Re: crush: straw is dead, long live straw2

[Date Prev][Date Next][Thread Prev][Thread Next][Date Index][Thread Index]

- *Subject*: Re: crush: straw is dead, long live straw2
- *From*: Sage Weil <sweil@xxxxxxxxxx>
- *Date*: Fri, 12 Dec 2014 08:20:36 -0800 (PST)
- *Cc*: Thorsten Behrens <tbehrens@xxxxxxxx>, ceph-devel@xxxxxxxxxxxxxxx
- *In-reply-to*: <548B0C16.4000509@gmail.com>
- *References*: <alpine.DEB.2.00.1412081516010.18281@cobra.newdream.net> <20141212091408.GN4150@thinkpad.thebehrens.net> <alpine.DEB.2.00.1412120644340.23559@cobra.newdream.net> <548B0C16.4000509@gmail.com>
- *User-agent*: Alpine 2.00 (DEB 1167 2008-08-23)

0

On Fri, 12 Dec 2014, Joe Landman wrote:
> On 12/12/2014 09:46 AM, Sage Weil wrote:
> > On Fri, 12 Dec 2014, Thorsten Behrens wrote:
> > > Sage Weil wrote:
> > > > Calculating the ln() function is a bit annoying because it is a floating
> > > > point function and CRUSH is all fixed-point arithmetic (integer-based).
> > > > The current draft implementation uses a 128 KB lookup table (2 bytes per
> > > > entry for 16 bits of input precision). It seems to be about 25% slower
> > > > than the original straw code in my simple microbenchmark, but I'm not
> > > > sure
> > > > that's a number we should trust since it loops through a zillion inputs
> > > > and will pull most of the lookup table into the processor caches.
> > > >
> > > > We could probably try a few others things:
> > > >
> > > [snip]
> > >
> > > My Hacker's Delight copy has a number of further variants (on
> > > pp. 215), unfortunately only covering log_2 (which is trivially the
> > > number of leading zeros in the binary representation) and log_10. But
> > > since the logarithms are simply related by a multiplicative constant,
> > > something clever might be doable here with a bit of thinking.
> > Any log will definitely do; the next steps are to do a division and then
> > take the max. Simply counting 0's isn't precise enough, though.. we need
> > a lot more precision than that.
>
> You want integer log? Looks like ln(x * 2^-16) from your original
> description., with x = (0 .. 2^16)
> So your log is going from undefined to 0 (ln 0 is not defined). Fire up
> gnuplot and do a quick "plot [0:1] (log(x)) " to see the shape of the
> function.
>
> It can be a real valued function scaled in a particular way, then we can use a
> nice set of series to calculate it in FP (single precision if you like).
> Conversion to int should be fast, and we could do this in SSE2/AVX to do
> multiple calcs at the same time.

We can't use floating point. The code needs to run in the kernel. We also need the rseults to be perfectly deterministic and consistent across all architectures; I'm not sure if all floating point implementations (and log implementations) will do that?

> *I would think the random value portion of*
> *the code would be the slow part. Where are you getting your random source?*
> *Could you not start that out with an exponential distribution rather than*
> *generate your own?*

That would simplify things, but I'm not sure how practical it is. In reality, the "random" value is pseudorandom, and looks like this:

```
 rjhash(crush_input_x, bucket_id, item_id, attempt)
```

The input 'x' is basically the PG id. Attempt is incremented as CRUSH makes multiple samples to find a mapping that is suitable. Note that the position of item in the bucket is not an input, so this result is stable when other bucket members change. The current hash function we're using is based on Richard Jenkin's 32-bit mix function. Here:

https://github.com/ceph/ceph/blob/master/src/crush/hash.c

> *Aside from that, I'd suggest doing something like this (as you don't really*
> *need the ln(x/65536)/weight term, just the ln(x) term ... as you are*
> *effectively throwing away the value and just using side effects*

The weight is actually item_weight and is the key to getting more mappings on more heavily weighted items. But it's a simple integer divide, so cheap.

If SSEx/AVX could be leveraged that'd be great, but I think it needs to be fixed-point arithmetic...

Thanks!
sage

```
>  max_x = -1
>  max_item = -1
>  mult_const = ln(1/65536)/weight
>  for each item:
>     x = random value from 0..65535
>     x = ln(x)
>     if x > max_x:
>        max_x = x
>        max_item = item
>  return item
>
> The random value is likely to be an expensive portion of the algorithm ...
> possibly have a thread to pre-queue a few thousand random values that you
> queue up in a ring buffer somewhere?
>
> I'd be happy to code up a quick log calculator if that really is the most
> expensive portion of this.  I think we might be able to do the logs and even
> the comparisons in SSEx/AVX if you are open for that (exploit some loop
> unrolling), though we'd need a serial finishing loop to deal with the
> unrollable portion.
>
>
>
> >
> > The table lookup is (I think) on the order of 50-100 cycles with a cold
> > cache; that's what we need to beat!
> >
```

> > sage
> >
> > --
> > To unsubscribe from this list: send the line "unsubscribe ceph-devel" in
> > the body of a message to majordomo@xxxxxxxxxxxxxxx
> > More majordomo info at http://vger.kernel.org/majordomo-info.html
>
> --
> To unsubscribe from this list: send the line "unsubscribe ceph-devel" in
> the body of a message to majordomo@xxxxxxxxxxxxxxx
> More majordomo info at http://vger.kernel.org/majordomo-info.html
>
>
--
To unsubscribe from this list: send the line "unsubscribe ceph-devel" in
the body of a message to majordomo@xxxxxxxxxxxxxxx
More majordomo info at http://vger.kernel.org/majordomo-info.html

---

- **Follow-Ups**:
  - **Re: crush: straw is dead, long live straw2**
    - *From:* Joe Landman
  - **Re: crush: straw is dead, long live straw2**
    - *From:* Joe Landman

- **References**:
  - **crush: straw is dead, long live straw2**
    - *From:* Sage Weil
  - **Re: crush: straw is dead, long live straw2**
    - *From:* Thorsten Behrens
  - **Re: crush: straw is dead, long live straw2**
    - *From:* Sage Weil
  - **Re: crush: straw is dead, long live straw2**
    - *From:* Joe Landman

- Prev by Date: **Re: Reducing backfilling/recovery long tail**
- Next by Date: **Re: crush: straw is dead, long live straw2**
- Previous by thread: **Re: crush: straw is dead, long live straw2**
- Next by thread: **Re: crush: straw is dead, long live straw2**
- Index(es):
  - **Date**
  - **Thread**