



Hoff, Foster <foster_hoff@brown.edu>

Nanocubes Capstone Project

9 messages

Hoff, Foster <foster_hoff@brown.edu>

Mon, Dec 14, 2015 at 3:30 PM

To: llins@research.att.com

Cc: Alfonso Subiotto Marques <alfonso_subiotto_marques@brown.edu>

Hi Mr. Lins,

I'm very impressed with your work on Nanocubes and Alfonso and I have decided to adopt the project for our capstone project at Brown University. My partner and I are working on adding features such as the parallel distribution of the Nanocubes generation process.

Unfortunately we're experiencing a hiccup in the regular unparallelized Nanocubes generation process of a larger crime dataset than the default one provided. As you can see from the image I've attached, after count 1,060,000 the process slows down immensely. Do you have any thoughts as to why we might be experiencing this? Would love to chat more about the project.

Regards,
Foster Hoff & Alfonso Subiotto Marques

Count	Time	Other Metrics
800000	86450	
810000	64527	
820000	52094	
830000	60490	
840000	63776	
850000	66655	
860000	66320	
870000	67242	
880000	67327	
890000	67703	
900000	67982	
910000	68251	
920000	68526	
930000	68811	
940000	69089	
950000	69366	
960000	72979	
970000	113220	
980000	133181	
990000	131125	
1000000	230639	
1010000	267333	
1020000	313693	
1030000	339159	
1040000	373034	
1050000	409725	
1060000	445350	
1070000	492039	
1080000	533964	

Screen Shot 2015-12-14 at 3.13.21 PM.png
57K

Lauro Lins <llins@research.att.com>

Mon, Dec 14, 2015 at 3:46 PM

To: "Hoff, Foster" <foster_hoff@brown.edu>

Cc: Alfonso Subiotto Marques <alfonso_subiotto_marques@brown.edu>

Hi Foster,

My guess is that the input records are not sorted in time. When inserting a record, the last (and special) dimension is time and it is stored in a standard C++ vector (See Figure 6 of the paper: "summed table sparse representation"). If the timestamp of the current record is smaller than a previously stored record in that same multidimensional-bin-of-all-dimensions-except-time, then we need to open a slot in the middle of the vector, push elements to the right and recompute cumulative values stored in that vector. This adds an extra linear cost in the length of the time series instead of constant time. It tends to get worse when the timeseries get larger and larger (which matches the evidence from your case).

Best,
Lauro

[Quoted text hidden]

> <Screen Shot 2015-12-14 at 3.13.21 PM.png>

Hoff, Foster <foster_hoff@brown.edu>

Mon, Dec 14, 2015 at 4:05 PM

To: Lauro Lins <llins@research.att.com>

Cc: Alfonso Subiotto Marques <alfonso_subiotto_marques@brown.edu>

Hi Lauro,

Thanks for getting back to me so quickly. The time dimension is indeed the special dimension and we are working hard to optimize for the distributed version of the Nanocube. I can see now how inserting a record with a smaller time than those previously stored would require linear cost in the length of the time series. However, isn't the data sorted by time in nanocube-binning-csv? I had to increase the chunksize to allow for the larger dataset of course.

Regards,
Foster

[Quoted text hidden]

Lauro Lins <llins@research.att.com>

Mon, Dec 14, 2015 at 5:28 PM

To: "Hoff, Foster" <foster_hoff@brown.edu>

Cc: Alfonso Subiotto Marques <alfonso_subiotto_marques@brown.edu>

Hi Foster,

I am double checking with my colleague that wrote the nanocube-binning-csv if it guarantees that the data will be sorted in time, given that the chunk size is large enough to fit all the records. From a quick scan on the nanocube-binning-csv python script you might be right and the data is already sorted in time.

If you point me to your .csv file and the command you are using, I can test on my side if there is something strange and things get slow after the 1M records mark.

Thanks,
Lauro

[Quoted text hidden]

Hoff, Foster <foster_hoff@brown.edu>

Mon, Dec 14, 2015 at 8:38 PM

To: Lauro Lins <llins@research.att.com>

Hi Lauro,

I do indeed believe that is the case, so please let me know what you hear back. The dataset I used is the full version of the Chicago crime dataset located here: <https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2>

I downloaded it in time-descending order as you suggested, and at first it seemed to fix the problem, but then I ran into an even bigger hiccup (see attached image). If you have any thoughts as to why this might be occurring please just let me know. I've also managed to generate a Nanocube with no hiccups on New York taxi data located here: <http://www.andresmh.com/nyctaxitrips/> which is also in time-descending order.

Cheers,
Foster

[Quoted text hidden]

ptd2a	count:	3320000	mem:	66780	time(ms):	165124
ptd2a	count:	3330000	mem:	66580	time(ms):	165585
ptd2a	count:	3340000	mem:	67280	time(ms):	166038
ptd2a	count:	3350000	mem:	67480	time(ms):	166493
ptd2a	count:	3360000	mem:	67880	time(ms):	166943
ptd2a	count:	3370000	mem:	67980	time(ms):	167407
ptd2a	count:	3380000	mem:	68280	time(ms):	167872
ptd2a	count:	3390000	mem:	68380	time(ms):	168336
ptd2a	count:	3400000	mem:	68580	time(ms):	168804
ptd2a	count:	3410000	mem:	68880	time(ms):	169255
ptd2a	count:	3420000	mem:	69380	time(ms):	169715
ptd2a	count:	3430000	mem:	69380	time(ms):	687944
ptd2a	count:	3440000	mem:	69580	time(ms):	1033832

Screen Shot 2015-12-14 at 5:33.38 PM.png
25K

Lauro Lins <llins@research.att.com>
To: "Hoff, Foster" <foster_hoff@brown.edu>

Mon, Dec 14, 2015 at 8:49 PM

Hi Foster,

I haven't tested your dataset yet, but one possibility is that the time resolution you are using is not enough and the timestamps are getting wrapped: i.e. 2^{16} time bins and we have data in more than 2^{16} bins. If we choose minute resolution and 16-bit timestamps we have 1440 minute time bins per day and after ~45 days the numbers would wrap around. Maybe you could check that.

Lauro

[Quoted text hidden]

[Quoted text hidden]

<Screen Shot 2015-12-14 at 5.33.38 PM.png>

Hoff, Foster <foster_hoff@brown.edu>
To: Lauro Lins <llins@research.att.com>

Mon, Dec 14, 2015 at 9:06 PM

Hi Lauro,

Astute observation. That does indeed seem to be the case, as the visualization cuts off around 2010 when it should continue through 2015. I can see how to increase the time resolution clearly using the '--timebinsize' argument to nanocube-binning-csv, but how would I go about increasing the number of time bins? Thanks again for all your help; results are soon to come!

Foster

[Quoted text hidden]

Lauro Lins <llins@research.att.com>
To: "Hoff, Foster" <foster_hoff@brown.edu>

Mon, Dec 14, 2015 at 11:48 PM

Hi Foster,

I see that you can increase the standard 2 bytes (2^{16} timebins) by using --timebytes=4 or --timebytes=8. One thing that you need to do to be able to load such data is to have the right binary executable for that schema. See this documentation,

<https://github.com/lauro/lin/nanocube/wiki/nanocube-ready-dmp>

specially the end to see how to compile a binary to get more temporal resolution.

Lauro

I see that `--timebytes`

[Quoted text hidden]

Hoff, Foster <foster_hoff@brown.edu>
To: Lauro Lins <llins@research.att.com>

Tue, Dec 15, 2015 at 10:35 AM

Hi Lauro,

Great, thank you. We're now in the process of benchmarking a couple datasets on our distributed system. The

12/16/2015

Brown University Mail - Nanocubes Capstone Project

only downside to this distributed system is the fact that the key space does not share between nodes so it takes a larger amount of memory in total. I saw your link to the quadtree partitioning scheme which we might have to try soon.

By the way, are any of the datasets you published (twitter, splom, cdrs) publicly available? The more benchmarks we can run, the better!

Regards,
Foster

[Quoted text hidden]