Randomized Load Balancing by Joining and Splitting Bins

James Aspnes[*][†] Yitong Yin[*][‡]

September 15, 2008

Abstract

We study the following load balancing game: initially there is only one bin, which contains all the load; and at each time, either one of the bins is split into two bins, or two bins are joined into one bin. The join operation is defined as joining two uniformly random bins. Two types of split operations are considered: a weighted split, where the bin to split is sampled according to the probability distribution proportional to the weights of bins; and a non-weighted split, where the bin to split is uniformly sampled from all bins. We analyze the load factor of the bins, which is the ratio between the maximum load and the average load. We show that for weighted splits with uniform joins, the expected load factor of n bins is always $\Theta(\log n)$, which is independent of the sequence of joins and splits. For non-weighted splits, we show that the expected load factor after applying n non-weighted splits to one initial bin is between $\Omega(n^{0.5})$ and $O(n^{0.741})$. We study the performance of the mixing of joins and non-weighted splits, and show that the expected load factor approaches $O(n^{1/\sqrt{\frac{1}{2}\log_2 n}})$ after alternatively applying sufficiently many joins and non-weighted splits to an arbitrary initial load assignment of n bins.

---

[*] Department of Computer Science, Yale University.

[†] Supported in part by NSF grant CNS-0435201. Email: `aspnes@cs.yale.edu`.

[‡] Supported by a Kempner Foundation Fellowship and NSF grant CNS-0435201. Email: `yitong.yin@yale.edu`.

1 Introduction

Consider the following load balancing scenario: a certain amount of work load is distributed among a set of machines, while machines are joining and leaving the system. Upon an arrival of a new machine, one of the existing machines gives some of its load to the new machine; and upon a departure of a machine, it gives all its load away to one of the existing machines in the system.

This is a typical situation for many load balancing tasks, especially in distributed systems, where typically the availability of machines changes over time. The above scheme has the desirable property that with each arrival or departure of a machine, only one additional machine is bothered.

Such kind of load balancing schemes can be modeled as a simple game of joining and splitting of weighted bins. Each bin corresponds to a machine in the system, and the weight of the bin represents the amount of load assigned to the machine. The arrival of a new machine corresponds to a split of a bin; and the departure of an existing machine is represented by joining two bins. So the model for the load balancing becomes: initially there is only one bin, which contains all the load; and at each time either a join operation or a split operation is applied to the bins.

By switching the identities of the arriving or departing machine with a random machine, we can transform any adversarial sequence of insertions and deletions of machines to a sequence of randomized joinings and splittings of bins. For example, on the departure of machine v, we uniformly choose a machine u, let u and v switch load with each other, and let v give away its load to a uniformly random machine w. In this way, any adversarial departure of a machine is transformed into the following operation on bins:

> **Join:** Uniformly choose two bins and join them together.

For the arrivals of new machines, by sampling a random machine and letting it share its load with the new machine, we can deal with the joining of a machine as a splitting of a random bin. However, depending on the method of sampling, there are two natural splitting models:

> **Weighted split[1]:** Sample a random bin according to the probability distribution proportional to the weights of bins, and split the sampled bin into two bins.
>
> **Non-weighted split:** Sample a uniform bin, and split the sampled bin into two bins.

It is not hard to see that weighted splits are more expensive than non-weighted splits. Directly implementing weighted splits requires keeping track of the current loads of all machines. Distributed hash tables (DHTs) based on consistent hashing [5–10] implement the weighted splits by assuming that loads are uniformly distributed on a unit circle, and splits are implemented by applying uniform cuts to the unit circle.

In systems where it is infeasible to keep track of the loads of machines and it is impossible to make a mapping of loads to some uniform metric space, non-weighted split is the most natural way to deal with the arrivals of machines, because it depends only on uniformly sampling bins, which is either supported directly by most systems or can be approximated by random walks.

We measure the load balance of bins by the maximum weight. We call the ratio between the maximum weight and the average weight as the **load factor** of the bins. We would like to have

---

[1] This model has been previously named as uniform split, such as in [4], because it is equivalent to applying uniform cuts to a line interval. In this paper, we call this model as weighted split, because uniform split can also mean that the split bin is sampled uniformly, which is the case of non-weighted split.

the load factor depend only on the current number of bins, but not on the sequence of joins and splits applied to get these bins. We say that a model with such a property is **stable**.

The paper is organized as follows.

1. In Section 2, we show that the model of weighted splits and joins is stable. That is, for any sequence of weighted splits and joins, if there are n bins after applying the sequence of operations, the expected load factor is $\Theta(\log n)$.

2. In Section 3, we analyze the process of all non-weighted splits (with no joins), and show that the expected load factor after n splits is between $\Omega(n^{0.5})$ and $O(n^{0.741})$. The lower bound gives us a separation between weighted and non-weighted splits. Thus we see that the additional costs for the weighted splits are necessary.

3. In Section 4, we analyze a natural case of interspersed joins and non-weighted splits, in which the two operations alternate. That is, starting with an arbitrary assignments of n bins, at each time we first apply a non-weighted split and then apply a random join to the bins. This process is motivated by the work stealing model [2,3]. We show that the expected load factor for this sequence is $O(\exp(\sqrt{2\ln 2 \ln n})) = O(n^{1/\sqrt{\frac{1}{2}\log_2 n}})$ in the limit. Therefore, the model of joins and non-weighted splits is not stable.

4. Several open problems are discussed in the last section.

2 Weighted split

In this section, we deal with the case of weighted splits, where the bin to split is sampled with probability proportional to its current weight.

We formalize the process of joins and splits by a continuous model. We assume that the total weights of all bins is always 1. Let $X_0^{(t)}, X_1^{(t)}, \ldots, X_{n-1}^{(t)} \in [0,1]$ be random variables which represent the weights of the n bins after t operations. It holds for any t that $\sum_i X_i^{(t)} = 1$. Initially, $n = 1$ and $X_0^{(0)} = 1$ with probability 1. Assuming that at time t, there are n bins $X_0^{(t)}, X_1^{(t)}, \ldots, X_{n-1}^{(t)}$. The join and split operations are formally defined as such:

Join: $\{r, s\}$ is a uniformly random member of $\binom{[n]}{2}$, where $r < s$.

$$X_i^{(t+1)} = \begin{cases} X_r^{(t)} + X_s^{(t)} & \text{if } i = r, \\ X_{i+1}^{(t)} & \text{if } s \leq i \leq n-2, \\ X_i^{(t)} & \text{if } i < r \text{ or } r < i < s. \end{cases}$$

Split: $r \in [n]$ follows the probability distribution that $\Pr[r = i] = X_i^{(t)}$, and λ is a uniformly random value in $[0,1]$.

$$X_i^{(t+1)} = \begin{cases} X_i^{(t)} & \text{if } i < r, \\ \lambda X_r^{(t)} & \text{if } i = r, \\ (1-\lambda) X_r^{(t)} & \text{if } i = r+1, \\ X_{i-1}^{(t)} & \text{if } r+2 \leq i \leq n. \end{cases}$$

We can alternatively treat $\{X_i^{(t)}\}_i$ as disjoint intervals in $[0,1]$, and the split operation can be equivalently defined as uniformly sampling a point $p \in [0,1]$ and cutting the interval $X_r \ni p$ into two intervals at point p. This gives us the familiar process of uniformly cutting the line interval $[0,1]$. We denote by $Z_0, Z_1, \ldots, Z_{n-1}$ the lengths of the n intervals ended from applying $(n-1)$ uniform and independent cuts to the line interval $[0,1]$. The following lemma holds for this sequence.

Lemma 1 *The following properties hold for the sequence* $Z_0, Z_1, \ldots, Z_{n-1}$*.*

1. *Let* r *be a uniformly random element of* $[n-1]$*. The distribution of the sequence* $Z_0, Z_1, \ldots, Z_r + Z_{r+1}, \ldots, Z_{n-1}$ *is the same as the distribution of* $Z_0, Z_1, \ldots, Z_{n-2}$
2. *The distribution of* $Z_0, Z_1, \ldots, Z_{n-1}$ *is invariant under permeation.*

Proof: It is easy to see that joining two neighbor intervals is equivalent to undoing a random cut. Since all cuts are independent, the distribution is the same as applying $(n-2)$ cuts. The first property holds.

Let C_k denote the kth cut in the interval $[0,1]$. It is easy to see that the probability measure of $(Z_0, Z_1, \ldots, Z_{n-1})$ is determined by the set $\{C_1, C_2, \ldots, C_{n-1}\}$. Permuting Z_i does not change the set $\{C_1, C_2, \ldots, C_{n-1}\}$, therefore the distribution of $Z_0, Z_1, \ldots, Z_{n-1}$ is invariant under permeation. ∎

With the above lemma, we can show that the distribution of the weights of bins formed by applying weighted splits and uniformly random joins are the same as the distribution of intervals by applying uniform and independent cuts.

Lemma 2 *For any* t*, if after* t *operations there are* n *bins, the distribution of* $X_0^{(t)}, X_1^{(t)}, \ldots, X_{n-1}^{(t)}$ *is the same as the distribution of* $Z_0, Z_1, \ldots, Z_{n-1}$*.*

Proof: We prove this by induction on t.

Induction Hypothesis: Assuming that after t operations, there are n bins. The distribution of $X_0^{(t)}, X_1^{(t)}, \ldots, X_{n-1}^{(t)}$ is the same as the distribution of $Z_0, Z_1, \ldots, Z_{n-1}$.

Induction Step: If the next operation is a split, it is trivial to see that the induction hypothesis is true.

If the next operation is a join, $X_0^{(t+1)}, X_1^{(t+1)}, \ldots, X_{n-2}^{(t+1)}$ can be formed by first applying a random permutation ρ to $X_0^{(t)}, X_1^{(t)}, \ldots, X_{n-1}^{(t)}$, joining a uniformly random bin to its neighbor, and permuting the rest bins back to the original order. According to Lemma 1, it is easy to see that the distribution of $X_0^{(t+1)}, X_1^{(t+1)}, \ldots, X_{n-2}^{(t+1)}$ is the same as the distribution of $Z_0, Z_1, \ldots, Z_{n-2}$. ∎

It is well known [7] that $\mathrm{E}\left[\max_{i\in[n]} Z_i\right] = \Theta(\frac{\log n}{n})$. Therefore the following theorem holds for the join and split model with weighted splits.

Theorem 3 *For any sequence of joins and weighted splits, if there are* n *bins after applying the sequence, the expected maximum load of bins is* $\Theta(\frac{\log n}{n})$*, i.e. the expected load factor is* $\Theta(\log n)$*.*

3 Non-weighted split

In this section, we analyze the performance of non-weighted splits, where the bin to split is sampled uniformly from all current bins. We first consider the split-only process. The effect of joins is studied in the next section.

Initially there is one bin with weight 1. For the split-only process, after n operations, there are $(n+1)$ bins. Let $\{X_i\}_{i\in[n]}$ be the weights of bins at time $(n-1)$. The bins at time n is inductively defined as such:

Uniformly choose a bin r from $[n]$.

$$X_i = \begin{cases} X_i & \text{if } i < r, \\ \frac{1}{2}X_r & \text{if } i = r \text{ or } i = r+1, \\ X_{i-1} & \text{if } r+2 \le i \le n. \end{cases}$$

3.1 A random tree model

The above process can be modeled as the evolution of a random binary tree. Initially there is a root node. At time n where $n = 1, 2, \ldots$, there are n leaves of the tree. Each leaf corresponds to a bin. To perform a split, we choose a uniform leaf node and add two children to that leaf node.

For each leaf node v, let $d(v)$ be the depth of v, i.e. the distance to the root. Let B_v denote the bin corresponding to the leaf v. It is easy to see that $d(v)$ is the number of splits applied to the bin B_v, therefore the weight of B_v is $2^{-d(v)}$.

Hence the expectation of the maximum weight is the expected weight of the shallowest leaf. Unlike the height of the tree, which is the largest depths of leaves, we are looking for the smallest depth of the leaves. A usual tool for analyzing the parameter of random trees is fringe analysis [1], which is more suitable for the average rather than worst-case parameter of leaves. Our analysis of the split-only process also gives a bound on the smallest depth of the leaves, which can not be achieved by fringe analysis.

3.2 Lower bound

Note that the maximum load of bins decreases by half once the last heaviest bin is split, and that split creates two new heaviest bins. The maximum load will not further decrease until both these two bins have been split. This gives us a lower bound estimation of the maximum load, which is formally stated by the following theorem.

Theorem 4 *Let $X_0, X_1, \ldots, X_n$ be the weights of $(n+1)$ bins after n non-weighted splits. It holds that*

$$\mathrm{E}\left[\max_{0\le i\le n} X_i\right] \ge \frac{1}{\sqrt{2n}}.$$

Proof: Let $W^{(n)}$ be a random variable defined as such. Consider a game where a player keeps collecting coupons of two types in each rounds. Initially, $W^{(1)} = \frac{1}{2}$ with probability 1. At round $n = 2, 3, \ldots$, with probability $(1 - \frac{2}{n})$, the player gets nothing, with probability $\frac{1}{n}$ she gets coupon 1, and with probability $\frac{1}{n}$ she gets coupon 2. Once she has collected coupons of both types, she discards all the coupons she has and has $W^{(n)} = \frac{1}{2}W^{(n-1)}$, in other cases, $W^{(n)} = W^{(n-1)}$.

It is easy to see that $W^{(n)}$ gives a lower bound of the maximum load of $(n+1)$ bins after n splits. In fact, we can alternatively define $W^{(n)}$ by coupling with the weight of one of the bins. The "coupons" are marked bins. At each round there are either one or two bins marked. If a marked bin is split then the mark on it is removed, and if the bin which is split is the only marked bin, then the two new bins created by the split are marked. After n splits, $W^{(n)}$ is the weight of the current marked bin(s) (note that if there are two marked bins, they have the same weights). Since $W^{(n)} = X_i$ for some i, it holds that $W^{(n)} \le \max_i X_i$.

Let $w(n) = \mathrm{E}\left[W^{(n)}\right]$. Let $w_b(n)$ be the part of $w(n)$ contributed by the cases that there are b marked bin(s), where $b \in \{1,2\}$. Formally, let $M^{(n)}$ be the set of marked bins after n splits, and $w_b(n) = \sum_v v \cdot \Pr[W^{(n)} = v \wedge |M^{(n)}| = b]$. By total probability, $w(n) = w_1(n) + w_2(n)$. Let r be a uniformly random bin from $[n]$, by definition of $W^{(n)}$, we have the following equations.

$$\begin{aligned}
w_1(n) &= \sum_v v \cdot \Pr[W^{(n)} = v \wedge |M^{(n)}| = 1] \\
&= \sum_v v \cdot \Pr[(W^{(n-1)} = v \wedge |M^{(n-1)}| = 1 \wedge r \notin M^{(n-1)}) \\
&\qquad\qquad \vee (W^{(n-1)} = v \wedge |M^{(n-1)}| = 2 \wedge r \in M^{(n-1)})] \\
&= \sum_v v \left(1 - \frac{1}{n}\right) \Pr[W^{(n-1)} = v \wedge |M^{(n-1)}| = 1] \\
&\quad + \sum_v v \cdot \frac{2}{n} \Pr[W^{(n-1)} = v \wedge |M^{(n-1)}| = 2] \\
&= \left(1 - \frac{1}{n}\right) w_1(n-1) + \frac{2}{n} w_2(n-1);
\end{aligned}$$

and

$$\begin{aligned}
w_2(n) &= \sum_v v \cdot \Pr[W^{(n)} = v \wedge |M^{(n)}| = 2] \\
&= \sum_v v \cdot \Pr[(W^{(n-1)} = 2v \wedge |M^{(n-1)}| = 1 \wedge r \in M^{(n-1)}) \\
&\qquad\qquad \vee (W^{(n-1)} = v \wedge |M^{(n-1)}| = 2 \wedge r \notin M^{(n-1)})] \\
&= \sum_v \frac{v}{2} \cdot \frac{1}{n} \Pr[W^{(n-1)} = v \wedge |M^{(n-1)}| = 1] \\
&\quad + \sum_v v \left(1 - \frac{2}{n}\right) \Pr[W^{(n-1)} = v \wedge |M^{(n-1)}| = 2] \\
&= \frac{1}{2n} w_1(n-1) + \left(1 - \frac{2}{n}\right) w_2(n-1).
\end{aligned}$$

Summing them up, we get $w(n)$.

$$\begin{aligned}
w(n) &= w_1(n) + w_2(n) \\
&= \left(1 - \frac{1}{2n}\right) w_1(n-1) + w_2(n-1) \\
&\geq \left(1 - \frac{1}{2n}\right) (w_1(n-1) + w_2(n-1)) \\
&= \left(1 - \frac{1}{2n}\right) w(n-1) \\
&= \frac{1}{2} \prod_{k=2}^{n} \left(1 - \frac{1}{2k}\right) \\
&\geq \frac{1}{\sqrt{2n}}.
\end{aligned}$$

Therefore, $\mathrm{E}\left[\max_i X_i\right] \geq w(n) \geq \frac{1}{\sqrt{2n}}$. ∎

Recall that for the random binary tree defined in Section 3.1, the least depths of the leaves $\min_v d(v) = -\log_2(\max_i X_i)$. The logarithmic function is concave, thus according to Jensen's inequality, $\mathrm{E}\left[\min_v d(v)\right] = -\mathrm{E}\left[\log_2(\max_i X_i)\right] \leq -\log_2 \mathrm{E}\left[\max_i X_i\right] \leq \frac{1}{2}(\log_2 n + 1)$. This gives us an upper bound on the expected depth of the shallowest leaf in the random binary trees incurred by inserting two children to a uniform leaf.

3.3 Upper bound

Let X be the vector of the weights of bins. Let $|X|_p$ be the ℓ_p-norm of X. The maximum weight can be therefore represented as $|X|_\infty$. It is well known that $|X|_\infty \leq |X|_p$ for any $p \geq 1$. This gives us an upper bound on the maximum weight. The following theorem is developed using this idea.

Theorem 5 *Let $X_0, X_1, \ldots, X_n$ be the weights of $(n+1)$ bins after n non-weighted splits. For any $\alpha \geq 1$, it holds that*

$$\mathrm{E}\left[\max_{0 \leq i \leq n} X_i\right] = O\left(n^{-\left(1-2^{(1-\alpha)}\right)/\alpha}\right).$$

Proof: Let $w_n = \mathrm{E}\left[\sum_{i=0}^{n} X_i^\alpha\right]$. Let $Y_0, Y_1, \ldots, Y_{n-1}$ be the weights of n bins after $(n-1)$ non-weighted splits, and r is a random bin sampled uniformly from $[n]$. By definition of the non-weighted split, we have that

$$\begin{aligned}
\sum_{i=0}^{n} X_i^\alpha &= 2\left(\frac{Y_r}{2}\right)^\alpha + \sum_{i \neq r} Y_i^\alpha \\
&= (2^{(1-\alpha)} - 1) Y_r^\alpha + \sum_{i=0}^{n-1} Y_i^\alpha.
\end{aligned}$$

By total probability,

$$\begin{aligned}
w_n &= \sum_{i=0}^{n-1} \Pr_{r\in[n]}[r=i]\cdot \mathrm{E}\left[\sum_{j=0}^{n} X_j^\alpha \mid r=i\right] \\
&= \sum_{i=0}^{n-1} \frac{1}{n}\cdot \mathrm{E}\left[\left(2^{(1-\alpha)}-1\right)Y_i^\alpha + \sum_{j=0}^{n-1} Y_j^\alpha\right] \\
&= \frac{1}{n}\left(2^{(1-\alpha)}-1\right)\sum_{i=0}^{n-1}\mathrm{E}\left[Y_i^\alpha\right] + \mathrm{E}\left[\sum_{j=0}^{n-1} Y_j^\alpha\right] \\
&= \left(1-\frac{1}{n}\left(1-2^{(1-\alpha)}\right)\right) w_{(n-1)}. \qquad (1)
\end{aligned}$$

And obviously $w_0 = 1$. Therefore,

$$\begin{aligned}
w_n &= \prod_{k=1}^{n}\left(1-\frac{1}{k}\left(1-2^{(1-\alpha)}\right)\right) \\
&= \exp\left(\sum_{k=1}^{n}\ln\left(1-\frac{1}{k}\left(1-2^{(1-\alpha)}\right)\right)\right) \\
&= \exp\left(O(1)-\left(1-2^{(1-\alpha)}\right)\sum_{k=1}^{n}\frac{1}{k}\right) \\
&= \exp\left(O(1)-\left(1-2^{(1-\alpha)}\right)\ln n\right) \\
&= O(n^{-\left(1-2^{(1-\alpha)}\right)}).
\end{aligned}$$

For $\alpha \geq 1$, the function $f(x) = x^{1/\alpha}$ is concave. According to Jensen's inequality,

$$\begin{aligned}
\mathrm{E}\left[\max_{0\le i\le n} X_i\right] &\leq \mathrm{E}\left[\left(\sum_{i=0}^{n} X_i^\alpha\right)^{1/\alpha}\right] \\
&\leq \left(\mathrm{E}\left[\sum_{i=0}^{n} X_i^\alpha\right]\right)^{1/\alpha} \\
&= O(n^{-\left(1-2^{(1-\alpha)}\right)/\alpha}).
\end{aligned}$$

∎

With a standard numerical routine, we minimize the above bound. By setting $\alpha = 2.4$, the expected maximum load is $O(n^{-0.259})$ and the corresponding expected load factor is $O(n^{0.741})$. Combining with Theorem 4, we have the following corollary.

Corollary 6 *Let σ be the expected load factor for the $(n+1)$ bins after applying n non-weighted splits to an initial bin. It holds that*

$$\Omega(n^{0.5}) \leq \sigma \leq O(n^{0.741}).$$

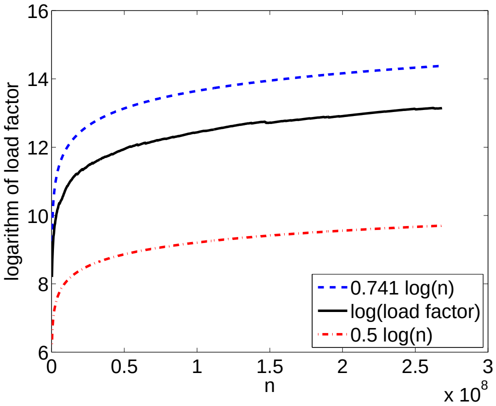

Figure 1: A comparison between the logarithms of the simulated load factor and the proved upper and lower bounds.

Figure 1 shows the curve of the logarithm of the load factor simulated on up to 2^{28} bins. Comparing it with the upper and lower bounds proved above, we can see that the real bound on load factor is closer to the upper bound.

4 Alternation of joins and non-weighted splits

It is a bit surprising so see that joins can actually optimize the load factor. In this section we study a natural case of the mixing of joins and non-weighted splits: alternatively applying split and join to n bins. We will see that this sequence yields an $n^{o(1)}$ expected load factor, which is much better than the case of just non-weighted splits.

We assume that initially there are $(n-1)$ bins $\{X_i^{(0)}\}_{i\in[n-1]}$ following some distribution. At time t, where the bins are $\{X_i^{(t)}\}_{i\in[n-1]}$, we first apply a non-weighted split and the bins become $\{Y_i^{(t+1)}\}_{i\in[n]}$; and then apply a join and the bins become $\{X_i^{(t+1)}\}_{i\in[n-1]}$. The transition is formally defined as such:

Uniformly choose a bin u from $[n-1]$.

$$Y_i^{(t+1)} = \begin{cases} X_i^{(t)} & \text{if } i < u, \\ \frac{1}{2}X_u^{(t)} & \text{if } i = u \text{ or } i = u+1, \\ X_{i-1}^{(t)} & \text{if } u+2 \le i \le n-1. \end{cases}$$

Uniformly choose two bins $\{r, s\}$ from $\binom{[n]}{2}$, where $r < s$.

$$X_i^{(t+1)} = \begin{cases} Y_r^{(t+1)} + Y_s^{(t+1)} & \text{if } i = r, \\ Y_{i+1}^{(t+1)} & \text{if } s \le i \le n-2, \\ Y_i^{(t+1)} & \text{if } i < r \text{ or } r < i < s. \end{cases}$$

To prove an upper bound on the maximum load, we have the following lemma.

Lemma 7 *Let $x \in [0,1]^n$ be a vector such that $\sum_{i=1}^n x_i = 1$. For any integers k, l such that $k > l > 0$, it holds that*

$$\sum_{i=1}^n x_i^{k-l} \cdot \sum_{j=1}^n x_j^l \;\; \leq \;\; \sum_{i=1}^n x_i^{k-1}.$$

Proof: For any x_i and x_j,

$$\left(x_i^{k-l}x_j^l + x_i^l x_j^{k-l}\right) - \left(x_i^{k-1}x_j + x_i x_j^{k-1}\right) = x_i x_j \left(x_j^{k-l-1} - x_i^{k-l-1}\right)\left(x_i^{l-1} - x_j^{l-1}\right) \leq 0.$$

Therefore,

$$\begin{aligned} \sum_{i=1}^n x_i^{k-l} \cdot \sum_{j=1}^n x_j^l &= \sum_{i=1}^n x_i^k + \sum_{i=1}^{n-1}\sum_{j=i+1}^n \left(x_i^{k-l}x_j^l + x_i^l x_j^{k-l}\right) \\ &\leq \sum_{i=1}^n x_i^k + \sum_{i=1}^{n-1}\sum_{j=i+1}^n \left(x_i^{k-1}x_j + x_i x_j^{k-1}\right) \\ &= \sum_{i=1}^n x_i^{k-1} \cdot \sum_{j=1}^n x_j \\ &= \sum_{i=1}^n x_i^{k-1} \end{aligned}$$

∎

With this lemma, we can prove an upper bound on the maximum load after applying sufficiently many alternative joins and splits. The theorem is proved with a norm-based technique which is similar to Theorem 5.

Theorem 8 *The expected load factor of bins after applying alternative joins and non-weighted splits approaches $O(\exp(\sqrt{2\ln 2 \ln n}))$ in the limit. That is, for sufficiently large n, with arbitrary distribution of $X^{(0)}$ over the space that $|X^{(0)}|_1 = 1$, it holds that*

$$\lim_{t\to\infty} \mathrm{E}\left[\max_i X_i^{(t)}\right] = O\left(\exp\left(\sqrt{2\ln 2\ln n}\right)\cdot n^{-1}\right).$$

Proof: We write $X_i = X_i^{(t)}$ and $Y_i = Y_i^{(t)}$ if no ambiguity is introduced. Let $k > 1$ be an integer. Define that

$$\begin{aligned} w_k^{(t)} &= \mathrm{E}\left[\left(|X^{(t)}|_k\right)^k\right] = \mathrm{E}\left[\sum_{i=0}^{n-1} X_i^k\right]; \\ u_k^{(t)} &= \mathrm{E}\left[\left(|Y^{(t)}|_k\right)^k\right] = \mathrm{E}\left[\sum_{i=0}^{n-2} Y_i^k\right]. \end{aligned}$$

By definition of the process, $X^{(t)}$ is formed by joining two uniformly random bins in $Y^{(t)}$. Assuming that the two sampled bins are $\{r,s\} \in \binom{[n]}{2}$, it holds that

$$\begin{aligned}\sum_{i=0}^{n-1} X_i^k &= (Y_r+Y_s)^k + \sum_{i\notin\{r,s\}} Y_i^k \\ &= \sum_{i=0}^{n-1} Y_i^k + \sum_{l=1}^{k-1}\binom{k}{l} Y_r^{k-l} Y_s^l. \qquad (2)\end{aligned}$$

By total probability, we have that for the join operation, the following recursive relation hold between $w_k^{(t)}$ and u_{k-1}^t.

$$\begin{aligned}w_k^{(t)} &= \sum_{\{r,s\}\in\binom{[n]}{2}} \Pr\left[\{r,s\}\right]\cdot \mathrm{E}\left[\sum_{i=0}^{n-1} X_i^k \mid \{r,s\}\right] \\ &= \sum_{\{r,s\}\in\binom{[n]}{2}} \frac{1}{\binom{n}{2}}\cdot \mathrm{E}\left[\sum_{i=0}^{n-1} Y_i^k + \sum_{l=1}^{k-1}\binom{k}{l} Y_r^{k-l}Y_s^l\right] \qquad (3) \\ &= \mathrm{E}\left[\sum_{i=0}^{n-1} Y_i^k\right] + \frac{\sum_{l=1}^{k-1}\binom{k}{l}}{2\binom{n}{2}}\cdot \mathrm{E}\left[\sum_{r=0}^{n-1}\sum_{s\neq r} Y_r^{k-l}Y_s^l\right] \\ &= u_k^{(t)} + \frac{\sum_{l=1}^{k-1}\binom{k}{l}}{2\binom{n}{2}}\cdot\left(\mathrm{E}\left[\sum_{r=0}^{n-1}\sum_{s=0}^{n-1} Y_r^{k-l}Y_s^l\right] - \mathrm{E}\left[\sum_{r=0}^{n-1} Y_r^k\right]\right) \\ &= \left(1-\frac{2^{k-1}-1}{\binom{n}{2}}\right)u_k^{(t)} + \frac{\sum_{l=1}^{k-1}\binom{k}{l}}{2\binom{n}{2}}\cdot \mathrm{E}\left[\sum_{r=0}^{n-1} Y_r^{k-l}\cdot\sum_{s=0}^{n-1} Y_s^l\right] \\ &\leq \left(1-\frac{2^{k-1}-1}{\binom{n}{2}}\right)u_k^{(t)} + \frac{\sum_{l=1}^{k-1}\binom{k}{l}}{2\binom{n}{2}}\cdot \mathrm{E}\left[\sum_{r=0}^{n-1} Y_r^{k-1}\right] \qquad (4) \\ &= \left(1-\frac{2^{k-1}-1}{\binom{n}{2}}\right)u_k^{(t)} + \frac{2^{k-1}-1}{\binom{n}{2}}u_{k-1}^{(t)}, \qquad (5)\end{aligned}$$

where (3) is due to (2), and (4) is due to Lemma 7.

By the same argument as (1) in Theorem 5, it holds for the split operation that

$$u_k^{(t)} = \left(1-\frac{1}{n}\left(1-2^{(1-k)}\right)\right)w_k^{(t-1)}. \qquad (6)$$

Combining (5) and (6), we have the recursion that

$$\begin{aligned}w_k^{(t)} &\leq \left(1-\frac{2^{k-1}-1}{\binom{n}{2}}\right)\left(1-\frac{1}{n}\left(1-2^{(1-k)}\right)\right)w_k^{(t-1)} \\ &\quad +\frac{2^{k-1}-1}{\binom{n}{2}}\left(1-\frac{1}{n}\left(1-2^{(2-k)}\right)\right)w_{k-1}^{(t-1)} \\ &\leq \left(1-\frac{1}{2n}\right)w_k^{(t-1)} + \frac{2^k}{n(n-1)}w_{k-1}^{(t-1)}.\end{aligned}$$

Assuming that $k = o(\log n)$, by induction on k it is routine to show that for all sufficient large n, as t goes to infinity, $\sup\{w_k^{(t)}\}$ converges to some value w_k, such that

$$w_k \leq \left(1 - \frac{1}{2n}\right) w_k + \frac{2^k}{n(n-1)} w_{k-1}.$$

Therefore, $w_k \leq \frac{2^{k+1}}{n-1} w_{k-1}$. It is trivial to see that $w_1 = \mathrm{E}\left[\sum_i X_i\right] = 1$. By induction on k, we have that

$$w_k \leq \frac{2^{(k^2+3k-4)/2}}{(n-1)^{k-1}}.$$

According to Jensen's inequality, it holds that

$$\begin{aligned}
\lim_{t\to\infty} \mathrm{E}\left[\max_i X_i^{(t)}\right] &\leq \lim_{t\to\infty} \mathrm{E}\left[|X^{(t)}|_k\right] \\
&\leq \lim_{t\to\infty} \left(\mathrm{E}\left[\left(|X^{(t)}|_k\right)^k\right]\right)^{1/k} \\
&\leq (w_k)^{1/k} \\
&= O\left(2^{\frac{k}{2}} n^{-(1-\frac{1}{k})}\right).
\end{aligned}$$

By setting $k = \lfloor \sqrt{2 \log_2 n} \rfloor$, we have that

$$\lim_{t\to\infty} \mathrm{E}\left[\max_i X_i^{(t)}\right] = O\left(\exp\left(\sqrt{2 \ln 2 \ln n}\right) \cdot n^{-1}\right).$$

∎

5 Open problems

One interesting open problem is to tighten the upper and lower bounds of the load factor for the non-weighted split only process, since this is a natural random process and is also related to an interesting parameter of a random tree.

Another important problem is to have more general bounds for the mode of non-weighted splits with joins, which may depend on the certain parameters of the sequence of joins and splits. It is also important to analyze the convergence rate of the sequence of joins and splits.

References

[1] R. Baeza-Yates. Fringe analysis revisited. *ACM Computing Surveys (CSUR)*, 27(1):109–119, 1995.

[2] R. Blumofe, C. Joerg, B. Kuszmaul, C. Leiserson, K. Randall, and Y. Zhou. Cilk: an efficient multithreaded runtime system. *Proceedings of the fifth ACM SIGPLAN symposium on Principles and practice of parallel programming*, pages 207–216, 1995.

[3] R. Blumofe and C. Leiserson. Scheduling multithreaded computations by work stealing. *Journal of the ACM (JACM)*, 46(5):720–748, 1999.

[4] J. Cichon, M. Klonowski, L. Krzywiecki, B. Rozanski, and P. Zielinski. Random Subsets of the Interval and P2P Protocols. *Lecture Notes in Computer Science*, 4627:409, 2007.

[5] K. Hildrum, J. Kubiatowicz, S. Rao, and B. Y. Zhao. Distributed object location in a dynamic network. In *Proceedings of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA 2002)*, pages 41–52, 2002.

[6] M. F. Kaashoek and D. R. Karger. Koorde: A simple degree-optimal distributed hash table. In *The 2nd International Workshop on Peer-to-Peer Systems (IPTPS 2003)*, pages 98–107, 2003.

[7] D. R. Karger, E. Lehman, F. T. Leighton, R. Panigrahy, M. S. Levine, and D. Lewin. 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 the Theory of Computing (STOC 1997)*, pages 654–663, 1997.

[8] M. Naor and U. Wieder. Novel architectures for p2p applications: the continuous-discrete approach. In *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms (SPAA 2003)*, pages 50–59, 2003.

[9] A. I. T. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware 2001, IFIP/ACM International Conference on Distributed Systems Platforms*, pages 329–350, 2001.

[10] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of ACM SIGCOMM 2001*, pages 149–160, 2001.