[ceph-users] trying to understanding crush more deeply

Maged Mokhtar mmokhtar at petasan.org
Fri Sep 22 10:01:26 PDT 2017

- Previous message (by thread): [ceph-users] trying to understanding crush more deeply
- Next message (by thread): [ceph-users] erasure code profile
- **Messages sorted by:** [date] [thread] [subject] [author]

```
If you have a random number generator rand() and variables A,B

A = rand()
B = rand()
and loop 100 times to see which is bigger A or B, on average A will win
50 times and B wins 50 times
Now assume you want to make A win twice as many times, you can add a
weight
A = 3 x rand()
B = 1 x rand()
If you loop 100 times, on average A will win 75 times and B wins 25
times

Hashing is like a random function but takes (in case of Jenkins) integer
inputs, the output is a random distribution but is repeatable if you
pass the same integer values..hence it is called pseudo random.

In code:

straw
draw = crush_hash32_3(bucket->h.hash, x, bucket->h.items[i], r);
draw &= 0xffff;
draw *= bucket->straws[i];

straw2
u = crush_hash32_3(bucket->h.hash, x, ids[i], r);
u &= 0xffff;
ln = crush_ln(u) - 0x1000000000000ll;

/*
* divide by 16.16 fixed-point weight. note
* that the ln value is negative, so a larger
* weight means a larger (less negative) value
* for draw.
*/
draw = div64_s64(ln, weights[i]);

In both straw and straw 2, we compute the hash based on pg number,
replica count, bucket id:
for straw: multiply hash value by weight (or function that depends on
weight)
for straw2: create a -ve number based on ln of hash value then divide by
weight (or function that depends on weight), as per comment in code we
divide rather than multiply since the value is negative.

In both cases the bucket with the highest value wins the PG

On 2017-09-22 18:05, Will Zhao wrote:
```

```
> Thanks !   I still have a question. Like the code in bucket_straw2_choose below:
>
> u = crush_hash32_3(bucket->h.hash, x, ids[i], r);
> u &= 0xffff;
> ln = crush_ln(u) - 0x1000000000000ll;
> draw = div64_s64(ln, weights[i]);
>
> Because the x , id, r , don't change, so the ln won't change for old
> bucket, add osd or remove osd only change the weight.  Suppose for
> pgi, there are 3 bucket(host) with weight 3w, add one host with weight
> w, there are 4 buckets with weight now.  This means the movement will
> depend on  ln value , am I understand right ? I don't understand how
> this make sure the new bucket get desirable pgs ?  I read
> https://en.wikipedia.org/wiki/Jenkins_hash_function and
> http://en.wikipedia.org/wiki/Exponential_distribution,  But I can't
> link them  together to understand this ? Can you explain something
> about this ?  Apologize for my dummy. And thank you very much   .   : )
>
> On Fri, Sep 22, 2017 at 3:50 PM, Maged Mokhtar <mmokhtar at petasan.org> wrote:
>
>> Per section 3.4.4 The default bucket type straw computes the hash of (PG
>> number, replica number, bucket id) for all buckets using the Jenkins integer
>> hashing function, then multiply this by bucket weight (for OSD disks the
>> weight of 1 is for 1 TB, for higher level it is the sum of contained
>> weights). The selection function chooses the bucket/disk with the max value:
>> c(r,x) = maxi ( f (wi)hash(x, r, i))
>>
>> So if you add a OSD disk, there is a new disk id that enters this
>> competition and will get PG from other OSDs proportional to its weight,
>> which is a desirable effect, but a side effect is that the weight hierarchy
>> has slightly changed so now some older buckets may win PGs from other older
>> buckets according to the hash function.
>>
>> So straw does have overhead when adding (rather than replacing), it does not
>> do minimal PG re-assignments. But it terms of overall efficiency of
>> adding/removing of buckets at end and in middle of hierarchy it is the best
>> overall over other algorithms as seen on chart 5 and table 2.
>>
>> On 2017-09-22 08:36, Will Zhao wrote:
>>
>> Hi Sage  and all :
>> I am tring to understand cursh more deeply. I have tried to read
>> the code and paper, and search the mail list archives ,  but I still
>> have some questions and can't understand it well.
>> If I have 100 osds, and when I add a osd ,  the osdmap changes,
>> and how the pg is recaulated to make sure the data movement is
>> minimal.  I tried to use crushtool --show-mappings --num-rep 3  --test
>> -i map , through changing the map for 100osds and 101 osds , to look
>> the result , it looks like the pgmap changed a lot .  Shouldn't the
>> remap  only happen to some of the pgs ? Or crush from adding  a pg is
>> different from a new osdmap ? I konw I must understand something
>> wrong. I appreciate if you can explain more about the logic of adding
>> a osd . Or is there  more doc that I can read ? Thank you very much
>> !!! : )
>> _______________________________________________
>> ceph-users mailing list
>> ceph-users at lists.ceph.com
>> http://lists.ceph.com/listinfo.cgi/ceph-users-ceph.com
-------------- next part --------------
An HTML attachment was scrubbed...
URL: <http://lists.ceph.com/pipermail/ceph-users-
ceph.com/attachments/20170922/ff7e1a83/attachment.html>
```

- Previous message (by thread): [ceph-users] trying to understanding crush more deeply

- Next message (by thread): [ceph-users] erasure code profile
- **Messages sorted by:** [date] [thread] [subject] [author]

More information about the ceph-users mailing list