

Results

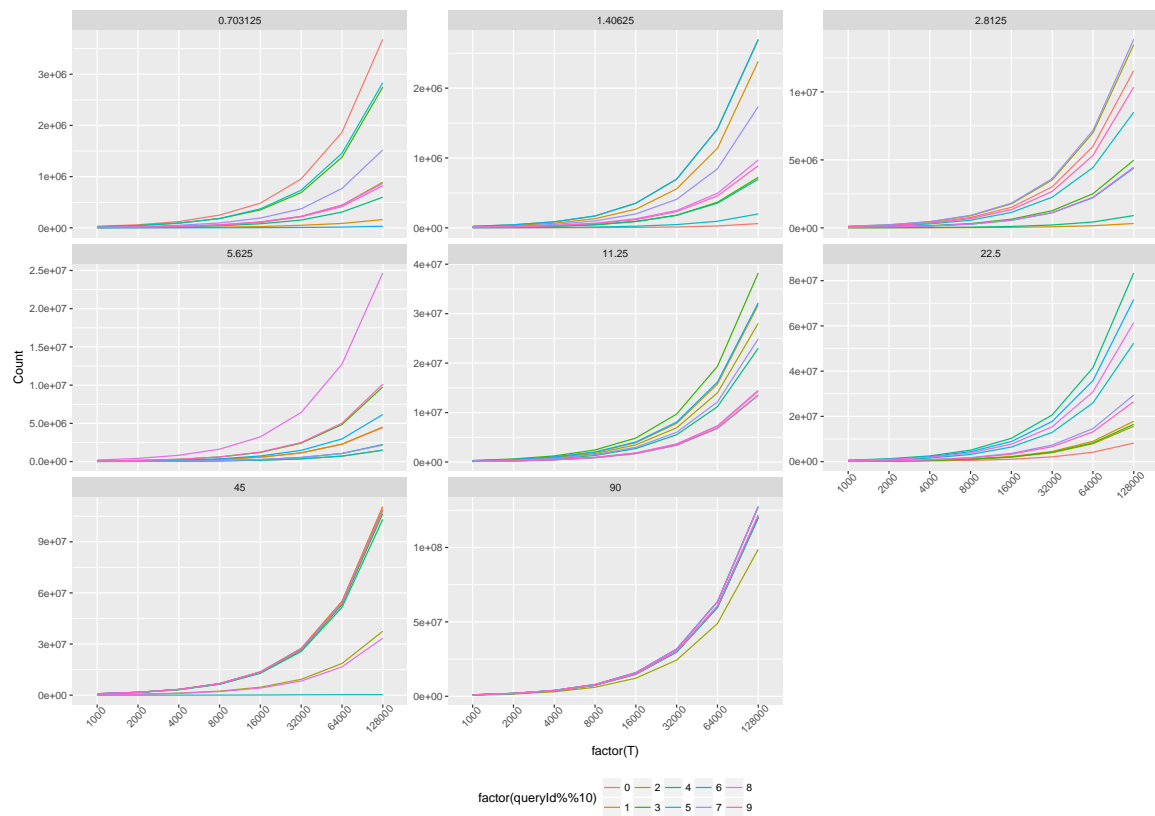
October 16, 2017

Contents

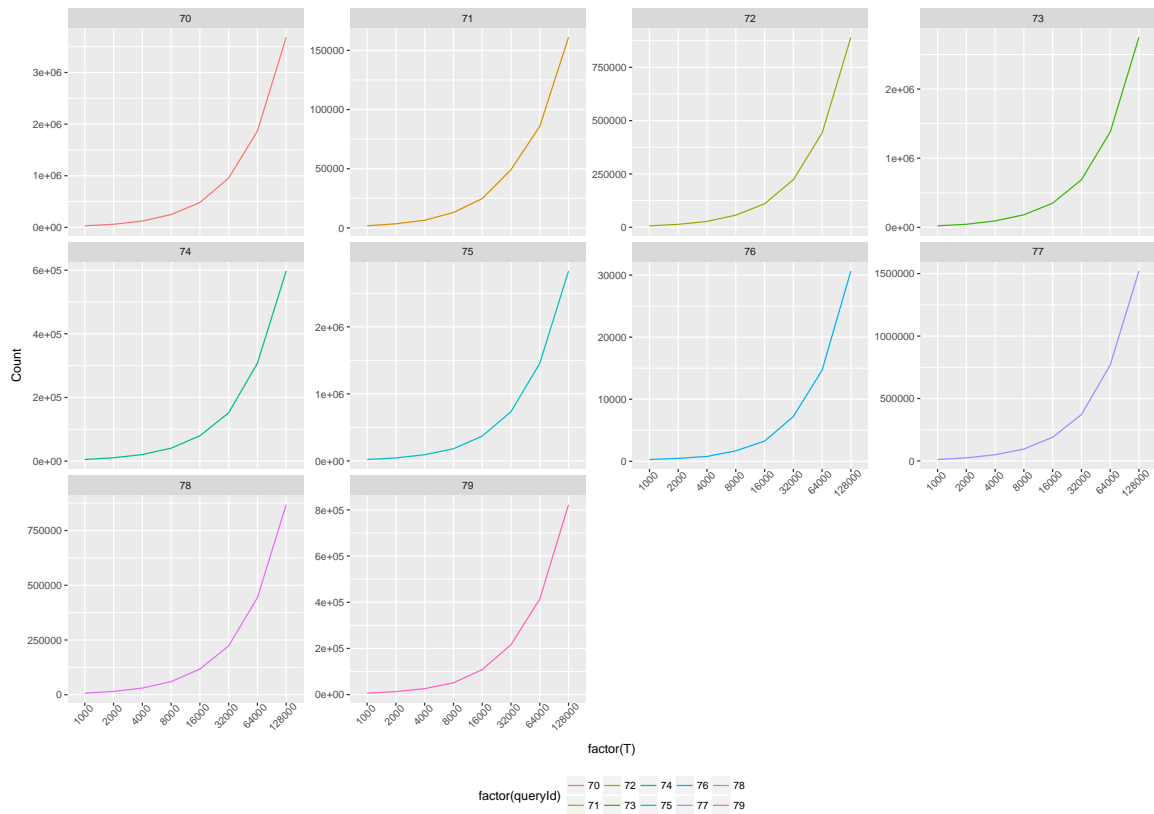
1	Plot: change of query count with size of T.	1
2	Plot: Scan Query Time by T facet by queryId	3
3	Plot: Scan query time by Query Count faceted by QueryId	4
4	Plot: Scan query throughput by Query Count faceted by QueryId	5

1 Plot: change of query count with size of T.

```
dfAvg %>%  
  filter(algo == "BTree" & bench == "apply_at_region") %>%  
  ggplot(aes(x = factor(T), y = Count, color = factor(queryId%%10))) +  
  geom_line(aes(group=queryId))+  
  facet_wrap(~queryWidth, scale = "free_y") +  
  theme(legend.position = "bottom",  
        axis.text.x = element_text(angle = 45, hjust = 0.75))
```

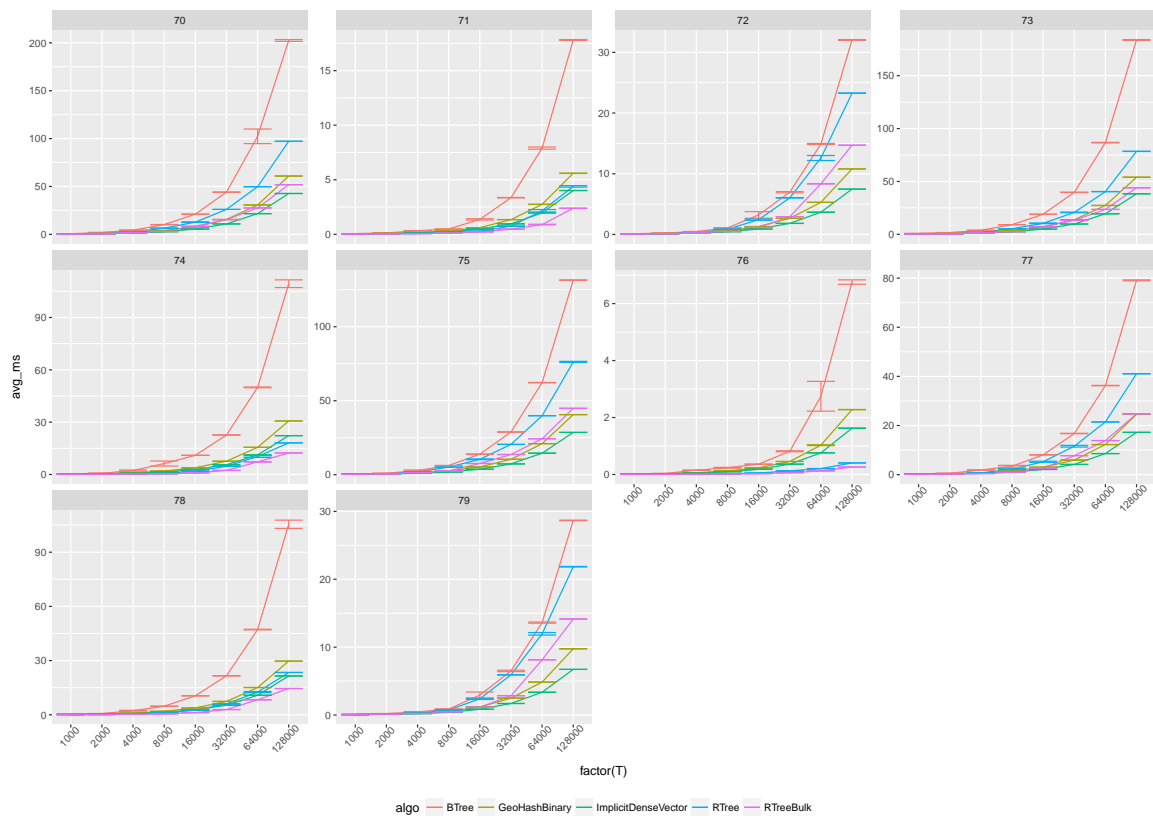


```
dfAvg %>%
  filter(algo == "BTree" & bench == "apply_at_region" & queryWidth < 1) %>%
  ggplot(aes(x = factor(T), y = Count, color = factor(queryId))) +
  geom_line(aes(group=queryId))+
  facet_wrap(~queryId, scale = "free_y") +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 45, hjust = 0.75))
```



2 Plot: Scan Query Time by T facet by queryId

```
dfAvg %>%
  filter(bench == "scan_at_region" & queryWidth < 1) %>%
  ggplot(aes(x = factor(T), y = avg_ms, color = algo)) +
  geom_line(aes(group=algo))+
  geom_errorbar(aes(ymin = avg_ms - stdv , ymax = avg_ms + stdv)) +
  facet_wrap(~queryId, scale = "free_y") +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 45, hjust = 0.75))
```



3 Plot: Scan query time by Query Count faceted by QueryId

```
dfAvg %>%
  ungroup() %>%
  filter(bench == "apply_at_region") %>%
  select(queryId, algo, Count, T) %>%
  left_join(
    filter(ungroup(dfAvg), bench == "scan_at_region") %>% select(-Count)
  ) -> dfCount
dfCount
```

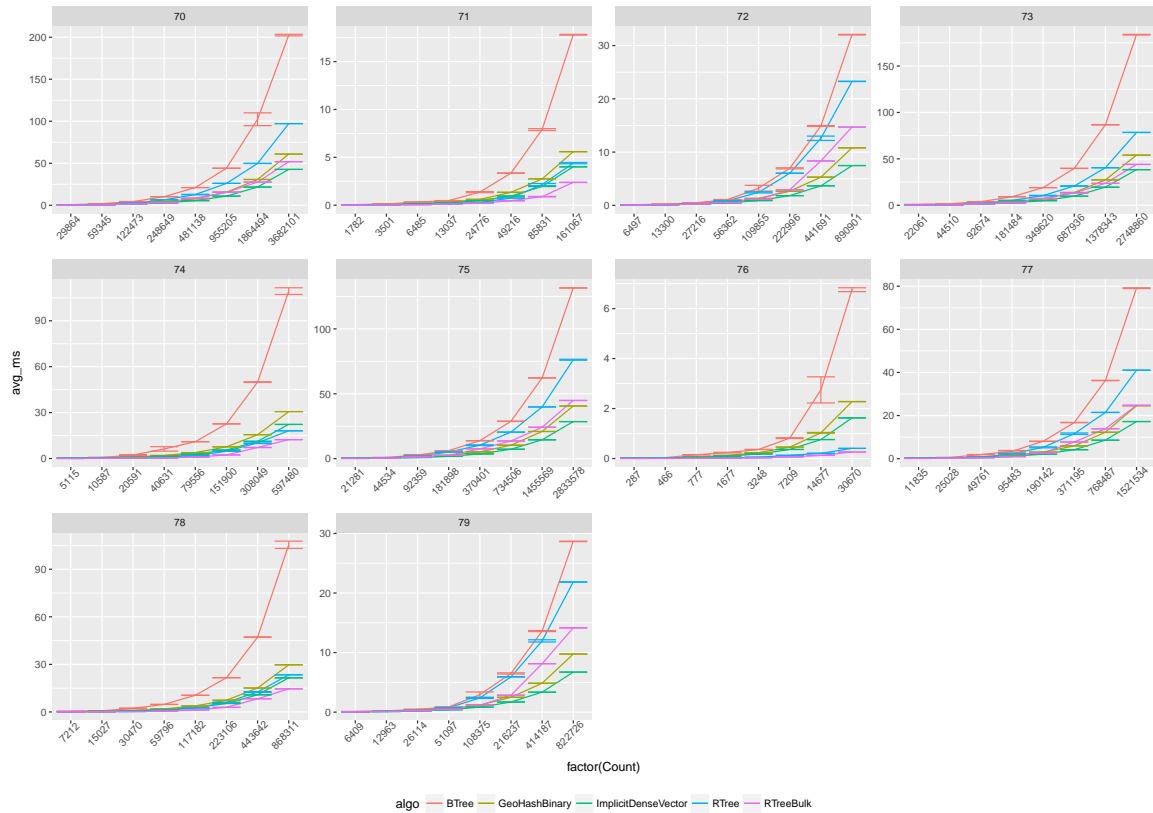
```
Joining, by = c("queryId", "algo", "T")
```

```
# A tibble: 3,200 x 9
```

	queryId	algo	Count	T	bench	Refine	queryWidth	avg_ms	stdv
	<int>	<chr>	<int>	<int>	<chr>	<int>	<dbl>	<dbl>	<dbl>
1	0	BTree	924827	1000	scan_at_region	29	90	17.44709	0.03102001
2	0	BTree	1855890	2000	scan_at_region	29	90	36.30205	0.03723738
3	0	BTree	3706387	4000	scan_at_region	33	90	74.14725	0.02328167
4	0	BTree	7417949	8000	scan_at_region	45	90	151.37550	0.03572192
5	0	BTree	14876686	16000	scan_at_region	52	90	307.56140	0.09391154
6	0	BTree	29764961	32000	scan_at_region	64	90	691.87770	0.59023687
7	0	BTree	59715461	64000	scan_at_region	89	90	1679.86400	1.45107010
8	0	BTree	119931295	128000	scan_at_region	118	90	3746.40000	14.13264542
9	1	BTree	929918	1000	scan_at_region	46	90	17.40384	0.03499172
10	1	BTree	1866101	2000	scan_at_region	46	90	36.21215	0.03681069

```
# ... with 3,190 more rows
```

```
dfCount %>%
  filter(queryWidth < 1) %>%
  ggplot(aes(x = factor(Count), y = avg_ms, color = algo)) +
  geom_line(aes(group=algo))+
  geom_errorbar(aes(ymin = avg_ms - stdv , ymax = avg_ms + stdv)) +
  facet_wrap(~queryId, scale = "free") +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 45, hjust = 0.75))
```



4 Plot: Scan query throughput by Query Count faceted by QueryId

```
dfCount %>%
  filter(queryWidth < 1) %>%
  arrange(Count,T) %>%
  mutate(lbls = paste(Count," (",T/1000,")",sep="")) %>%
  ggplot(aes(x = factor(lbls,levels=unique(lbls)), group=algo, y = Count / avg_ms, color = algo)) +
  geom_errorbar(aes(ymin = Count / (avg_ms - stdv) , ymax = Count / (avg_ms + stdv)) ) +
  geom_line() +
  facet_wrap(~queryId,scale="free") +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 45, hjust = 1))+
  labs(x = "Query Count ( Dataset size x 106 )")
```

