

Week#5 Space Utilization of MySQL

Sangeun Chae
2018314760

1. INTRODUCTION

Space amplification 이란 Database 의 Data 크기에 대한 Database Size 의 비율을 뜻한다 (The size of the database / The size of the data in the database). MySQL 에서는 space amplification 문제가 대두되는데, 이는 MySQL 에서 사용하는 자료 구조인, B-+Tree 의 fragmentation 에 의해 space amplification 현상이 심화된다. 따라서 이번 lab 에서는, Tpc-C benchmark 전후로 MySQL 각각의 table 의 space utilization 을 비교 분석할 예정이다.

2. METHODS

Tpc benchmark 에서 사용되는 8 개의 table 은 다음과 같다 (item.ibd, district.idb, order_line.ibd, stock.ibd, customer.ibd, history.ibd, new_orders.ibd, orders.ibd). 해당 table 을 space utilization 분석하는 2 개의 binary 파일 (cal-free-space-from-ibd, cal-free-space-percentile-from-parsed-file)을 실행하여, benchmark 전후로 각각 output file 을 저장하여 비교 분석한다.

3. Performance Evaluation

3.1 Experimental Setup

Type	Specification
OS	Ubuntu 20.04.3 LTS
CPU	AMD Ryzen 7 5800X 8-Core Processor (VMware support 4 Core)
Memory	4GB
Kernel	Linux ubuntu 5.11.0.34-generic
Disk	VMware Virtual 40GB

Table 1: System setup

Type	Configuration
DB size	2GB (20 Warehouse)
Buffer Pool Size	600MB (30% of DB size)
Benchmark Tool	Tpc-mysql
Runtime	1200s
Connections	8

Table 2: Benchmark setup

3.2 Experimental Results

3.2.1 Result new_order.ibd

new_order start 와 new_order_end 는 각각, benchmark 전후를 나타낸다. 해당 table 에서는, benchmark 전후로, 총 page 의 수는 차이가 나지 않았다. 하지만, 0-10 퍼센트의 free page 의 분포도는 1.6% 정도 올라간 것을 알 수 있고, 나머지 퍼센트의 free page 의 비율도 0.1% 정도로 미세하게 바뀐 것을 알 수 있다.



Figure 1: new_order.ibd table result [1]

Percentag	new_order_star	new_order_end
1	57.70%	59.30%
2	1.40%	1.10%
3	1.70%	1.80%
4	1.60%	1.30%
5	1.80%	1.60%
6	1.40%	1.30%
7	26.90%	26.60%
8	1.50%	1.40%
9	4.30%	4.20%
10	1%	1%
Total	3581	3581

Figure 2: new_order.ibd table result [2]

	New_order_start	New_order_end
Average free space per page	2692 (65.74%)	2701 (65.96%)
Average free space per leaf page	2716 (66.33%)	2726 (66.56%)
Average free space per non-leaf page	1963 (47.93%)	1968 (48.06%)

Figure 3: new_order.ibd table result [3]

3.2.2 Result customer.ibd

customer_start 와 customer_end 는 각각, benchmark 전후를 나타낸다. 해당 table 에서는, benchmark 전후로, 총 page 수가 차이 나지 않았고, free page 의 space percentile 의 결과도 변하지 않았다.

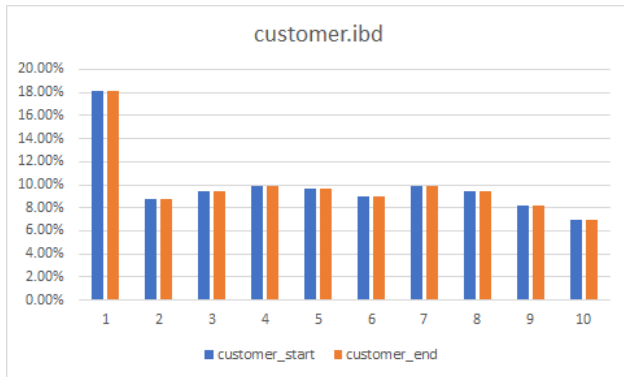


Figure 4: customer.ibd table result [1]

Percentag	customer_star	customer_end
1	18.10%	18.10%
2	8.80%	8.80%
3	9.40%	9.40%
4	9.90%	9.90%
5	9.70%	9.70%
6	9.00%	9.00%
7	9.90%	9.90%
8	9.40%	9.40%
9	8.20%	8.20%
10	7%	7%
Total	120773	120773

Figure 5: customer.ibd table result [2]

	Customer_start	Customer_end
Average free space per page	950 (23.20%)	950 (23.20%)
Average free space per leaf page	952 (23.26%)	952 (23.26%)
Average free space per non-leaf page	526 (12.85%)	526 (12.85%)

Figure 6: customer.ibd table result [3]

3.2.3 Result district.ibd

district_start 와 district_end 는 각각, benchmark 전후를 나타낸다. 해당 table 에서는, benchmark 전후로, 총 page 수가 차이 나지 않았고, free page 의 space percentile 의 결과도 변하지 않았다.

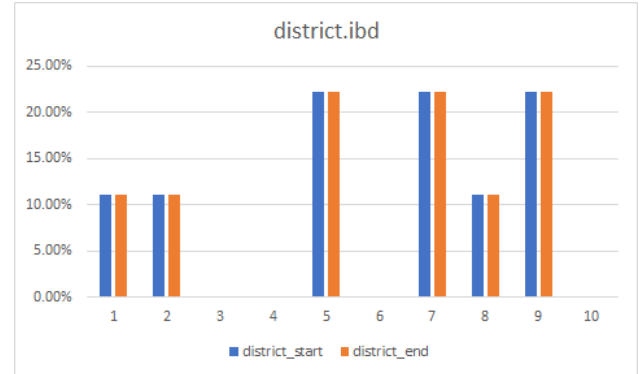


Figure 7: district.ibd table result [1]

Percentag	district_star	district_end
1	11.10%	11.10%
2	11.10%	11.10%
3	0%	0%
4	0%	0%
5	22.20%	22.20%
6	0%	0%
7	22.20%	22.20%
8	11.10%	11.10%
9	22%	22%
10	0%	0%
Total	9	9

Figure 8: district.ibd table result [2]

	District_start	District_end
Average free space per page	1674 (40.88%)	1674 (40.88%)
Average free space per leaf page	1400 (34.19%)	1400 (34.19%)
Average free space per non-leaf page	3866 (94.38%)	3866 (94.38%)

Figure 9: district.ibd table result [3]

3.2.4 Result history.ibd

history_start 와 history_end 는 각각, benchmark 전후를 나타낸다. 해당 table 에서는, benchmark 전후로, 총 page 의 수가 20,000 정도 차이가 났다. 또한, 0-10 퍼센트의 free page 의 분포도는 2.3% 정도 증가한 것을 알 수 있고, 80-100 퍼센트의 free page 의 분포도는 benchmark 를 시작하기 전이 0.1% ~ 0.5% 정도 미세하게 높음을 알 수 있다.

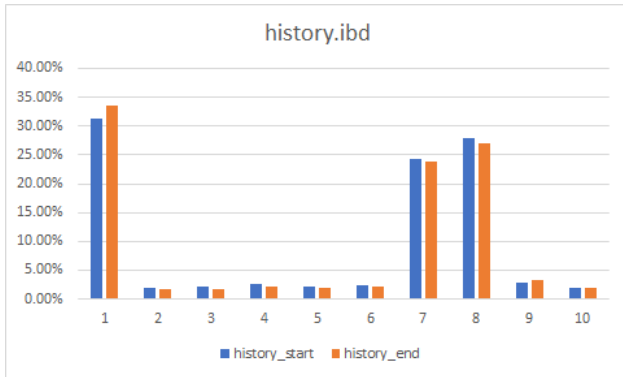


Figure 10: history.ibd table result [1]

Percentag	history_star	history_end
1	31.20%	33.50%
2	2%	1.60%
3	2.10%	1.70%
4	2.70%	2.10%
5	2.20%	1.90%
6	2.40%	2.10%
7	24.40%	23.90%
8	28.00%	27.00%
9	2.80%	3.40%
10	2%	2%
Total	37869	56291

Figure 11: history.ibd table result [2]

	History_start	History_end
Average free space per page	1213 (29.63%)	1320 (32.24%)
Average free space per leaf page	1215 (29.67%)	1324 (32.32%)
Average free space per non-leaf page	814 (19.88%)	546 (13.34%)

Figure 12: history.ibd table result [3]

3.2.5 Result item.ibd

item_start 와 item_end 는 각각, benchmark 전후를 나타낸다. 해당 table 에서는, benchmark 전후로, 총 page 수가 차이 나지 않았고, free page 의 space percentile 의 결과도 변하지 않았다.

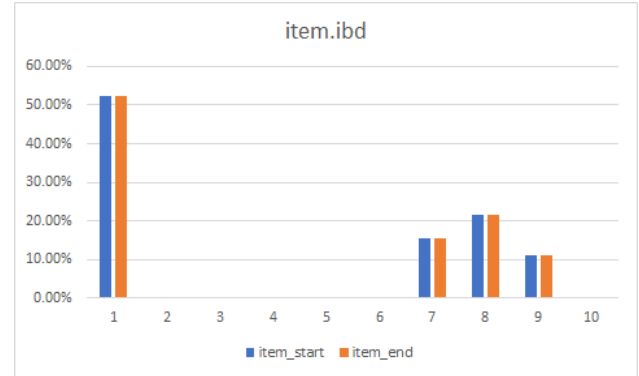


Figure 13: item.ibd table result [1]

Percentag	item_star	item_end
1	52.20%	52.20%
2	0.00%	0.00%
3	0.00%	0.00%
4	0%	0%
5	0%	0%
6	0%	0%
7	15.40%	15.40%
8	21.50%	21.50%
9	10.90%	10.90%
10	0.00%	0.00%
Total	5117	5117

Figure 14: item.ibd table result [2]

	Item_start	Item_end
Average free space per page	2278 (55.62%)	2278 (55.62%)
Average free space per leaf page	2281 (55.71%)	2281 (55.71%)
Average free space per non-leaf page	732 (17.89%)	732 (17.89%)

Figure 15: item.ibd table result [3]

3.2.6 Result order_line.ibd

order_line_start 와 order_line_end 는 각각, benchmark 전후를 나타낸다. 해당 table 에서는, benchmark 전후로, 총 page 의 수가 약 **150,000** 정도 차이가 남을 확인할 수 있다. 또한, 10-20 퍼센트의 free page 의 분포도가 **11.1%** 정도 비율이 증가했고, 80-100 퍼센트의 free page 의 분포도는 benchmark 를 시작하기 전이 **2%~5%**정도 높음을 알 수 있다. 또한, 전반적인 page 당 free space 의 비율이 benchmark 전보다 높아졌음을 알 수 있다.



Figure 16: order_line.ibd table result [1]

Percentag	order_line_star	order_line_end
1	21%	21.30%
2	15.80%	26.90%
3	2.10%	1.70%
4	2.30%	1.90%
5	2.30%	2.00%
6	2.40%	2.20%
7	10.60%	8.30%
8	23.00%	17.80%
9	10.40%	9.00%
10	10%	8%
Total	373579	520965

Figure 17: order_line.ibd table result [2]

	Order_line_start	Order_line_end
Average free space per page	1431 (34.96%)	1486 (36.30%)
Average free space per leaf page	1434 (35.01%)	1488 (36.35%)
Average free space per non-leaf page	1010 (24.66%)	1159 (28.31%)

Figure 18: order_line.ibd table result [3]

3.2.7 Result orders.ibd

order_start 와 order_end 는 각각, benchmark 전후를 나타낸다. 해당 table 에서는, benchmark 전후로, 총 page 의 수가 약 **10,000** 정도 차이가 남을 확인할 수 있다. 다른 table 과는 다르게, orders.ibd table 은 0-10 퍼센트의 free page 분포도가 benchmark 를 시작하기 전의 비율이 benchmark 후의 비율보다 **3.7%** 정도 높음을 알 수 있다. 하지만, 다른 퍼센트의 free page 의 분포도가 benchmark 후의 비율이 더 높기 때문에, 전반적인 free page 의 비율은 benchmark 후의 비율이 benchmark 전 비율보다 높아졌음을 알 수 있다.

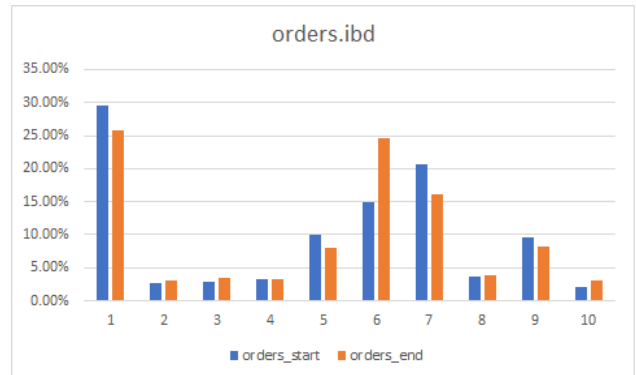


Figure 19: orders.ibd table result [1]

Percentag	orders_star	orders_end
1	29.60%	25.90%
2	2.70%	3.10%
3	2.90%	3.50%
4	3.30%	3.30%
5	9.90%	8.10%
6	14.90%	24.60%
7	20.70%	16.10%
8	3.60%	3.90%
9	9.50%	8.20%
10	2%	3%
Total	25587	347.97

Figure 20: orders.ibd table result [2]

	Orders_start	Orders_end
Average free space per page	1496 (36.53%)	1634 (39.91%)
Average free space per leaf page	1496 (36.55%)	1639 (40.02%)
Average free space per non-leaf page	1399 (34.17%)	527 (12.87%)

Figure 21: orders.ibd table result [3]

3.2.8 Result stock.ibd

stock_start 와 stock_end 는 각각, benchmark 전후를 나타낸다. 해당 table 에서는, benchmark 전후로, 총 page 수가 차이 나지 않았고, free page 의 space percentile 의 결과도 변하지 않았다.

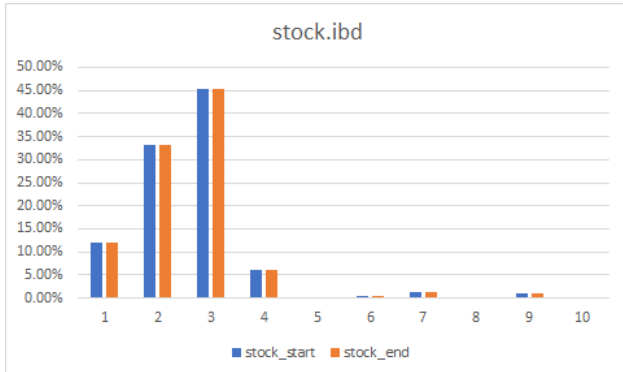


Figure 22:stock.ibd table result [1]

Percentag	stock_star	stock_end
1	12.10%	12.10%
2	33.30%	33.30%
3	45.40%	45.40%
4	6.00%	6.00%
5	0.00%	0.00%
6	0.40%	0.40%
7	1.20%	1.20%
8	0.00%	0.00%
9	1.00%	1.00%
10	0%	0%
Total	210841	210841

Figure 23: stock.ibd table result [2]

	Stock_start	Stock_end
Average free space per page	861 (21.04%)	861 (21.04%)
Average free space per leaf page	865 (21.12%)	865 (21.12%)
Average free space per non-leaf page	123 (3.02%)	123 (3.02%)

Figure 24: stock.ibd table result [3]

3.2.9 Overall Analysis

3.2.9.1 B+-Tree

B+-Tree 는 Database 와 Filesystem 에서 널리 사용되는 트리 자료구조의 일종으로, binary tree 를 확장해 하나의 노드가 가질 수 있는 leaf node 의 최대 숫자가 2 보다 큰 트리 구조이다. B+-Tree 의 기본 개념은 internal node 의 leaf node 의 수가 미리 정해진 범위 내에서 변경이 가능하고, 항목이 삽입되거나 삭제될 때, internal node 는 해당 범위의 leaf node 의 수를 만족시키기 위해 분리되거나 혹은 다른 노드와 합쳐지게 된다. Leaf node 의 수가 일정 범위 내에서만 유지되면 되므로 분리 및 합침을 통한 재 균형 과정은 다른 이진 트리만큼 자주 일어나지 않지만, 해당 원인이 space amplification 의 원인이 된다. B+-Tree leaf-node에서 50%에서 70%정도로 page에 빈 공간(free page)가 random update 를 통해 생기게 된다.

3.2.9.2 TPC-C Benchmark

TPC-C 는 OLTP & Batch Application 을 위한 대표적인 성능 benchmark 이다. Benchmark 대상 서버에 대한 성능 측정치는 tmpC 로 표시하는데, 이는 분당 처리하는 New-Order 의 “비즈니스 성취도”를 나타낸다. 즉, DB 수준의 Transaction 이 아닌 다섯 가지 종류의 비즈니스 Transaction 을 각 비율대로 동시에 발생시킨 후 안정적인 상태에서 분당 처리된 New-Order Transaction 건수를 말한다.

3.2.9.3 Analysis

TPC-C benchmark 전과 후를 비교하는 실험을 진행한 결과, space utilization 변화를 나타내는 table 은 new_order.ibd, history.ibd, order_line.ibd, orders.ibd 이다. 이는 TPC-C Benchmark 가 앞에서 언급한 바와 같이, 분당 처리된 New-Order Transaction 처리의 성취도를 나타내는 지표 임으로, New-Order Transaction 과 관련된 table 에 변화가 있는 것으로 보인다. 변화의 양상은 다음과 같다. 전반적으로 4 개중 3 개의 table 의 page 수가 증가되는 양상을 보이고, benchmark 전보다 후에 free page 의 percentage 가 증가함을 보인다. Page 수가 증가하는 것은, 해당 New-Order transaction 을 처리하는 과정에서, 해당 table 에 추가된 data 의 size 가 증가했다는 뜻으로 유추할 수 있다. 또한 전반적으로 free page 가 증가하고 변화한 것은 B+-Tree 특성에서 유추할 수 있다. 특정 항목이 삽입되거나 삭제될 때, internal node (non-leaf node)는 leaf node 의 수를 만족시키기 위해 분리되거나 합쳐지는 과정을 수행한다. 이러한 과정에서 database 에 불필요하게 사용되는 공간이

증가하게 되고, 이는 곧 space amplification 으로 이어지는 것으로 유추할 수 있다.

4. Conclusion

Database 에서 search query 의 처리 성능은 중요한 지표로 볼 수 있다. 검색을 위한 자료구조 중에서 이진 트리는 비록 하나의 부모가 두개의 자식밖에 가지질 못하고 자칫 균형이 맞지 않으면 검색 효율이 월등히 떨어진다. 그렇지만 이진 트리는 구조의 간결함과 균형만 맞으면, search, insert, delete 모두 우수한 성능을 보이는 장점이 있다. 그 중에서도 B-Tree 는 이진 트리를 확장해서 더 많은 수의 자식을 가질 수 있게 일반화한 자료구조이다. 따라서 대량의 데이터를 처리해야 하는 구조인 경우, 하나의 노드에 많은 데이터를 가질 수 있다는 것은 큰 장점이다. 하지만 이러한 장점이 단점으로 이어질 수 있다. 특정 데이터가 삽입되거나 삭제될 때, 내부 노드의 병합과 분리로 인해서 불필요하게 사용되는 공간(Free page)이 늘어나게 된다. 이는 TPC-C benchmark 를 통해 특정 table 의 space utilization 을 비교 분석하며 유추할 수 있었다. 따라서 이러한 문제점은, HDD 와 같은 “cheap disk”에는 문제가 되지 않겠지만, SSD와 같은 “expensive flash storage”에 있어서는 문제가 된다.

5. REFERENCES

- [1] Week5. Space Utilization of MySQL.pdf
- [2] <https://bit.ly/dbproject-week5>
- [3] ko.wikipedia.org/wiki/B_트리