

why is p2 commit faster than p1 commit despite more rows and more db hits according to plan for 1st , 12th and last run?

```
// P1 commit

PROFILE
CALL apoc.periodic.commit('MATCH (n:thg)-[:child]->(child:thg)
WHERE n.VGsubgraph<>[] and n.VGsubgraph<> child.VGsubgraph AND any(val in n.VGsubgraph WHERE NOT apoc.coll.contains(child.VGsubgraph,val))
WITH n, child, [val IN n.VGsubgraph WHERE NOT val IN child.VGsubgraph ] AS to_append WHERE to_append <> []
SET child.VGsubgraph=child.VGsubgraph + to_append
WITH count(child) as limit
RETURN limit',
{});

// P2 commit

CALL apoc.periodic.commit('
MATCH (n:thg)-[:child]->(child:thg)
UNWIND n.VGsubgraph AS val
WITH DISTINCT val, child
WHERE NOT val IN child.VGsubgraph
WITH child, COLLECT (val) AS vals
SET child.VGsubgraph = child.VGsubgraph + vals
WITH count(child) AS limit
RETURN limit'
,{});
```

• first run

```
1 // P1
2 PROFILE MATCH (n:thg)-[:child]->(child:thg)
3 WHERE n.VGsubgraph<>[] and n.VGsubgraph<> child.VGsubgraph AND
4 any(val in n.VGsubgraph WHERE NOT apoc.coll.contains(child.VGsubgraph,val))
5 WITH n, child, [val IN n.VGsubgraph WHERE NOT val IN child.VGsubgraph ] AS to_append
6 WHERE to_append <> []
7 SET child.VGsubgraph=child.VGsubgraph + to_append
8 WITH count(distinct child) as limit
9 RETURN limit
```

Plan Table RAW

Planner COST
Runtime PIPELINED
Runtime version 5.11
Batch size 1024

Operator	Id	Details	Estimated Rows	Rows	DB Hits	Memory (Bytes)	Page Cache Hits/Misses	Time (ms)	Pipeline
+ProduceResults	0	limit	1	1	0		0/0	0,007	In Pipe
+EagerAggregation	1	count(DISTINCT child) AS limit	1	1	0	7856			
+SetProperty	2	child.VGsubgraph = child.VGsubgraph + to_append	10810	240	960				
+Eager	3		10810	240	0	78256	1350/0	3,830	Fused in Pipeline 2
+Filter	4	NOT to_append = \$autolist_1	10810	240	0				
+Projection	5	[val IN cache[n.VGsubgraph] WHERE NOT val IN cache[child.VGsubgraph]] AS to_append	12012	240	0	0			
+Eager	6		12012	240	0	40104	0/0	0,100	Fused in Pipeline 1
+Filter	7	any(val IN cache[n.VGsubgraph] WHERE NOT apoc.coll.contains(cache[child.VGsubgraph], val)) AND NOT c ache[n.VGsubgraph] = cache[child.VGsubgraph] AND child:thg	12012	240	996				
+Expand(All)	8	(n)-[anon_0:child]->(child)	33830	252	362				
+Filter	9	NOT cache[n.VGsubgraph] = \$autolist_0	22671	13	95458				
+NodeByLabelScan	10	n:thg	47729	47729	47730	376	60882/0	82,685	Fused in Pipeline 0

Total database accesses: 145906 total allocated memory: 79120

Set 240 properties

Started streaming: 1 record in less than 1ms and completed after 89ms

```
1 // p2
2 PROFILE MATCH (n:thg)-[:child]->(child:thg)
3 UNWIND n.VGsubgraph AS val
4 WITH DISTINCT val, child
5 WHERE NOT val IN child.VGsubgraph
6 WITH child, COLLECT (val) AS vals
7 SET child.VGsubgraph = child.VGsubgraph + vals
8 WITH count(child) AS limit
9 RETURN limit
```

Plan Table RAW

Planner COST
Runtime PIPELINED
Runtime version 5.11
Batch size 1024

Operator	Id	Details	Estimated Rows	Rows	DB Hits	Memory (Bytes)	Page Cache Hits/Misses	Time (ms)	Ordered by	Pipeline
----------	----	---------	----------------	------	---------	----------------	------------------------	-----------	------------	----------

Operator	Id	Details	Estimated Rows	Rows	DB Hits	Memory (Bytes)	Page Cache Hits/Misses	Time (ms)	Pipeline
+ProduceResults	0	limit	1	1	0		0/0	0,015	
+EagerAggregation	1	count(child) AS limit	1	1	0	32	0/0	0,160	In Pipeline 4
+SetProperty	2	child.VGsubgraph = child.VGsubgraph + vals	411	232	696		1043/0	1,830	
+OrderedAggregation	3	child, COLLECT(val) AS vals	411	232	0	1824	0/0	0,844	In Pipeline 3
+Eager	4		168585	240	0	36600	0/0	1,103	In Pipeline 2
+Filter	5	NOT val IN child.VGsubgraph	168585	240	480		650/0	6,388	
+OrderedDistinct	6	child, val	674339	240	0	464	0/0	1,705	In Pipeline 1
+Unwind	7	n.VGsubgraph AS val	709830	240	140120				
+Eager	8		709830	70060	0	1689080	259303/0	120,724	Fused in Pipeline 1
+Filter	9	n:thg	70983	70060	141966				
+Expand(All)	10	(child)←[anon_0:child]-(n)	70983	70983	286885				
+NodeByLabelScan	11	child:thg	47729	47729	47730	376	350162/0	121,064	child ASC Fused in Pipeline 0

Total database accesses: 617877 total allocated memory: 1690272

Set 232 properties

Started streaming 1 record in less than 1ms and completed after 270ms.

- after 12 run

```

1 // P1
2 PROFILE MATCH (n:thg)-[:child]→(child:thg)
3 WHERE n.VGsubgraph <> [] and n.VGsubgraph <> child.VGsubgraph AND
4 any(val in n.VGsubgraph WHERE NOT apoc.coll.contains(child.VGsubgraph, val))
5 WITH n, child, [val IN n.VGsubgraph WHERE NOT val IN child.VGsubgraph ] AS to_append
6 WHERE to_append <> []
7 SET child.VGsubgraph=child.VGsubgraph + to_append
8 WITH count(distinct child) as limit
9 RETURN limit

```

Planner COST

Runtime PIPELINED

Runtime version 5.11

Batch size 1024

Operator	Id	Details	Estimated Rows	Rows	DB Hits	Memory (Bytes)	Page Cache Hits/Misses	Time (ms)	Pipeline
+ProduceResults	0	limit	1	1	0		0/0	0,021	In Pipe
+EagerAggregation	1	count(DISTINCT child) AS limit	1	1	0	177000			
+SetProperty	2	child.VGsubgraph = child.VGsubgraph + to_append	10810	6623	26492				
+Eager	3		10810	6623	0	16787556	32315/0	72,308	Fused in Pipeline 2
+Filter	4	NOT to_append = \$autolist_1	10810	6623	0				
+Projection	5	[val IN cache[n.VGsubgraph] WHERE NOT val IN cache[child.VGsubgraph]] AS to_append	12012	6623	0				
+Eager	6		12012	6623	0	13943864	0/0	53,049	Fused in Pipeline 1
+Filter	7	any(val IN cache[n.VGsubgraph] WHERE NOT apoc.coll.contains(cache[child.VGsubgraph], val)) AND NOT c ache[n.VGsubgraph] = cache[child.VGsubgraph] AND child:thg	12012	6623	135918				
+Expand(All)	8	(n)-[anon_0:child]→(child)	33830	60892	254795				
+Filter	9	NOT cache[n.VGsubgraph] = \$autolist_0	22671	40434	95468				
+NodeByLabelScan	10	n:thg	47729	47729	47730	376	530785/0	914,649	Fused in Pipeline 0

Total database accesses: 560403, total allocated memory: 16788420

Set 6623 properties

Started streaming 1 record after 69ms and completed after 1115ms.

```

// p2
PROFILE MATCH (n:thg)-[:child]→(child:thg)
UNWIND n.VGsubgraph AS val
WITH DISTINCT val, child
WHERE NOT val IN child.VGsubgraph
WITH child, COLLECT (val) AS vals
SET child.VGsubgraph = child.VGsubgraph + vals
WITH count(child) AS limit
RETURN limit

```

Planner COST

Runtime PIPELINED

Runtime version 5.11

Batch size 1024

Operator	Id	Details	Estimated Rows	Rows	DB Hits	Memory (Bytes)	Page Cache Hits/Misses	Time (ms)	Ordered by	Pipeline
----------	----	---------	----------------	------	---------	----------------	------------------------	-----------	------------	----------

Operator	Id	Details	Estimated Rows	Rows	DB Hits	Memory (Bytes)	Page Cache Hits/Misses	Time (ms)	Pipeline
+ProduceResults	0	limit	1	1	0		0/0	0,049	
+EagerAggregation	1	count(child) AS limit	1	1	0	32	0/0	0,843	In Pipeline 4
+SetProperty	2	child.VGsubgraph = child.VGsubgraph + vals	411	6003	18009		20279/0	31,324	
+OrderedAggregation	3	child, COLLECT(val) AS vals	411	6003	0	6408	0/0	4,147	In Pipeline 3
+Eager	4		168585	7591	0	1182660	0/0	6,248	In Pipeline 2
+Filter	5	NOT val IN child.VGsubgraph	168585	7591	362612	0	196484/0	313,920	
+OrderedDistinct	6	child, val	674339	181306	0	1360	0/0	49,211	In Pipeline 1
+Unwind	7	n.VGsubgraph AS val	709830	257430	140120				
+Eager	8		709830	70060	0	1689080	259818/0	183,485	Fused in Pipeline 1
+Filter	9	n:thg	70983	70060	141966				
+Expand(All)	10	(child)←[anon_0:child]-(n)	70983	70983	286885				
+NodeByLabelScan	11	child:thg	47729	47729	47730	376	350162/0	85,191	child ASC Fused in Pipeline 0

Total database accesses: 997322, total allocated memory: 1700968

Set 6003 properties

Started stream

- after 24 runs // p1

Operator	Id	Details	Estimated Rows	Rows	DB Hits	Memory (Bytes)	Page Cache Hits/Misses	Time (ms)	Pipeline
+ProduceResults	0	limit	1	1	0		0/0	0,006	In Pipeline 4
+EagerAggregation	1	count(DISTINCT child) AS limit	1	1	0	248			
+SetProperty	2	child.VGsubgraph = child.VGsubgraph + to_append	10810	1	4				
+Eager	3		10810	1	0	4924	3/0	0,100	Fused in Pipeline 2
+Filter	4	NOT to_append = \$autolist_1	10810	1	0				
+Projection	5	[val IN cache[n.VGsubgraph] WHERE NOT val IN cache[child.VGsubgraph]] AS to_append	12012	1	0	0			
+Eager	6		12012	1	0	4560	0/0	0,036	Fused in Pipeline 1
+Filter	7	any(val IN cache[n.VGsubgraph] WHERE NOT apoc.coll.contains(cache[child.VGsubgraph], val)) AND NOT cache[n.VGsubgraph] = cache[child.VGsubgraph] AND child:thg	12012	1	122674				
+Expand(All)	8	(n)-[anon_0:child]→(child)	33830	60892	254795				
+Filter	9	NOT cache[n.VGsubgraph] = \$autolist_0	22671	40434	95458				
+NodeByLabelScan	10	n:thg	47729	47729	47730	376	532809/0	1280,200	Fused in Pipeline 0

Total database accesses: 520661, total allocated memory: 5788

Set 1 property

Started streaming 1 record after 38ms and completed after 1319ms.

// p2

Planner COST

Runtime PIPELINED

Runtime version 5.11

Batch size 1024

Operator	Id	Details	Estimated Rows	Rows	DB Hits	Memory (Bytes)	Page Cache Hits/Misses	Time (ms)	Ordered by	Pipeline
+ProduceResults	0	limit	1	1	0		0/0	0,006		
+EagerAggregation	1	count(DISTINCT child) AS limit	1	1	0	32	0/0	0,179		In Pipeline 4
+SetProperty	2	child.VGsubgraph = child.VGsubgraph + vals	411	1	3		4/0	0,051		
+OrderedAggregation	3	child, COLLECT(val) AS vals	411	1	0	772	0/0	0,013		In Pipeline 3
+Eager	4		168585	1	0	540	0/0	0,873		In Pipeline 2
+Filter	5	NOT val IN child.VGsubgraph	168585	1	408276	0	218057/0	380,580		
+OrderedDistinct	6	child, val	674339	204138	0	1440	0/0	39,812		In Pipeline 1
+Unwind	7	n.VGsubgraph AS val	709830	295091	140120					
+Eager	8		709830	70060	0	1689080	260084/0	159,779		Fused in Pipeline 1
+Filter	9	n:thg	70983	70060	141966					
+Expand(All)	10	(child)←[anon_0:child]-(n)	70983	70983	286885					
+NodeByLabelScan	11	child:thg	47729	47729	47730	376	350162/0	80,834	child ASC	Fused in Pipeline 0

Total database accesses: 1024980, total allocated memory: 1691464

Set 1 property

Started streaming 1 record after 102ms and completed after 775ms.

- last run // p1

Operator	Id	Details	Estimated Rows	Rows	DB Hits	Memory (Bytes)	Page Cache Hits/Misses	Time (ms)	Pipeline
+ProduceResults	0	limit	1	1	0		0/0	0,004	In Pipeline 4
+EagerAggregation	1	count(DISTINCT child) AS limit	1	1	0	224			
+SetProperty	2	child.VGsubgraph = child.VGsubgraph + to_append	10810	0	0				
+Eager	3		10810	0	0	272	0/0	0,015	Fused in Pipeline 2
+Filter	4	NOT to_append = \$autolist_1	10810	0	0				
+Projection	5	[val IN cache[n.VGsubgraph] WHERE NOT val IN cache[child.VGsubgraph]] AS to_append	12012	0	0				
+Eager	6		12012	0	0	272	0/0	0,000	Fused in Pipeline 1
+Filter	7	any(val IN cache[n.VGsubgraph] WHERE NOT apoc.coll.contains(cache[child.VGsubgraph], val)) AND NOT cache[n.VGsubgraph] = cache[child.VGsubgraph] AND child:thg	12012	0	122672				
+Expand(All)	8	(n)-[anon_0:child]→(child)	33830	60892	254795				
+Filter	9	NOT cache[n.VGsubgraph] = \$autolist_0	22671	40434	95458				
+NodeByLabelScan	10	n:thg	47729	47729	47730	376	532809/0	1165,712	Fused in Pipeline 0

Total database accesses: 520655 total allocated memory: 1256

Started streaming 1 record after 2ms and completed after 1168ms.

// p2

Planner COST

Runtime PIPELINED

Runtime version 5.11

Batch size 1024

Operator	Id	Details	Estimated Rows	Rows	DB Hits	Memory (Bytes)	Page Cache Hits/Misses	Time (ms)	Ordered by	Pipeline
+ProduceResults	0	limit	1	1	0		0/0	0,005		
+EagerAggregation	1	count(child) AS limit	1	1	0	32	0/0	0,067		In Pipeline 0
+SetProperty	2	child.VGsubgraph = child.VGsubgraph + vals	411	0	0		0/0	0,004		
+OrderedAggregation	3	child, COLLECT(val) AS vals	411	0	0	520	0/0	0,016		In Pipeline 3
+Eager	4		168585	0	0	272	0/0	1,004		In Pipeline 2
+Filter	5	NOT val IN child.VGsubgraph	168585	0	408276	0	218074/0	395,157		
+OrderedDistinct	6	child, val	674339	204138	0	1440	0/0	41,517		In Pipeline 1
+Unwind	7	n.VGsubgraph AS val	709830	295091	140120					
+Eager	8		709830	70060	0	1689080	260084/0	169,124		Fused in Pipeline 1
+Filter	9	n:thg	70983	70060	141966					
+Expand(All)	10	(child)←[anon_0:child]-(n)	70983	70983	286885					
+NodeByLabelScan	11	child:thg	47729	47729	47730	376	350162/0	91,204	child ASC	Fused in Pipeline 0

Total database accesses: 1024977, total allocated memory: 1691464

Started streaming 1 record in less than 1ms and completed after 711ms.