For any DHT based in the hash function h, the goal of the model is to ensure that each vnode is responsible for a similar share of R_h.

More precisely, if Q_v is the fraction/quota of R_h specific to the vnode v, the model aims to minimize $\sigma(Q_v, \overline{Q}_v)$, the standard deviation of all values of Q_v from the (ideal) average $\overline{Q}_v$. The quota Q_v of a vnode v is calculated by summing up the size of all partitions bound to v, and then dividing the result by the size of the range of h, 2^{B_h}.

The relative standard deviation, $\overline{\sigma}(Q_v, \overline{Q}_v) = \sigma(Q_v, \overline{Q}_v)/\overline{Q}_v$, weights the standard deviation against the average. This measure, often expressed in percentage, is more intuitive than $\sigma(Q_v, \overline{Q}_v)$, and so is used instead.

2.4 Quality metric

In general, if X_i and Y_i represent two series of numbers, such that $Y_i = c.X_i$, for any i, with c constant, then $\sigma(Y_i, \overline{Y}_i) = c.\sigma(X_i, \overline{X}_i)$ and $\overline{\sigma}(Y_i, \overline{Y}_i) = \overline{\sigma}(X_i, \overline{X}_i)$.

If we consider that, in the global approach, all partitions share the same size S, then $Q_v = (P_v.S)/2^{B_h} = c.P_v$, for any vnode v. It then follows that $\overline{\sigma}(Q_v, \overline{Q}_v) = \overline{\sigma}(P_v, \overline{P}_v)$, meaning that, in the global approach, we may also use $\overline{\sigma}(P_v, \overline{P}_v)$ to measure the quality of balancement.

Thus, the assignment of partitions to vnodes should minimize $\sigma(P_v, \overline{P}_v)$. To achieve this goal, it is necessary to carefully select the vnodes that will handover partitions (and in what number) whenever a new vnode is created.

2.5 Creation of vnodes

A snode triggers the creation of a vnode by issuing a *creation request* to the totality of the snodes of the DHT. The request will be completed only when the GPDR becomes synchronized in all snodes and all the necessary transfer of partitions have been concluded.

The following algorithm is executed by all the snodes in the DHT to handle the creation of a new vnode:

0-7695-2132-0/04/$17.00 (C) 2004 IEEE

```
1. create a new entrance in the local GPDR table, for the
   new vnode, and set its number of partitions as zero;
2. compute $\overline{\sigma}(P_v, \overline{P}_v)$;
3. sort the entrances of the local GPDR table by the number
   of partitions of each vnode and find the vnode with
   more partitions (the victim vnode);
4. if removing one partition from the victim vnode and
   giving it to the new vnode decreases $\overline{\sigma}(P_v, \overline{P}_v)$ then
     (a) if the snode hosts the victim vnode then choose a
         victim partition from it and schedule/perform its
         transfer to the new vnode endif
     (b) go to step 3;
   else stop; endif
```

As a consequence of the continuous creation of vnodes, the number of partitions contained at each vnode evolves, following a pattern enforced by the invariants and the reassignment algorithm, previously presented.

To conform to invariant $\mathcal{G}_5$, when V (the overall number of vnodes) is a power of 2, all vnodes must have P_{min} partitions, thus ensuring that R_h is perfectly balanced across all the vnodes. In this situation, when a new vnode is created, some of the older vnodes will have to handover some of its partitions, accordingly to the assignment algorithm. Because invariant $\mathcal{G}_4$ does not allow $P_v < P_{min}$, all the older vnodes binary split their own partitions, doubling its number to $P_v = P_{max}$, which is the maximum number of partitions per vnode allowed by the invariant.

The continuous creation of vnodes will force the number of partitions on the existing vnodes to decrease toward P_{min}. At some moment, the overall number of vnodes, V, will double, reaching the next power of 2. In that moment, for any vnode v, the number of partitions will be exactly $P_v = P_{min}$ and R_h will be, again, perfectly balanced across all vnodes.

3 Local approach

In [7], we have shown that the global approach achieves a high quality of balancement of the hash table across the set of vnodes. However, it requires each snode to have global knowledge about the partition distribution across the totality of the snodes in the DHT. In addition, as every snode is, necessarily, involved in the creation of every vnode, consecutive creations of vnodes are executed serially, thus limiting the parallelism and reducing the scalability of the DHT to a small number of snodes.

The local approach defines new structures and algorithms that allow for the logical definition of regions of the DHT that may evolve independently in the time, requiring only partial knowledge of the overall partition distribution, thus promoting scalability and enhancing parallelism.

3.1 Groups of vnodes

In the local approach, the global set of vnodes is fully divided in mutually exclusive subsets, named *groups*. Within each group, balancement is based on the same algorithm used by the global approach, though restricted to the vnode set of the group. Local balancement events may take place simultaneously at different groups.

Moreover, because the number of vnodes in each group is allowed to fluctuate between strict bounds, the overall number of groups may change, as vnodes are created. Such variation in the number of groups provides for a dynamic and adaptive level of parallelization.

In figure 2 we show two snodes of a DHT for which the global set of vnodes is currently divided in at least two groups (group 0 and group 1). As shown in the figure, the subset of vnodes that make each group is typically scattered among several snodes of the DHT.

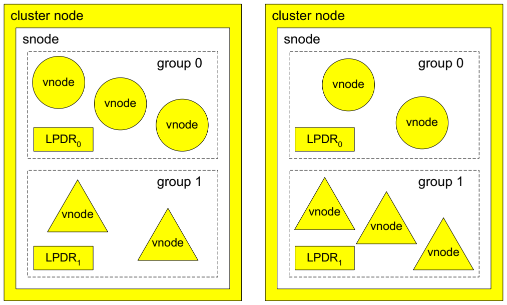

Figure 2. Entities of the model (local approach).

3.2 Local Partition Distribution Record

Each snode keeps an instance of the *Local Partition Distribution Record* (LPDR) of each group in which participate local vnodes. The LPDR of a group keeps information about the number of partitions bound to each vnode of the group. A LPDR is a table that may be viewed as a downsized version of the GPDR, having its same basic structure.

3.3 Invariants

To be able to adapt to the new objectives, the local approach introduces the following two new invariants:

$\mathcal{L}_1$: the global set of vnodes is fully divided into non–overlapping groups;

$\mathcal{L}_2$: for any group g, its number of vnodes, V_g, is bounded: $V_{min} \leq V_g \leq V_{max}$, where V_{min} is a (fixed) power of 2 and $V_{max} = 2.V_{min}$;

All invariants of the global approach are inherited, though modified to reflect the aggregation of vnodes in groups:

$\mathcal{G}_{1'}$: R_h is fully divided into non–overlapping partitions;

$\mathcal{G}_{2'}$: the overall number of partitions of a group[1] g, P_g, is always a power of 2;

$\mathcal{G}_{3'}$: every partition of a group g has the same size $S_g = 2^{B_h}/2^{l_g}$, where l_g is the common splitlevel of all partitions of g;

$\mathcal{G}_{4'}$: for any vnode v of a group g, its number of partitions, $P_{v,g}$, is bounded: $P_{min} \leq P_{v,g} \leq P_{max}$, where P_{min} is a (fixed) power of 2 and $P_{max} = 2.P_{min}$;

$\mathcal{G}_{5'}$: when the number of vnodes in a group g, V_g, is a power of 2, any vnode of the group will have P_{min} partitions.

3.4 Splitlevel

Considering that, in our model, every partition of R_h results from the *binary split* (division, in two equal parts) of another partition, the *splitlevel* of a partition may be defined as the number of binary splits needed, departing from R_h, to reach the current size of the partition. Thus, a partition in splitlevel l will have $1/2^l$ the size of R_h (which is 2^{B_h}).

In the global approach, all partitions share the same splitlevel l and so they have the same size $S = 2^{B_h}/2^l$ (implying that $P = 2^l$, accordingly with invariant $\mathcal{G}_3$).

In the local approach, partitions are not guaranteed to share the same splitlevel/size unless, accordingly with invariant $\mathcal{G}_{3'}$, they belong to vnodes of the same group.

3.5 Quality metric

Under the local approach, the goal of the model remains the same, that is, to ensure that every vnode has a similar share of the DHT.

However, $\overline{\sigma}(P_v, \overline{P}_v)$ can no longer be used in place of $\overline{\sigma}(Q_v, \overline{Q}_v)$ to measure the quality of the balancement, once the equality $\overline{\sigma}(Q_v, \overline{Q}_v) = \overline{\sigma}(P_v, \overline{P}_v)$ is not assured: the size of partitions will now depend on their specific group containers; as a consequence, $Q_{v,g} = (P_{v,g}.S_g)/2^{B_h} \neq c.P_{v,g}$, where $Q_{v,g}$ is the quota of the vnode v from the group g and c is a constant.

Thus, for the reasons presented above, $\overline{\sigma}(Q_v, \overline{Q}_v)$ is the only valid quality metric for the local approach.

3.6 Creation of vnodes

The creation of a new vnode starts by selecting a group for it: a random number $r \in R_h$ is chosen and a lookup is performed in order to find the vnode which holds the partition to where r belongs; we name this vnode as the *victim vnode*, and its group as the *victim group*.

The analysis of the LPDR of the victim group, located at the snode that hosts the victim vnode, allows to identify the snodes that host vnodes of the victim group[2]. These snodes will apply, in conjunction, the same algorithm used in the global approach, with base on the LPDR of the victim group (now including the new vnode, initially with zero partitions). Afterward, the number of partitions becomes balanced among all the vnodes of the victim group, and all copies of the LPDR become synchronized.

3.7 Creation of groups

The creation of a new group occurs whenever: **a)** the first vnode of a DHT is created, and so is the first group; **b)** the victim group chosen during the creation of a new vnode is already full (*i.e.*, it already contains V_{max} vnodes).

The first group (group 0), will always be elected as the victim group when creating the first V_{max} vnodes; as such, $1 \leq V_0 \leq V_{max}$, which is the sole exception to invariant $\mathcal{L}_2$, that states $V_{min} \leq V_g \leq V_{max}$, for any group g.

When a victim group is full, trying to add it a new vnode triggers the split of the group into two groups, each one with V_{min} vnodes, randomly selected from the original victim group. One of these two groups will then be randomly chosen to be the container of the new vnode.

3.7.1 Group identifiers

Each group is identified by an integer $g \in \mathbb{N}_0$, using a scheme that allows to define a unique global identifier, in an autonomous, decentralized way. The process is illustrated in figure 3, in which numbers represented in base 2, and their equivalents in base 10, are properly denoted.

The first group is group 0_2; when the first group becomes full, it splits in groups 0_2 and 1_2. Afterward, each time a group splits, the resulting groups inherit its binary identifier, prefixed either by the binary digit **0** or **1**.

By following this scheme, only the snode that coordinates the splitting of a group needs to be involved in the definition of the identifiers for the resulting groups.

[1] *I.e.*, the total number of partitions bound to all the vnodes of the group.

[2] Like in the GPDR, vnodes in the LPDR are identified by their canonical name, which follows the generic format `snode_id.vnode_id`.

0-7695-2132-0/04/$17.00 (C) 2004 IEEE

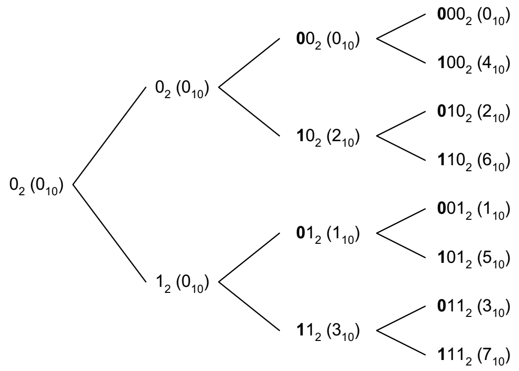

Figure 3. Generation of unique group identifiers.

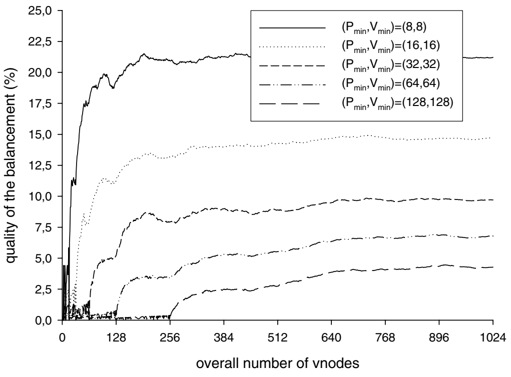

Figure 4. $\overline{\sigma}(Q_v, \overline{Q}_v)$ **when** $P_{min} = V_{min}$**.**

4 Evaluation

We now present the results of a preliminary evaluation of the local approach and compare its balancement capabilities with those of a reference model, following the comparison we made for the global approach [7].

In all simulations performed, 1024 vnodes were consecutively created and, after the creation of each vnode, the metric under analysis was measured. All the results presented are averages of 100 runs of the same test, in order to account for the random choice of a victim group.

4.1 Quality of the balancement

The most important parameters of the local approach are P_{min} and V_{min}. Basically, P_{min} defines the grain of the balancement inside each group, whereas V_{min} controls the size of groups (recall invariant $\mathcal{L}_2$). We thus have measured $\overline{\sigma}(Q_v, \overline{Q}_v)$ for several combinations of P_{min} and V_{min}.

We found that, when increasing both P_{min} and V_{min}: **a)** $\overline{\sigma}(Q_v, \overline{Q}_v)$ decreases, *i.e.*, the quality of the balancement improves; **b)** increasing P_{min} beyond the same value of V_{min} decreases $\overline{\sigma}(Q_v, \overline{Q}_v)$ by a very marginal amount; accordingly, we only present in figure 4 the results of the simulation when $P_{min} = V_{min}$.

The analysis of figure 4 reveals that the larger the value of V_{min} (meaning there will be few and big groups of vnodes), the more influence has the increasing of P_{min} in the improvement of the quality of the balancement.

In other words, when converging to the situation when there is one sole global group of vnodes, the dominant factor on $\overline{\sigma}(Q_v, \overline{Q}_v)$ becomes P_{min}. The reverse also holds: when V_{min} is small (implying many and small groups of vnodes), the effect of P_{min} in $\overline{\sigma}(Q_v, \overline{Q}_v)$ is very limited, whereas V_{min} is the dominant factor.

4.1.1 Zooming in

We may further observe in figure 4 two distinct zones for each of the curves. In the 1st zone, delimited by $1 \leq V \leq V_{max}$ (recall that $V_{max} = 2.V_{min}$), the evolution of $\overline{\sigma}(Q_v, \overline{Q}_v)$ matches the one under the global approach, for the same value of P_{min}. The reason for this behavior comes from the fact that when $1 \leq V \leq V_{max}$, there still is one sole group of vnodes (group 0), in which circumstances only P_{min} influences $\overline{\sigma}(Q_v, \overline{Q}_v)$.

In the 2nd zone, $V > V_{max}$, meaning that more groups of vnodes are created, degrading $\overline{\sigma}(Q_v, \overline{Q}_v)$. The slope of the curve and the amount on the increase of $\overline{\sigma}(Q_v, \overline{Q}_v)$ are larger for smaller values of V_{min}, that is, with many and small groups it is more difficult to achieve good values for the overall quality of balancement. After a sudden increase, $\overline{\sigma}(Q_v, \overline{Q}_v)$ remains relatively stable (this observation was confirmed by additional tests made with 8192 vnodes).

We have also uncovered a relation between the different values of $\overline{\sigma}(Q_v, \overline{Q}_v)$ in the 2nd zone: each time P_{min} and V_{min} double, $\overline{\sigma}(Q_v, \overline{Q}_v)$ decreases by nearly 30%.

4.1.2 Choice of P_{min} and V_{min}

Once set, P_{min} and V_{min} remain constant for the lifetime of a DHT. It is therefore important to make an informed choice of their values.

When the improvement of the quality of the balancement is the only objective, using the largest possible values for P_{min} and V_{min} is the obvious choice. There is, however, a tradeoff between the quality of the balancement and the storage/time consumed to achieve it: if V_{min} increases, there will be fewer, bigger groups of vnodes, with larger LPDR tables; the time consumed to sort a LPDR table will also grow with its number of records; finally, depending on the underlying protocol, bigger groups may require more synchronization time during local balancement events.

Establishing a value for V_{min} may be accomplished in a way that guarantees a good compromise between the quality of the balancement and the resources (storage/time) required. From the observations above, we may conclude that the amount of such resources is proportional to the value of V_{min}. Thus, ideally, we should be able to simultaneously lower the value of V_{min} and $\overline{\sigma}(Q_v, \overline{Q}_v)$. The value of V_{min} that achieves this objective is the one that minimizes any function directly proportional to both V_{min} and $\overline{\sigma}(Q_v, \overline{Q}_v)$.

We then define $\theta = \alpha \times [V_{min}/\max(V_{min})] + \beta \times [\overline{\sigma}(Q_v, \overline{Q}_v)/\max(\overline{\sigma}(Q_v, \overline{Q}_v))]$, where α and β are complementary weights ($\alpha + \beta = 1$) for the contributions of V_{min} and $\overline{\sigma}(Q_v, \overline{Q}_v)$, both normalized with respect to their maximum values. Figure 5 plots θ for $V_{min} \in \{8, 16, 32, 64, 128, \}$, when $\alpha = \beta = 0.5$.

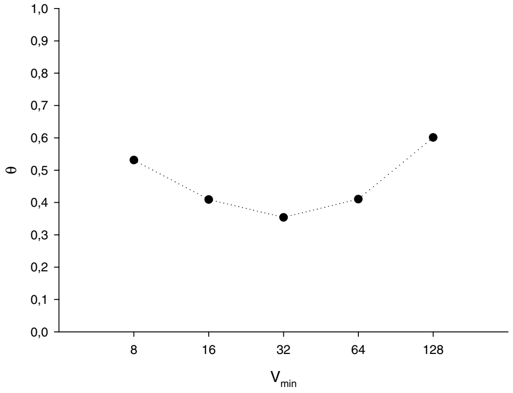

Figure 5. θ **for** $V_{min} \in \{8, 16, 32, 64, 128, \}$**.**

It may be observed that θ minimizes for $V_{min} = 32$. We thus have set $P_{min} = V_{min} = 32$ for the remaining tests of the simulation, except when otherwise specified.

4.2 Degradation of the quality of balancement

By following the local approach, we expected the quality of the balancement to be inferior to that obtained using the global approach: the LPDR of each group provides a partial view of the overall vnode set and so an optimal distribution of the DHT cannot be achieved. Such degradation of the quality of the balancement can be observed in figure 6, which shows $\overline{\sigma}(Q_v, \overline{Q}_v)$ when $P_{min} = 32$ and V_{min} varies.

When $V_{min} = 512$, there will be only one group (once $V_{max} = 1024$), and so the values of $\overline{\sigma}(Q_v, \overline{Q}_v)$ match those of the global approach. As V_{min} decreases, there will be more groups, with fewer vnodes, and $\overline{\sigma}(Q_v, \overline{Q}_v)$ degrades.

Nevertheless, we expected a smaller degradation of $\overline{\sigma}(Q_v, \overline{Q}_v)$, in the local approach: while $V \leq V_{max}$, group 0 will be the only group; the share/quota of R_h represented by the partitions bound to all the vnodes of group 0 will be 100%; when the $(V_{max} + 1)$'th vnode is created, the first group splits into groups 0 and 1, each one with a quota of 50%; therefore, both groups will share the same probability (50%) of receiving new vnodes, and so we expect both groups to become full and split at aproximatelly the same time; the same kind of reasoning may then be applied to the resulting four groups, and so on.

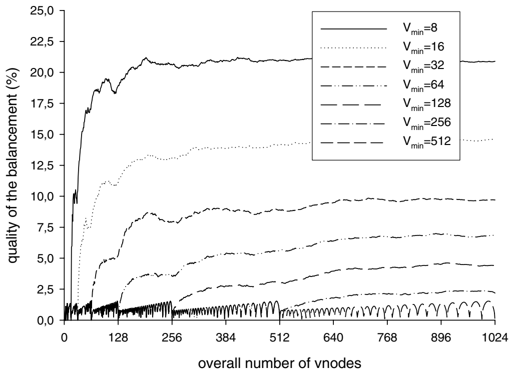

Figure 6. $\overline{\sigma}(Q_v, \overline{Q}_v)$ **when** $P_{min} = 32$**.**

Such synchrony in the creation of new groups should make each group to have a similar number of vnodes, a similar number of partitions per vnode and a similar size for partitions. Thus, the quota/share of all vnodes should be similar, regardless of the group where they belong. In turn, this should translate in small values for $\overline{\sigma}(Q_v, \overline{Q}_v)$.

4.2.1 Asynchrony in the creation of groups

In reality, however, the synchrony in the creation of groups is less than expected, as may be observed in figure 7, which plots the evolution of the ideal (G_{ideal}) and real (G_{real}) overall number of groups, when $P_{min} = V_{min} = 32$.

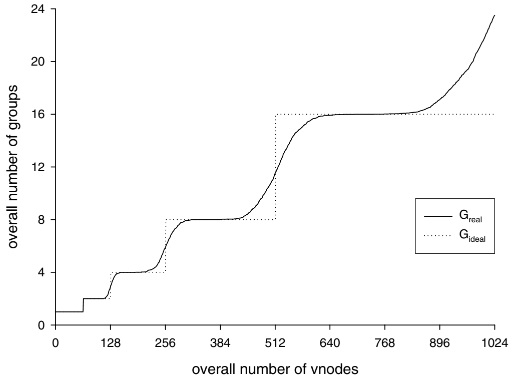

Figure 7. Evolution of the number of groups.

Ideally, the number of groups should double every time V (the overall number of vnodes) crosses a power of two boundary. However, as can be observed by following the evolution of G_{real}, there are premature and late creation of groups. Moreover, as V grows, the creation of new groups starts sooner and ends later.

0-7695-2132-0/04/$17.00 (C) 2004 IEEE

The lack of synchrony in the creation of groups also translates in the coexistence of groups with very different quotas of R_h. This may be deduced from the evolution of $\overline{\sigma}(Q_g, \overline{Q}_g)$, shown in figure 8, and registered during the same test in which G_{ideal} and G_{real} were collected.

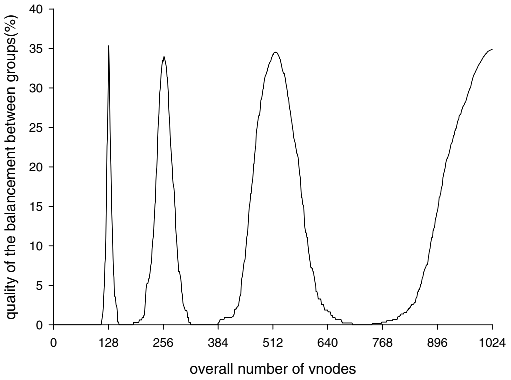

Figure 8. Evolution of $\overline{\sigma}(Q_g, \overline{Q}_g)$.

Informally, we may refer to $\overline{\sigma}(Q_g, \overline{Q}_g)$ as a metric that measures the quality of the balancement between groups. More precisely, $\overline{\sigma}(Q_g, \overline{Q}_g)$ is the relative standard deviation of the quotas of all groups, with relation to the ideal average quota, $\overline{Q}_g = 1/G$, where G is the real number of group. The quota of a group results from the sum of the quotas of all its vnodes: $Q_g = \sum_{v \in g}(Q_{v,g})$.

The correlation between the graphics of figures 7 and 8 is clear: whenever G_{ideal} and G_{real} diverge, groups with very different amounts of quota will coexist, which explains the spikes in the evolution of $\overline{\sigma}(Q_g, \overline{Q}_g)$.

4.2.2 Motives for the degradation

As we have seen, our optimistic predictions about the evolution of $\overline{\sigma}(Q_g, \overline{Q}_g)$ failed. The reason behind it is that we have not considered that, as the number of groups increase, their quota becomes smaller and so it gets more difficult to create the same number of newer vnodes in each group.

To understand the previous observation, suppose a certain number S of random shots are taken to a target divided in Z zones and each zone z represents a fraction Q_z of the target (*i.e.*, Q_z is the "theoretical" quota of z). Increasing S, when Z is fixed, makes the fraction of shots that hit any zone z (*i.e.*, the "real" quota of z) converge to Q_z. Reversely, increasing Z, when S is fixed, makes the real quota of any zone to diverge from its theoretical quota.

In the scenario of our model, increasing the number of vnodes/shots makes the number of groups/zones to also increase (once groups have limited capacity). Increasing the number of groups is enough to prevent a fair distribution of the number of vnodes among them. However, because the number of vnodes also increases, there will be contradictory effects which result on a sustained equilibrium on the values of $\overline{\sigma}(Q_v, \overline{Q}_v)$, as can be seen on figure 4.

Finally, the reason why $\overline{\sigma}(Q_v, \overline{Q}_v)$ have different values, accordingly with V_{min} (recall figures 4 and 6), should now also be clear: bigger values for V_{min} translate in fewer, bigger groups/zones, thus with bigger quotas; therefore, it is easier to ensure a more fair distribution of newer vnodes/shots, which explains lower values for $\overline{\sigma}(Q_v, \overline{Q}_v)$.

4.3 Comparison with Consistent Hashing

Our reference model is the Consistent Hashing (CH) approach [4], well known in the context of DHTs, and responsible for the introduction of the virtual server/node concept.

In CH, the hash table is divided in partitions, with random size, and each partition is bound to a virtual server. Each physical node may host more than one virtual server.

To ensure a fair distribution of the hash table, among a set of N homogeneous physical nodes, CH requires that each node receives at least $k.log_2N$ partitions/virtual servers. As shown in [3], CH may also be used accounting for node heterogeneity, by allocating to each node a different number of virtual servers.

In our model the virtual server/node (vnode) concept has a different meaning: a vnode is a set of partitions of the hash table, all of equal size, whose number fluctuates (though between strict bounds). Ultimately, it is this fluctuation, performed in a controlled manner, that allows to achieve good levels of balancement.

Thus, to compare our local approach with CH, it is convenient to abstract from each one's definition of virtual server/node. In order to do so, we define Q_n as the quota/fraction of R_h handled by each physical node n. Basically, Q_n results from dividing the sum of the ranges of all partitions hosted at node n, by 2^{B_h}.

We limit the comparison to the situation where physical nodes are homogeneous. For our local approach, we also assume only one vnode per snode; under such conditions, $\overline{\sigma}(Q_n, \overline{Q}_n) = \overline{\sigma}(Q_v, \overline{Q}_v)$ and thus we may compare the values of $\overline{\sigma}(Q_v, \overline{Q}_v)$ measured in the simulation of the local approach, with the values of $\overline{\sigma}(Q_n, \overline{Q}_n)$ from a specific simulation of the CH approach we have also performed.

Figure 9 shows the evolution of $\overline{\sigma}(Q_n, \overline{Q}_n)$, for the two approaches, as the number of (physical) nodes that join the DHT grows from 1 to 1024. For the local approach, $P_{min} = 32$ and V_{min} varies across $\{32, 64, 128, 256, 512\}$. The values for the CH approach are also averages of 100 runs of its simulation, because of the random size of partitions. Also, once the number of partitions per node is fixed in CH, but varies in our model from $P_{min} = 32$ to $P_{max} = 64$, we show the results for CH when considering both 32 and 64 partitions per node.

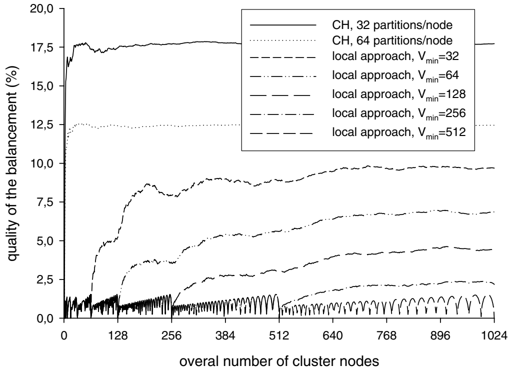

Figure 9. Evolution of $\overline{\sigma}(Q_n, \overline{Q}_n)$.

The results of the comparison with CH clarify the relative merits of the local approach. In effect, with regard to the quality of the balancement of the DHT, it is still able to show better values than the reference model. However, to achieve such improvement we have to carefully choose the value of V_{min}. This stresses, once again, the importance of the correct parameterization of the model.

5 Related work

Distributed Hash Tables have been the subject of intensive research, mainly in the area of Peer–to–Peer (P2P) systems [5], where lookup schemes usually have a DHT interface ([2] provides a recent survey). However, work that addresses the heterogeneity of nodes in such systems, and proposes load balancing mechanisms accordingly, is very recent [6, 1]. This may have to do with the fact that the balancement problem in such large scale, Internet–wide systems is intrinsically harder to solve.

Our model is cluster oriented, and thus operates at a very different scale (several orders of magnitude bellow, in the number of nodes) than that of the P2P systems. Moreover, it takes advantage of certain basic properties of clusters, namely: the lower rate of failures (which explains the absence of fault tolerance mechanisms in the model), and the short (typically one–hop) communication paths and high bandwidth (which make bearable events that may require synchronization between many nodes).

The balancement mechanisms proposed in [6] and [1] act continuously, and so they are able to handle situations where data distributions and/or accesses to data are non-uniform. In our model, we have assumed, until now, uniform data distributions in the DHT, and no hotspots in the access to data. As such, the level of dynamism in the balancement of the DHT is lower: the DHT is only re-balanced when changing the number of entities enrolled in it.

6 Conclusions

The work presented in this paper has extended a model for cluster oriented DHTs, previously presented [7], in order to increase its level of parallelism. The enhancements introduced will allow for a system based on the model to scale better and to react more promptly to changes in the number of nodes that participate in the DHT, or in their enrollment level. At the same time, our evaluation has demonstrated that the loss of quality on the balancement of the DHT, expected in result of the parallelization of certain operations, may be put under desired bounds, by a proper selection of the values for the main parameters of the model.

We intend to refine our model in several fronts. Increasing the scalability of coarse-grain balancement is fundamental in order to tackle more complex scenarios: *e.g.*, to maximize the cluster usage and, at the same time, minimize the response time, nodes may dedicate to several different user tasks, with variable resource demands during its lifetime; therefore, the balancement of a DHT should take into consideration its possible coexistence with other parallel/distributed applications running in the cluster. The mechanisms of the model for fine-grain balancement should also evolve, to deal with situations where access to data and/or storage utilization is non-uniform.

References

[1] K. Aberer, A. Datta, and M. Hauswirth. The Quest for Balancing Peer Load in Structured Peer-to-Peer Systems. Technical report, Distributed Information Systems Laboratory, Ecole Polytechnique Federale de Lausanne, 2003.

[2] H. Balakrishnan, M. Kaashoek, D. Karger, R. Morris, and I. Stoica. Looking Up Data in P2P Systems. *Communications of the ACM*, 46(2):43–48, 2003.

[3] F. Dabek, M. Kaashoek, D. Karger, and R. Morris. Wide-area Cooperative Storage with CFS. In *Proceedings of the ACM SOSP'01*, 2001.

[4] D. Karger, E. Lehman, F. Leighton, D. Levine, and R. Panigrahy. Consistent Hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, 1997.

[5] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-Peer Computing. Technical Report HPL-2002-57, HP Labs, 2002.

[6] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica. Load Balancing in Structured P2P Systems. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, 2003.

[7] J. Rufino, A. Pina, A. Alves, and J. Exposto. Toward a dynamically balanced cluster oriented DHT. In *Proceedings of the International Conference on Parallel and Distributed Computing and Networks (PDCN'04)*, 2004. To be presented.