Apache Flink

基于Log的通用增量Checkpoint

俞航翔/Apache Flink Contributor 2022-9-24

Apache Flink meetup 北京站

CONTENT

目录 >>

01 /
Checkpoint性能优化之路

02 /
解析Changelog

03 /
一览State/Checkpoint优化

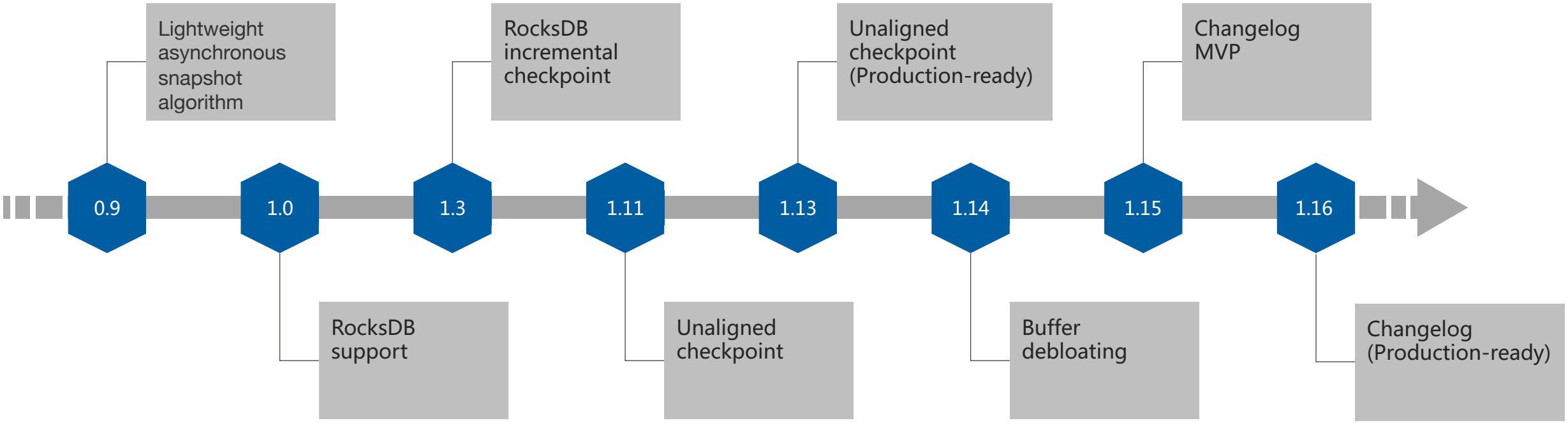
04 /
总结

01 Checkpoint性能优化之路

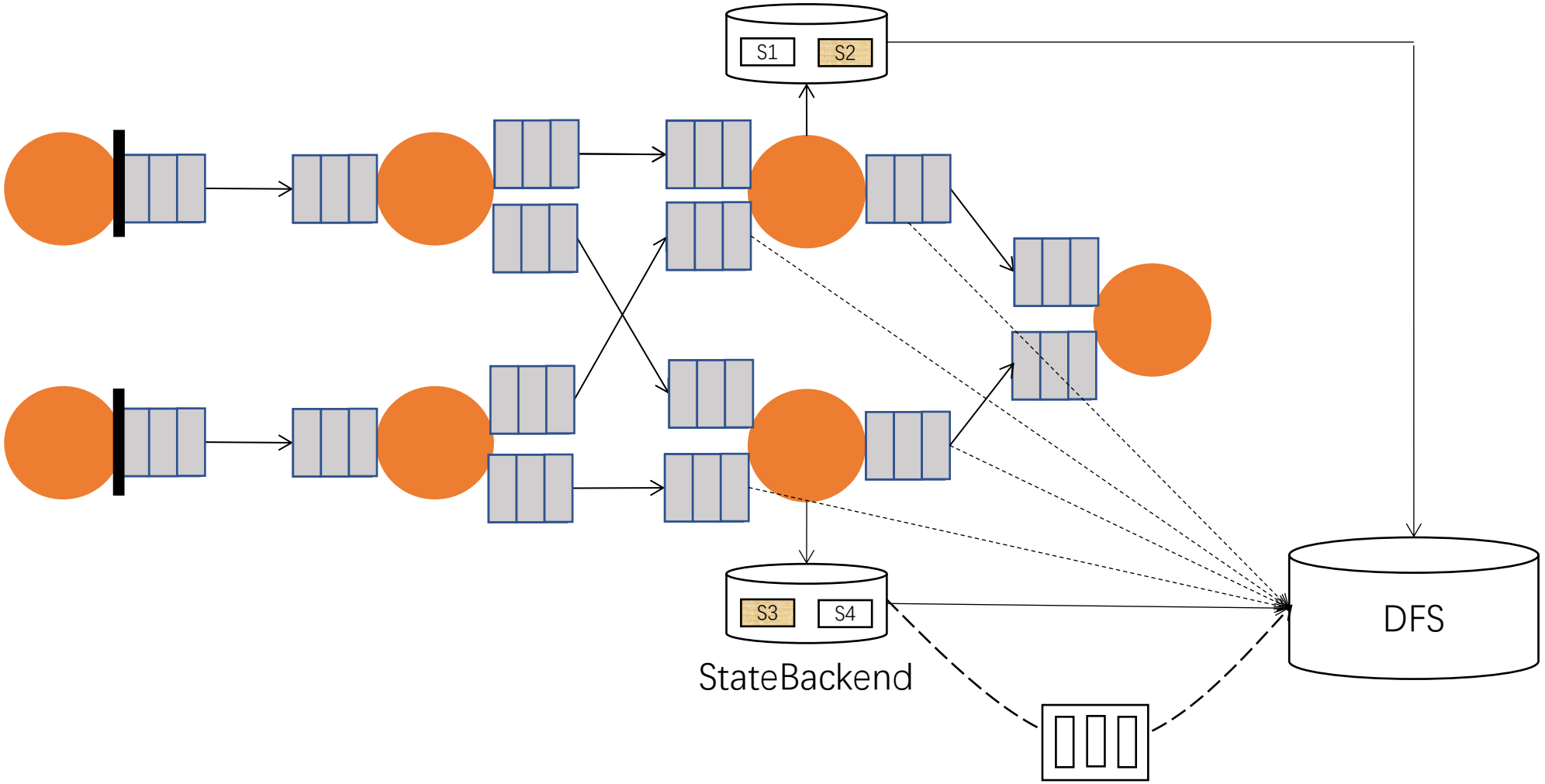
What's Checkpoint

State Persistence	Owned by Flink	Lightweight/Fast
Fault Tolerance	Native Format	Fast to Restore from

Checkpoint Performance Improving



Checkpoint链路



Checkpoint Metrics

–	8	COMPLETED	16/16	2022-01-23 22:19:28	2022-01-23 22:19:28	132ms	213 KB	650 KB	5.80 KB (0 B)

Checkpoint Detail: **Path:** file:/tmp/chk/f67778208458872c17dc7a3ce979ebd2/chk-8 **Discarded:** true **Checkpoint Type:** aligned checkpoint

Operators:

	Name	Acknowledged	Latest Acknowledgment	End to End Duration	Checkpointed Data Size	Full Checkpoint Data Size	Processed (persisted) in-flight data
+	Source: Custom Source	8/8 (100%)	2022-01-23 22:19:28	13ms	0 B	0 B	0 B (0 B)
–	Flat Map -> Sink: Print to Std. Out	8/8 (100%)	2022-01-23 22:19:28	132ms	213 KB	650 KB	5.80 KB (0 B)

SubTasks:

	End to End Duration	Checkpointed Data Size	Full Checkpoint Data Size	Sync Duration	Async Duration	Processed (persisted) Data	Alignment Duration	Start Delay
Minimum	43ms	25.8 KB	79.6 KB	13ms	16ms	602 B (0 B)	3ms	6ms
Average	86ms	26.7 KB	81.3 KB	21ms	52ms	742 B (0 B)	3ms	6ms
Maximum	132ms	27.4 KB	83.2 KB	30ms	90ms	952 B (0 B)	3ms	6ms

ID	Acknowledged	End to End Duration	Checkpointed Data Size	Full Checkpoint Data Size	Sync Duration	Async Duration	Processed (persisted) Data	Alignment Duration	Start Delay	Unaligned Checkpoint
0	2022-01-23 22:19:28	81ms	26.4 KB	79.9 KB	19ms	49ms	658 B (0 B)	3ms	6ms	false
1	2022-01-23 22:19:28	106ms	25.8 KB	80.8 KB	15ms	78ms	742 B (0 B)	3ms	6ms	false
2	2022-01-23 22:19:28	43ms	26.6 KB	80.3 KB	13ms	16ms	756 B (0 B)	3ms	6ms	false
3	2022-01-23 22:19:28	119ms	26.5 KB	81.3 KB	30ms	76ms	840 B (0 B)	3ms	6ms	false
4	2022-01-23 22:19:28	54ms	27.2 KB	82.5 KB	25ms	17ms	602 B (0 B)	3ms	6ms	false
5	2022-01-23 22:19:28	93ms	26.7 KB	79.6 KB	18ms	63ms	952 B (0 B)	3ms	6ms	false
6	2022-01-23 22:19:28	132ms	27.4 KB	83.2 KB	28ms	90ms	686 B (0 B)	3ms	6ms	false
7	2022-01-23 22:19:28	67ms	26.8 KB	82.7 KB	22ms	32ms	700 B (0 B)	3ms	6ms	false

02 解析Changelog

核心目标

更稳定的Checkpoint

大幅减少Checkpoint duration突增，
减少CPU抖动，平稳网络带宽

更快速的Checkpoint

Checkpoint期间上传相对固定的增量，
秒级/亚秒级完成Checkpoint

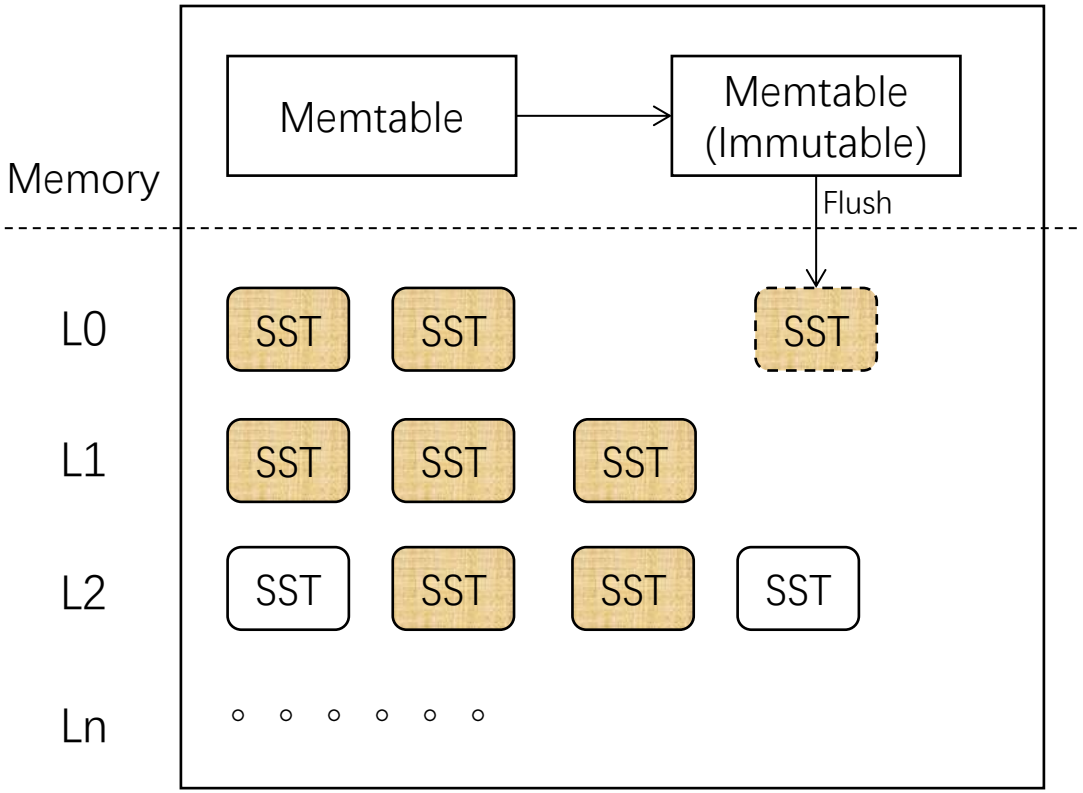
更小的端到端延迟

Checkpoint越快，
Transactional sinks的提交可以越频繁

更少的数据回追

通过设置更小的Checkpoint Interval，进一步提供更快速的failover过程

RocksDB Incremental Checkpoint

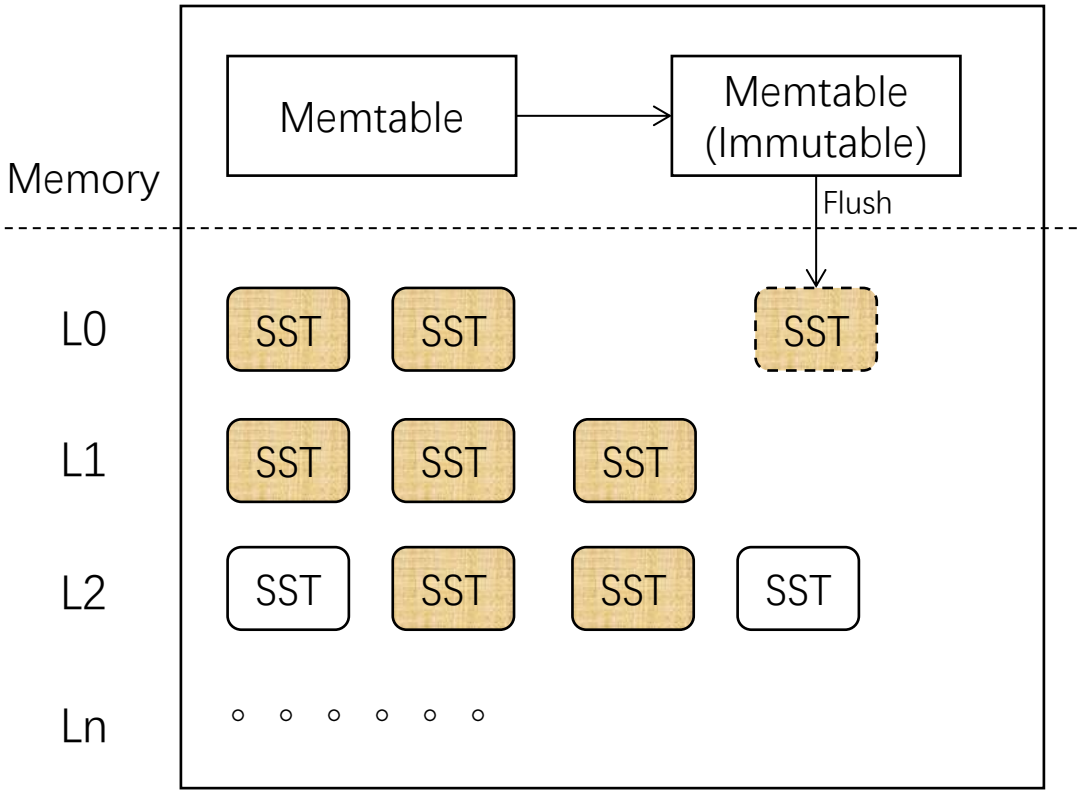


Column Family X
(State X)

Sync:
1. Force flush memtables
2. Local checkpoint(Hard link SST Files)

Async:
1. Write metas
2. Upload files

RocksDB Incremental Checkpoint



Column Family X
(State X)

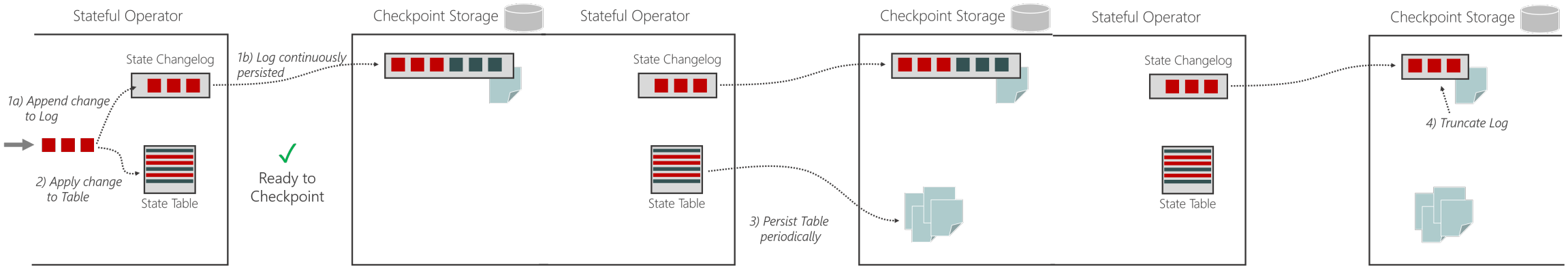
Flush的触发时机：
1. 数据量达到阈值
2. CP的同步阶段

Compaction与Checkpoint耦合的问题：
1. Flush可能会触发多层Level Compaction，进一步导致大量文件需要上传
2. 大规模作业，每次cp都可能因为某个subtask异步时间过长而导致cp e2e duration变长

Changelog Incremental Checkpoint

术语	描述
State table	本地状态数据读写结构，如RocksDB
Materialization	State table的持久化过程，目前会定时触发，在完成一次成功的Materialization后会Truncate Changelog
DSTL	Durable Short-term Log，Changelog的存储组件
Changelog	以Append-only Log形式存储的状态记录

Changelog Incremental Checkpoint



Read/Write

- 状态写入
  - State table
  - DSTL
- 状态读取
  - State table

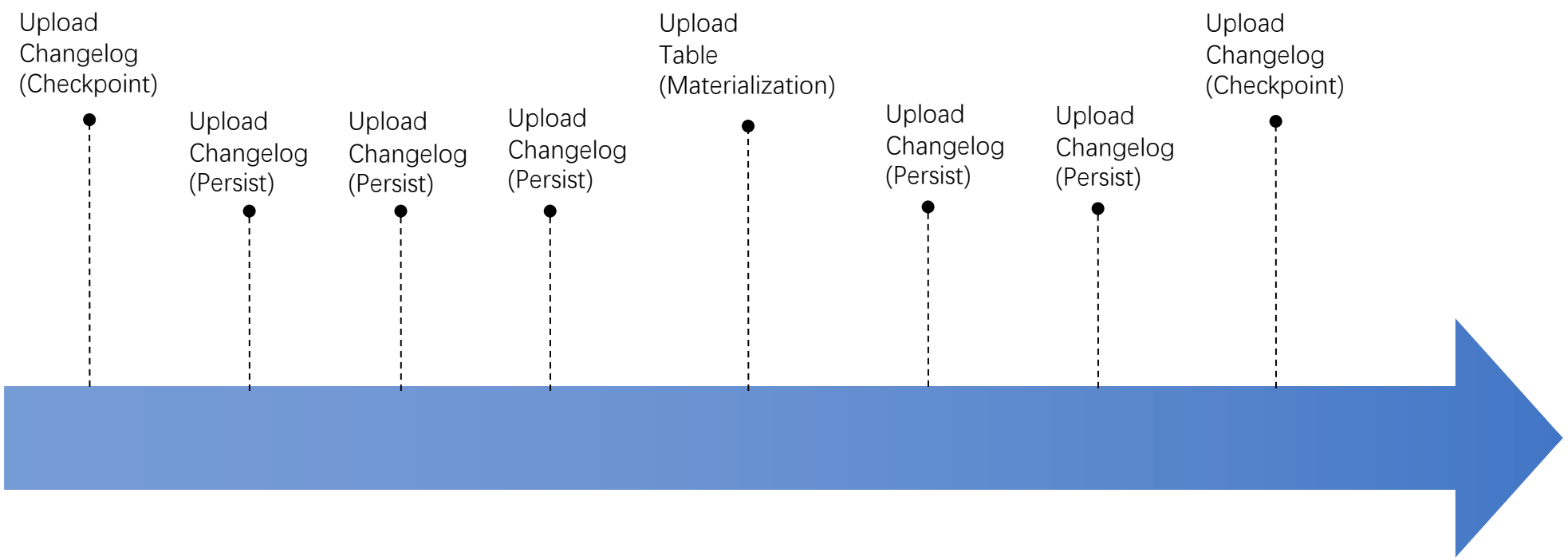
Checkpoint

- 定时Materialize & Truncate
- 定期Persist Changelog
- ChangelogStateHandle = Materialization part + Changelog part

Restore

- Restore State table
- Apply Changelog

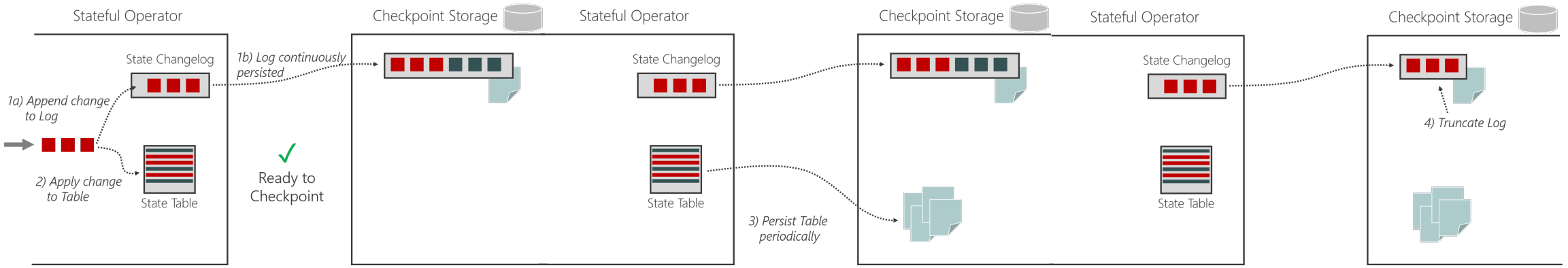
Changelog Incremental Checkpoint



常用参数

参数	默认值	含义
state.backend.changelog.enabled	false	是否开启Changelog，1.16中支持了该参数的兼容性
state.backend.changelog.periodic-materialize.interval	10mins	Materialization的间隔
state.backend.changelog.storage	memory	Changelog部分的存储介质： • memory：存储在内存中，推荐测试使用 • filesystem： 存储在dfs中，推荐生产环境使用
dstl.dfs.base-path	<none>	Changelog部分在dfs上存储的路径，当storage设置为filesystem时需要设置
dstl.dfs.compression.enabled	false	Changelog部分是否开启压缩，可以在性能和空间上取舍

Trade-offs



更稳定且快速的Checkpoint

VS

额外的存储空间开销
Changelog部分额外的恢复开销
额外的性能开销

Benchmark Config

Config	Value
Parallelism	10
State backend	RocksDB(incremental)
Storage	OSS
Checkpoint interval	1s
Materialization interval	3mins
Source Rate	10K/s

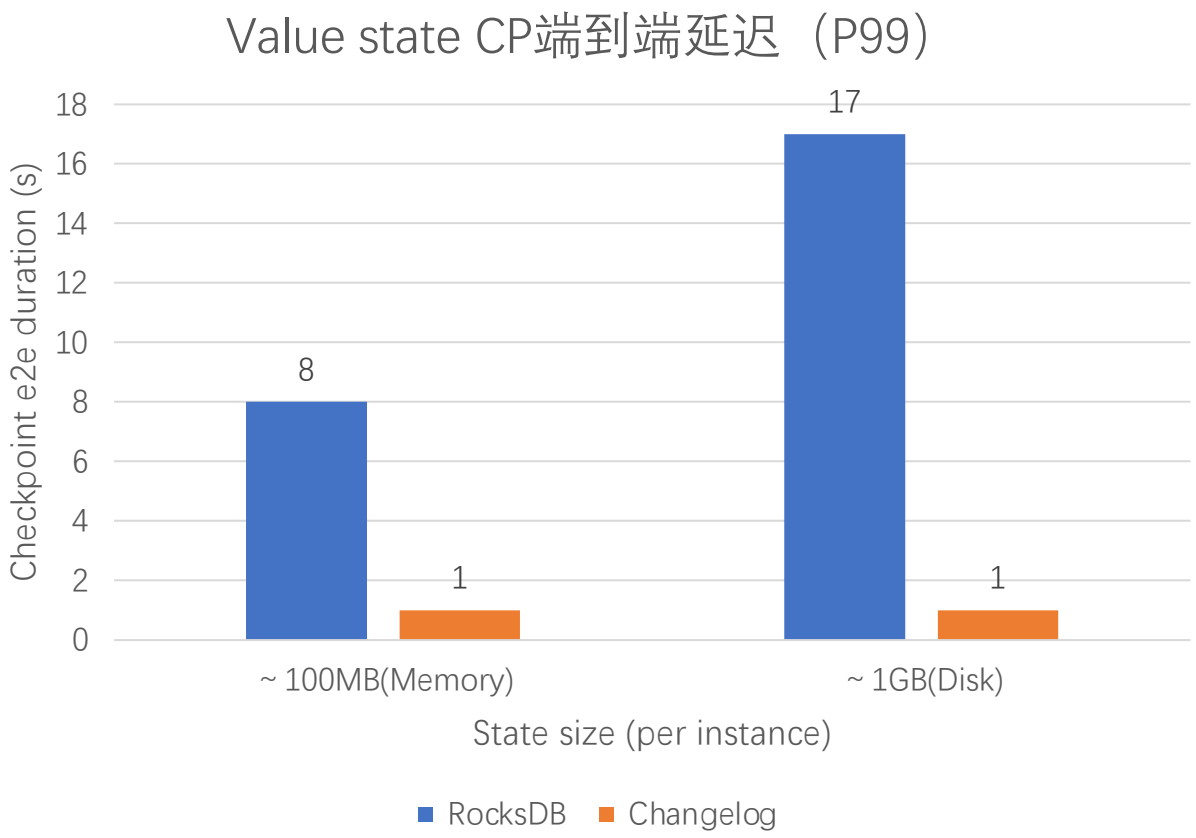
Benchmark Result – 更稳定的Checkpoint

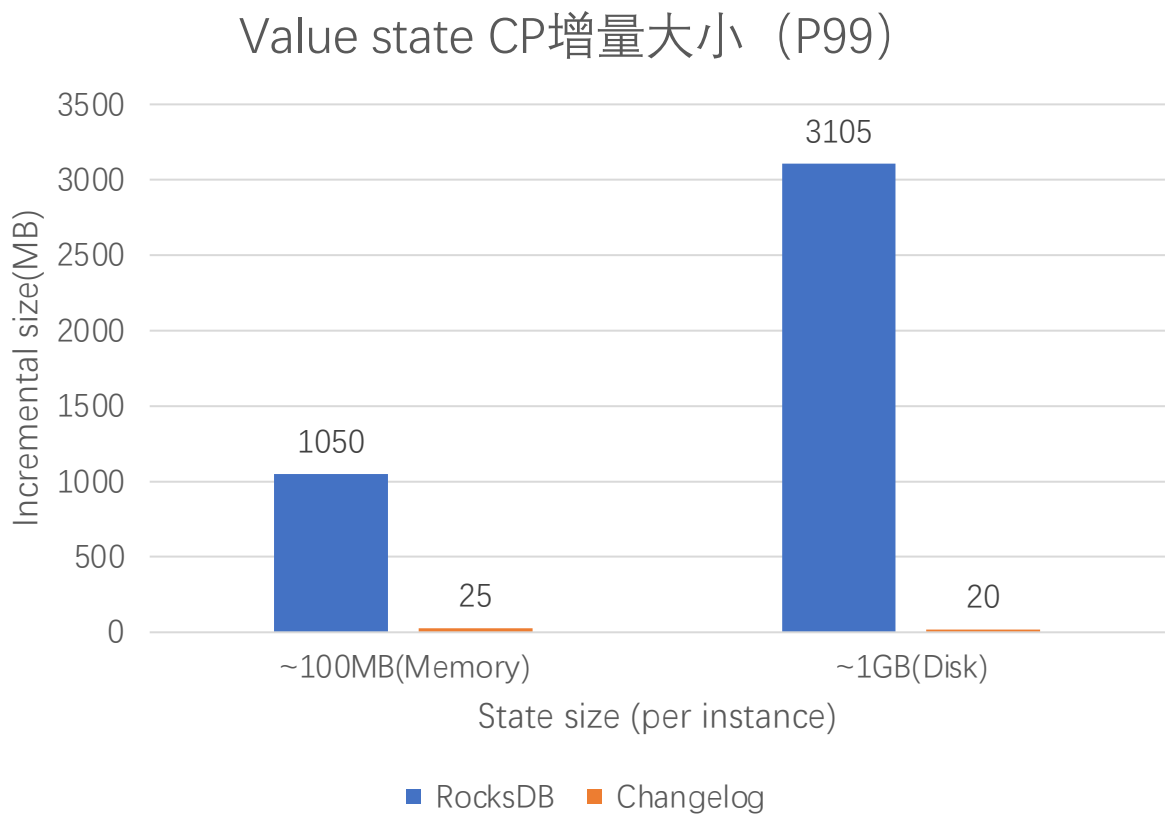
ID	Status	Acknowledged	Trigger Time	Latest Acknowledgement	End to End Duration	Checkpointed Data Size
175	IN_PROGRESS	17/20 (85%)	2022-09-23 13:24:11	2022-09-23 13:24:13	1s	120.28 MB
174	COMPLETED	20/20	2022-09-23 13:23:54	2022-09-23 13:24:10	16s	3.26 GB
173	COMPLETED	20/20	2022-09-23 13:23:44	2022-09-23 13:23:53	8s	69.22 MB
172	COMPLETED	20/20	2022-09-23 13:23:42	2022-09-23 13:23:43	378ms	22.83 MB
171	COMPLETED	20/20	2022-09-23 13:23:40	2022-09-23 13:23:41	1s	133.12 MB
170	COMPLETED	20/20	2022-09-23 13:23:27	2022-09-23 13:23:39	12s	2.63 GB
169	COMPLETED	20/20	2022-09-23 13:23:19	2022-09-23 13:23:25	6s	73.1 MB
168	COMPLETED	20/20	2022-09-23 13:23:17	2022-09-23 13:23:18	921ms	96.09 MB
167	COMPLETED	20/20	2022-09-23 13:23:07	2022-09-23 13:23:16	8s	1.53 GB
166	COMPLETED	20/20	2022-09-23 13:22:54	2022-09-23 13:23:06	12s	2.04 GB
165	COMPLETED	20/20	2022-09-23 13:22:47	2022-09-23 13:22:53	5s	72.35 MB
164	COMPLETED	20/20	2022-09-23 13:22:44	2022-09-23 13:22:45	943ms	93.75 MB
163	COMPLETED	20/20	2022-09-23 13:22:35	2022-09-23 13:22:43	8s	1.17 GB
162	COMPLETED	20/20	2022-09-23 13:22:24	2022-09-23 13:22:34	9s	1.47 GB
161	COMPLETED	20/20	2022-09-23 13:22:19	2022-09-23 13:22:23	3s	67.06 MB

ID	Status	Acknowledged	Trigger Time	Latest Acknowledgement	End to End Duration	Checkpointed Data Size
154	IN_PROGRESS	12/20 (60%)	2022-09-23 13:32:35	2022-09-23 13:32:35	425ms	3.2 MB
153	COMPLETED	20/20	2022-09-23 13:32:33	2022-09-23 13:32:33	266ms	15.05 MB
152	COMPLETED	20/20	2022-09-23 13:32:32	2022-09-23 13:32:32	311ms	23.85 MB
151	COMPLETED	20/20	2022-09-23 13:32:29	2022-09-23 13:32:30	1s	29.98 MB
150	COMPLETED	20/20	2022-09-23 13:32:26	2022-09-23 13:32:28	1s	17.41 MB
149	COMPLETED	20/20	2022-09-23 13:32:24	2022-09-23 13:32:25	498ms	15.62 MB
148	COMPLETED	20/20	2022-09-23 13:32:23	2022-09-23 13:32:23	262ms	16.25 MB
147	COMPLETED	20/20	2022-09-23 13:32:21	2022-09-23 13:32:22	371ms	16.83 MB
146	COMPLETED	20/20	2022-09-23 13:32:19	2022-09-23 13:32:20	417ms	18.8 MB
145	COMPLETED	20/20	2022-09-23 13:32:18	2022-09-23 13:32:18	623ms	27.98 MB
144	COMPLETED	20/20	2022-09-23 13:32:15	2022-09-23 13:32:16	1s	20.41 MB
143	COMPLETED	20/20	2022-09-23 13:32:13	2022-09-23 13:32:14	873ms	16.26 MB
142	COMPLETED	20/20	2022-09-23 13:32:11	2022-09-23 13:32:11	386ms	15.94 MB
141	COMPLETED	20/20	2022-09-23 13:32:09	2022-09-23 13:32:10	307ms	14.25 MB
140	COMPLETED	20/20	2022-09-23 13:32:08	2022-09-23 13:32:08	225ms	15.04 MB
139	COMPLETED	20/20	2022-09-23 13:32:06	2022-09-23 13:32:07	269ms	14.45 MB

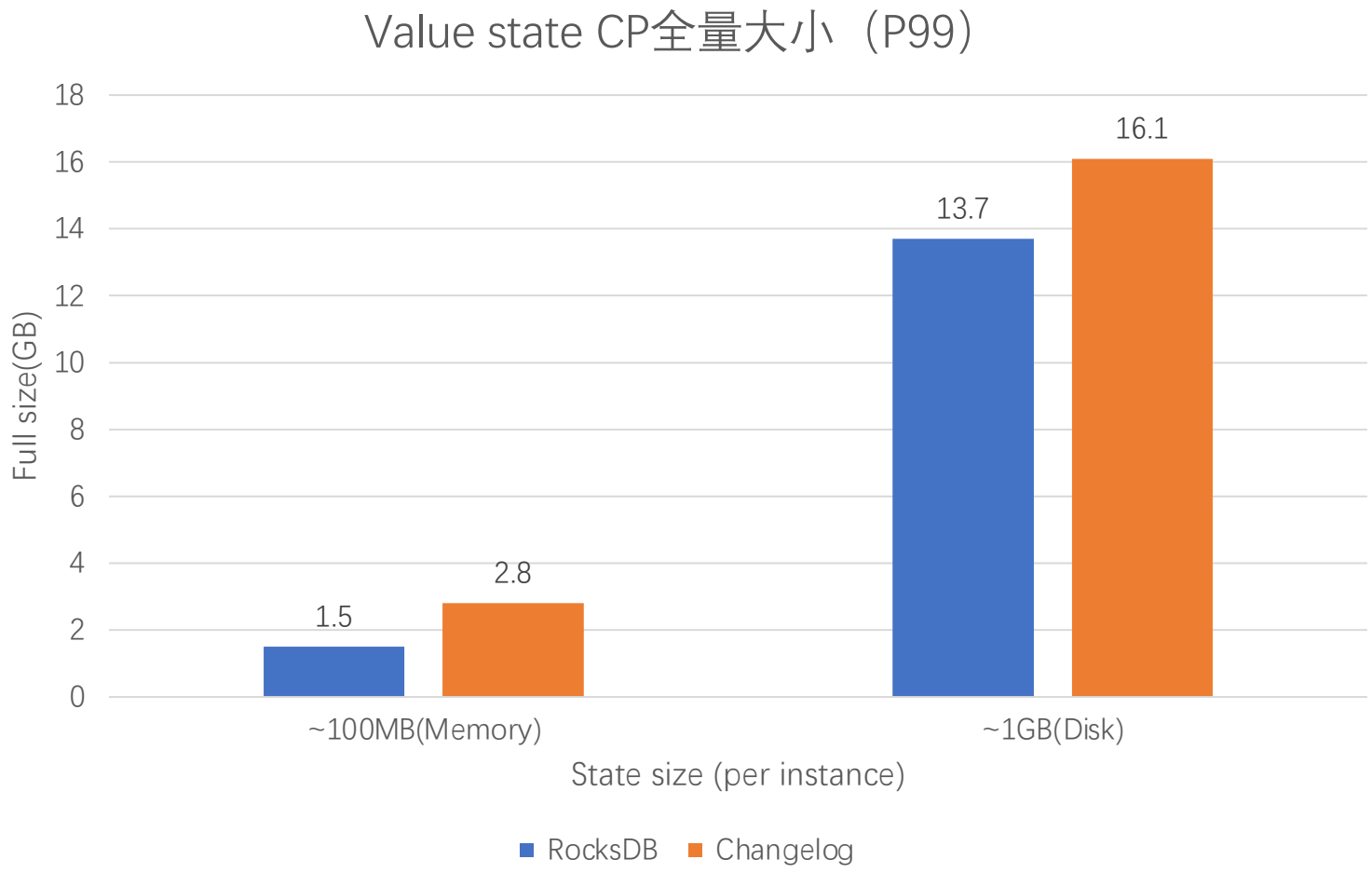
RocksDB **VS** Changelog

Benchmark Result – 更快速的Checkpoint

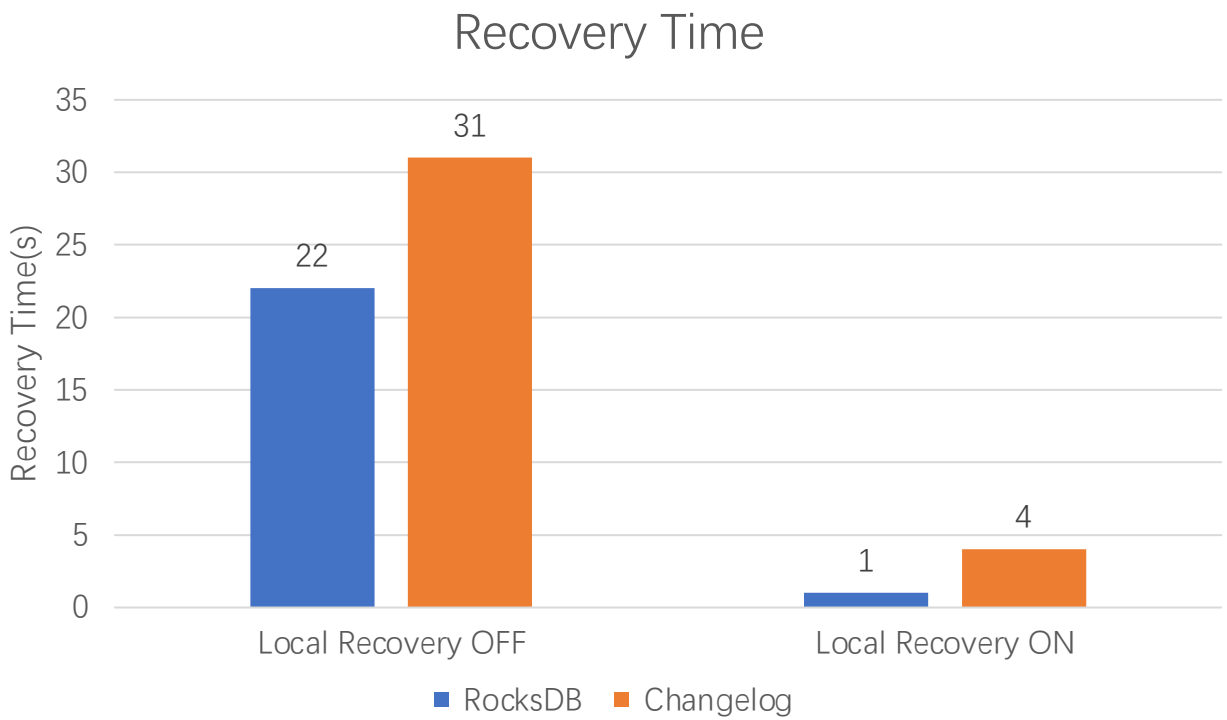




Benchmark Result – 额外空间消耗

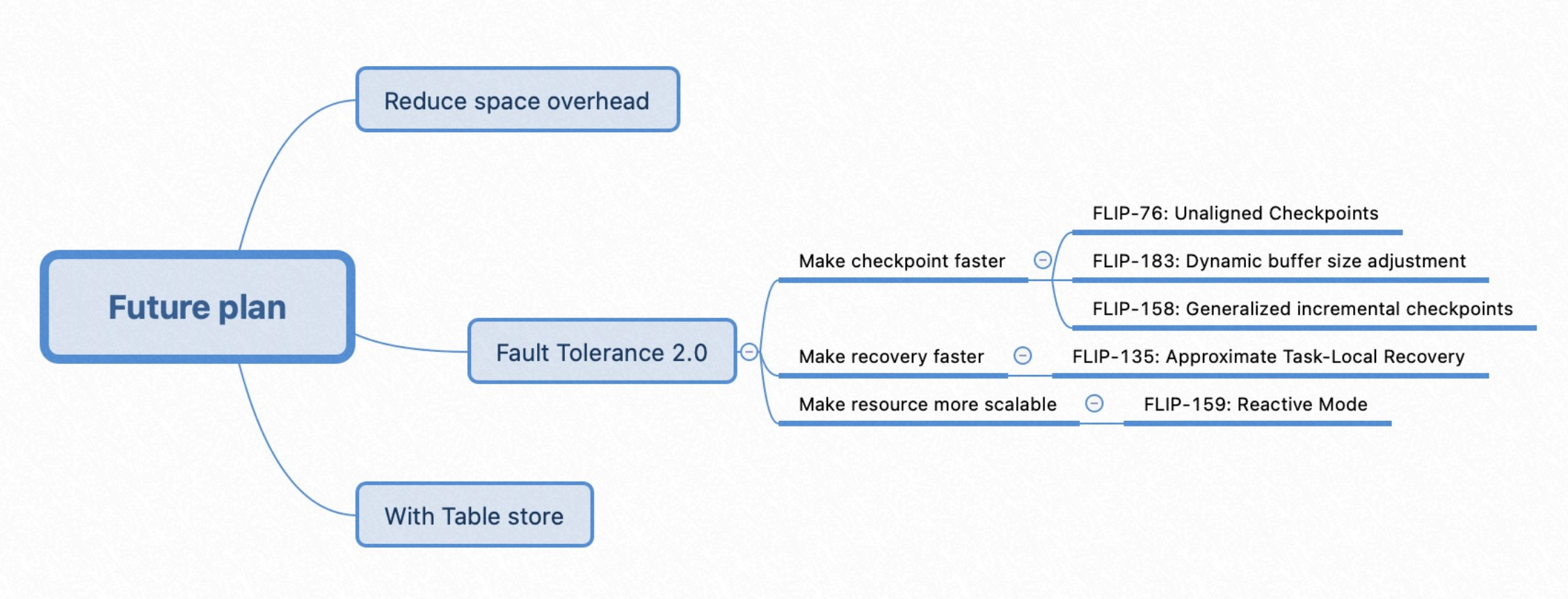


Benchmark Result – 额外恢复开销 & 性能开销



	Changelog vs RocksDB 极限TPS
Local Recovery OFF	-10%
Local Recovery ON	-10%

未来规划


Future plan
Reduce space overhead
Fault Tolerance 2.0
Make checkpoint faster
FLIP-76: Unaligned Checkpoints
FLIP-183: Dynamic buffer size adjustment
FLIP-158: Generalized incremental checkpoints
Make recovery faster
FLIP-135: Approximate Task-Local Recovery
Make resource more scalable
FLIP-159: Reactive Mode
With Table store

03 一览State/Checkpoint优化

1.16 State/Checkpoint优化项

可用性提升

- 【FLINK-24783】提升状态后端监控和可用性
- 【FLINK-27251】提升AC和UC切换可用性

性能优化

- 【FLINK-28038】基于deleteRange提升RocksDB Rescale性能
- 【FLINK-27530】基于overdraft buffer提升Checkpoint性能

04 总结

总结

Checkpoint性能优化之路

- Async checkpoint
- Incremental checkpoint
- Unaligned checkpoint
- Buffer debloating
- ChangelogStateBackend

解析Changelog

- RocksDB incremental checkpoint机制
- Changelog incremental checkpoint机制
- Changelog使用方式
- Changelog trade-offs及Benchmark结果
- Changelog未来规划

一览State/Checkpoint优化

- 可用性优化
- 性能优化

Thanks

Apache Flink meetup 北京站