

Assignment 2 Report

Team14 106070038 杜葳葳 106030019 吳岱容

● Implement

■ RTE :

- ◆ 原本 type 只有 READ_ITEM，現在新增 UPDATE_ITEM，Executor 也要新增一個 update 的
- ◆ Type 切換：在 getNextType() 中，給一個 Random 子(介於 1-100 的 int)，然後判斷該值是否大於 ratio(於 vanilladb.properties 中定義)，如果是，執行 update，反之，則為 read

■ JDBC :

- ◆ 於 enum As2BenchTxnType 新增 UPDATE_ITEM(true)
- ◆ 於 class As2BenchJdbcExecutor 新增 case UPDATE_ITEM，會 call 到 UpdateItemTxnJdbcJob
- ◆ 新增 class UpdateItemTxnJdbcJob，產生 random price 的參數(0.0-5.0 的 double)，在 SELECT 的 sql 要多選 i_price，並多下一個 UPDATE 的 sql(要使用 executeUpdate)，UPDATE 前需判斷原本的 price 是否大於 MAX_PRICE，來決定要變成 MIN_PRICE 或是加上 random price
- ◆ 新增 class ParamGenerator

■ SP :

- ◆ 於 As2BenchStoredProcFactory 中新增一個 case UPDATE_ITEM，會 call 到 UpdateItemTxnProc
- ◆ 新增 class UpdateItemTxnProc，類似 JDBC 的 UpdateItemTxnJdbcJob
- ◆ 新增 class UpdateItemProcParamHelper

■ STATISTIC :

- ◆ 產生 csv 的 file，每五秒紀錄一次 time, throughput...，用一個 array 存每次 txn 的 time，然後用 Collections.sort 一大小做排序，用此方式看各個統計量

● CSV report

JDBC(Ratio=50)

time(sec)	throughput	avg_lat(ms)	min(ms)	max(ms)	25th_lat(r)	median_lat	75th_lat(ms)
65	120	41	23	118	28	31	45
70	91	54	25	215	31	42	57
75	116	42	23	126	28	31	47
80	119	42	21	113	28	31	49
85	103	48	24	200	29	40	51
90	110	45	20	163	28	32	52
95	121	41	23	149	28	30	47
100	120	41	23	126	28	29	47
105	120	41	23	139	27	30	48
110	120	41	23	131	28	32	41
115	126	39	16	117	27	30	38

SP(Ratio=50)

time(sec)	throughput	avg_lat(ms)	min(ms)	max(ms)	25th_lat(r)	median_lat	75th_lat(ms)
65	164	30	9	191	19	21	32
70	200	25	13	95	18	19	20
75	203	24	12	79	18	19	21
80	153	32	13	235	19	20	35
85	197	25	8	80	18	19	21
90	206	24	13	95	18	19	20
95	200	25	9	121	19	20	21
100	149	33	8	191	20	26	31
105	147	34	13	188	19	22	35
110	189	26	16	117	19	20	23
115	166	29	17	116	20	23	26

● Experiments

- Environment: Intel(R) Core(TM) i7-7700HQ CPU@2.80GHz, 8.00 GB,128

SSD, Windows10 Home 1903

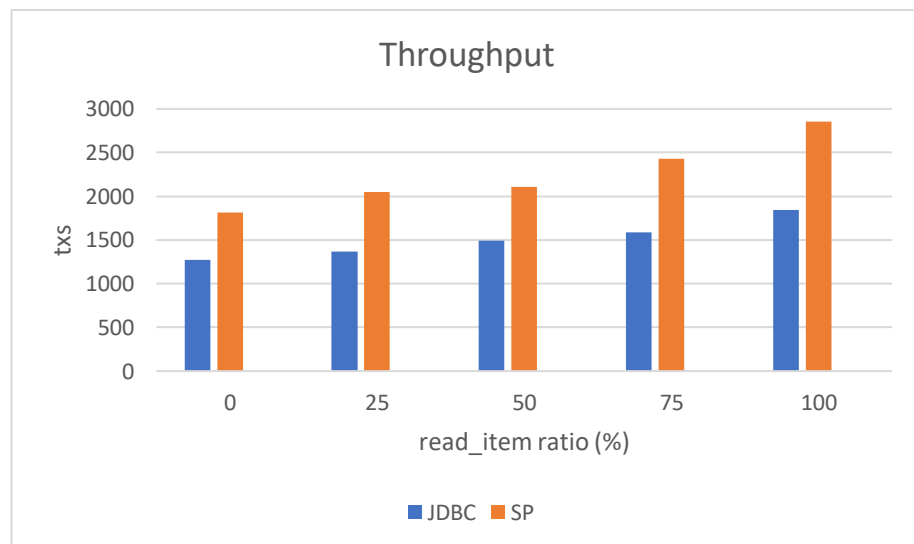
■ Throughput

Ratio \ Mode	0(all update)	25	50	75	100(all read)
JDBC	1269	1369	1493	1585	1847
SP	1813	2049	2109	2429	2854

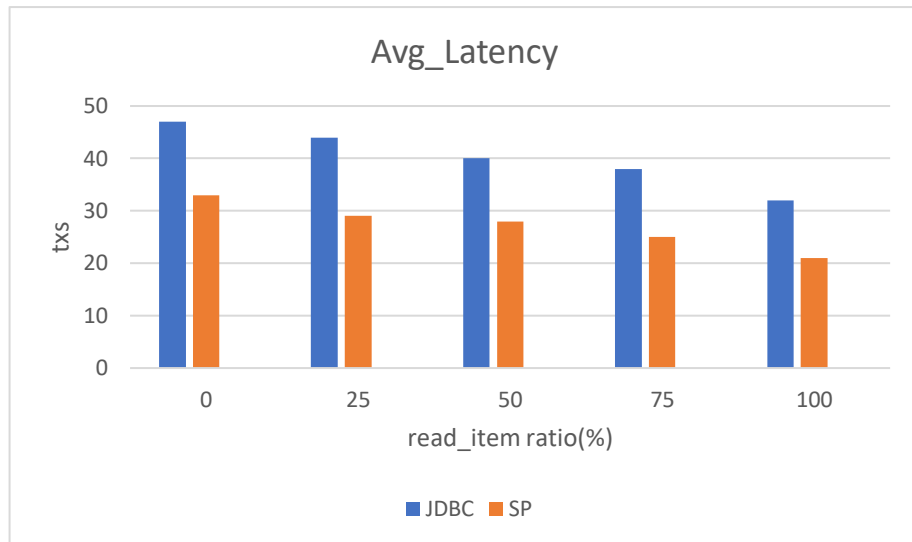
■ Avg Latency

Ratio \ Mode	0	25	50	75	100
JDBC	47	44	40	38	32
SP	33	29	28	25	21

● Analysis and explanation



上圖為 JDBC/SP 和不同 ratio 的 Throughput，可以觀察到 SP 的 Throughput 比 JDBC 高。另外，read_item ratio 愈高，代表 read_item (read) 越多次、update_item (read+write) 較少次，因此 Throughput 會上升。



上圖為 JDBC/SP 和不同 ratio 的 Avg_Latency，可以觀察到 JDBC 的 Avg_Latency 比 SP 高。另外，read_item ratio 愈高，代表 read_item (read) 越多次、update_item (read+write) 較少次，因此 Avg_Latency 會下降。

- 結論

- Throughput：SP>JDBC、READ>UPDATE，因為 READ 是只有讀，UPDATE 包含讀和寫。
- 改變 ratio 時，SP Throughput 的變化較 JDBC 明顯。