

腾讯社交广告  
Tencent Social Ads

广点通

CSDN  
CCTC 中国云计算技术大会  
Cloud Computing Technology Conference

中国 Spark 技术峰会 2016

# Spark Streaming 在腾讯广点通的应用

# 关于腾讯广点通

## 腾讯社交广告 (Tencent Social Ads)

- 基于腾讯社交网络体系的广告平台

## 流量覆盖

- QQ 客户端、手机 QQ
- 微信
- QQ 音乐客户端、腾讯新闻客户端

## 主动型效果广告

- 有效实现更加智能的广告匹配和高效的广告资源利用

## 广告形式

- Banner 广告、插屏广告、开屏广告、信息流广告等



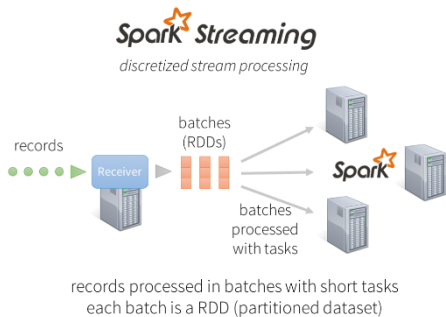
# 广点通与 Spark Streaming

Spark 0.x ~ 1.1

- 调研、试验

Spark 1.2 ~ 1.6

- 使用
- 1.2 加入了 driver 故障恢复



## 社区贡献

### Spark Streaming 源码解析系列

「腾讯广点通」技术团队荣誉出品

本系列内容适用范围:

- \* 2016.02.25 update, Spark 2.0 全系列 √ (2.0.0-SNAPSHOT 尚未正式发布)
- \* 2016.03.10 update, Spark 1.6 全系列 √ (1.6.0, 1.6.1)
- \* 2015.11.09 update, Spark 1.5 全系列 √ (1.5.0, 1.5.1, 1.5.2)
- \* 2015.07.15 update, Spark 1.4 全系列 √ (1.4.0, 1.4.1)

- 概述
  - 0.1 Spark Streaming 实现思路与模块概述
- 模块 1: DAG 静态定义
  - 1.1 DStream, DStreamGraph 详解
  - 1.2 DStream 生成 RDD 实例详解
- 模块 2: Job 动态生成
  - 2.1 JobScheduler, Job, JobSet 详解
  - 2.2 JobGenerator 详解
- 模块 3: 数据产生与导入
  - 3.1 Receiver 分发详解
  - 3.2 Receiver, ReceiverSupervisor, BlockGenerator, ReceivedBlockHandler 详解
  - 3.3 ReceiverTracker, ReceivedBlockTracker 详解
- 模块 4: 长时容错
  - 4.1 Executor 端长时容错详解
  - 4.2 Driver 端长时容错详解
- StreamingContext
  - 5.1 StreamingContext 详解

Google

广点通 spark streaming



# Agenda

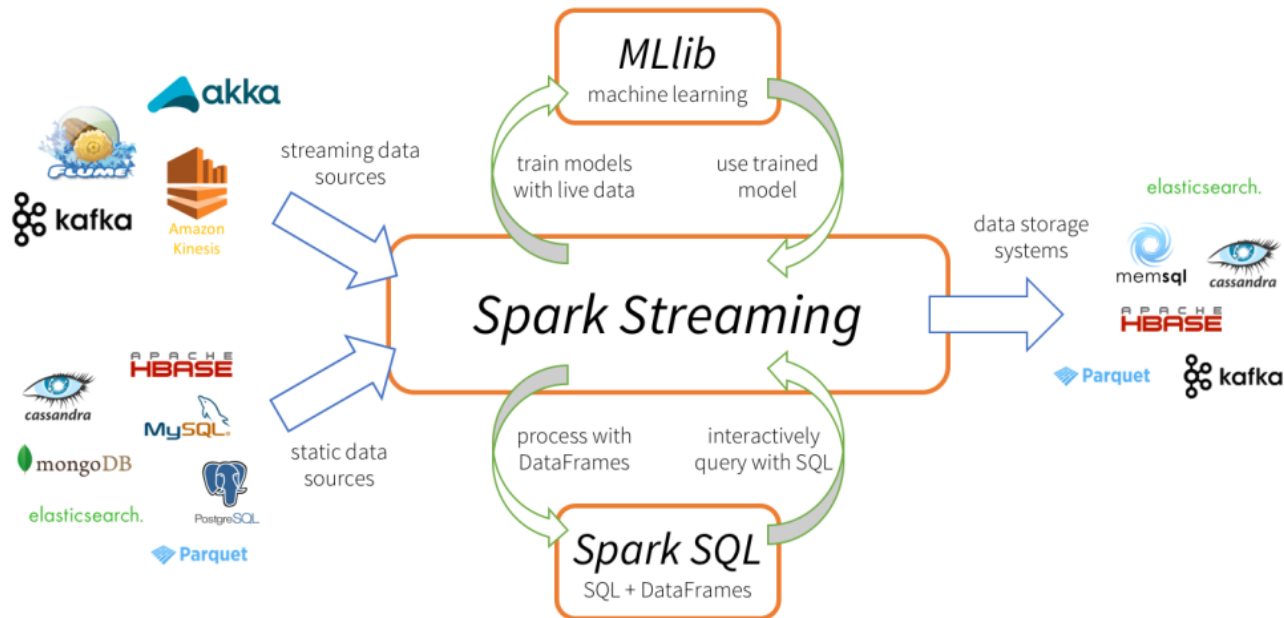
## 概述

- Spark Streaming 基本架构
- Spark Streaming at 广点通

特性与应用

优化经验

# Spark Streaming 基本架构





# Spark Streaming at 广点通

腾讯 Spark 技术栈 (powered by  数据平台部 )



TDBank  
*message queue*



Spark Streaming at 广点通

#apps  
~30

#cores  
~2000

data volume  
~70 TB/day

peak rate  
~600 k/sec

# Agenda

概述

特性与应用

- (1) exactly-once
- (2) 可靠状态
- (3) batch 调度

优化经验



# 特性与应用-(1) exactly-once

## Spark 执行单元

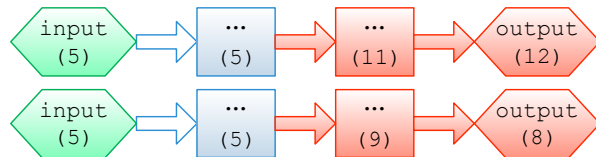
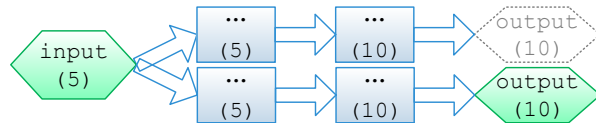
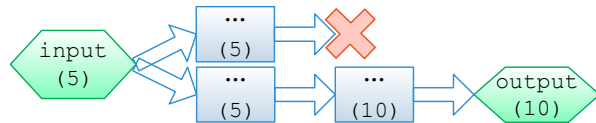
- 任务（即一批数据）
- 一批数据全部成功/全部失败

## Task 重做

- 失败重做：task 重做、stage 重做
- 推测执行：另一个节点同时做
- Committer: 任务唯一成功

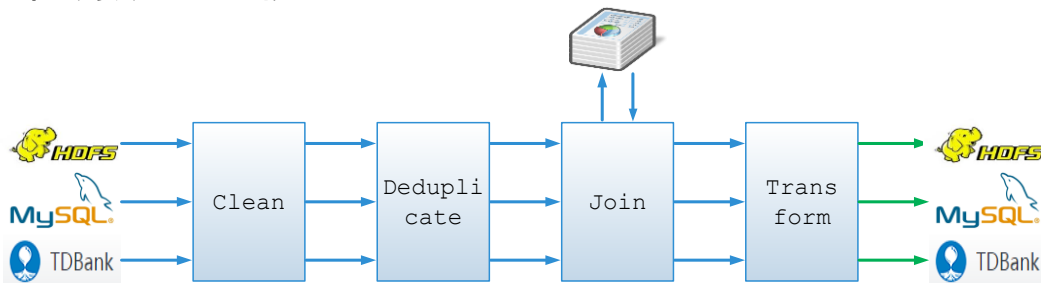
## 其它系统

- Storm: at-most-once  
at-least-once
- MapReduce: exactly-once

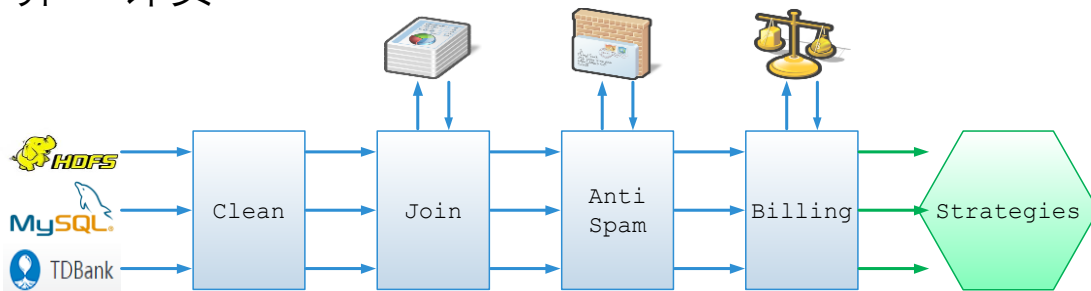


# 特性与应用-(1) exactly-once

应用：实时准确数据转移



应用：反作弊 + 计费！！



# 特性与应用-(2) 可靠状态

## Spark Streaming 天然面向状态

- RDD: Resilient Distributed Datasets
- RDD lineage
  - 容错：重做
- rdd.checkpoint()
  - HDFS, S3... 等

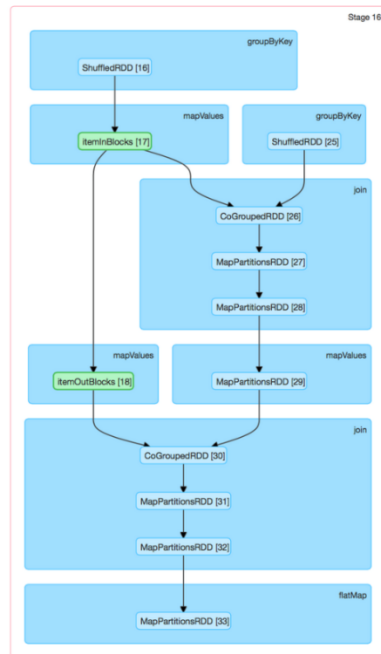
## 可靠状态管理

- Spark 1.5: updateByKey()
- Spark 1.6: [kv store] mapWithState()
- Spark 2.0: [kv store] StateStore

### Details for Stage 16 (Attempt 0)

Total Time Across All Tasks: 0.1 s  
Input Size / Records: 1088.0 B / 4  
Shuffle Read: 3.2 KB / 16  
Shuffle Write: 3.2 KB / 16

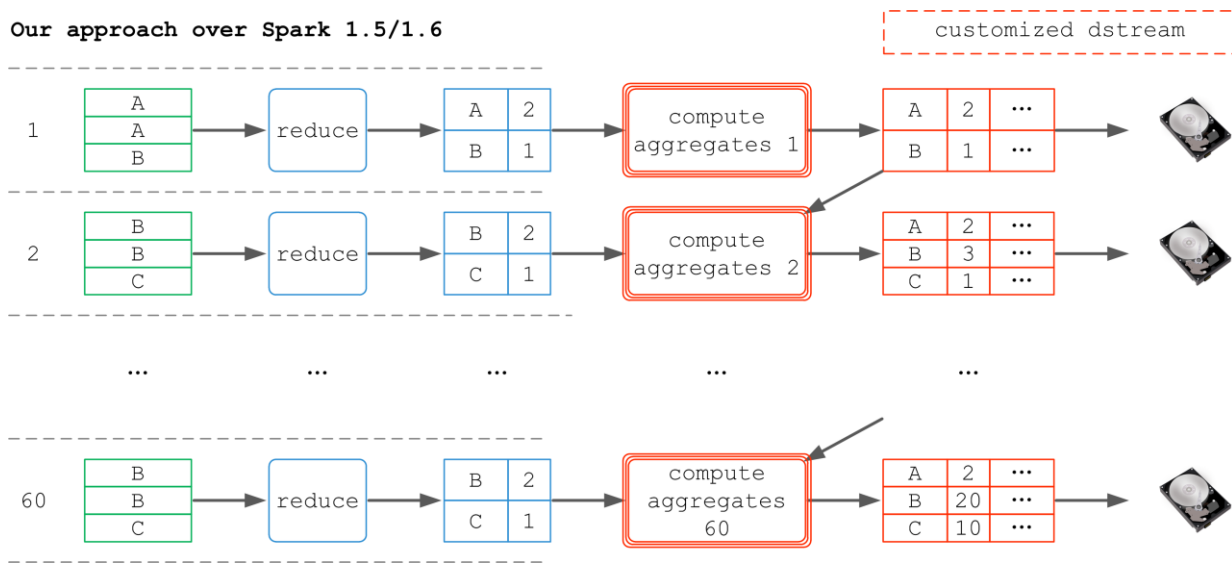
↳ DAG Visualization



## 特性与应用-(2) 可靠状态

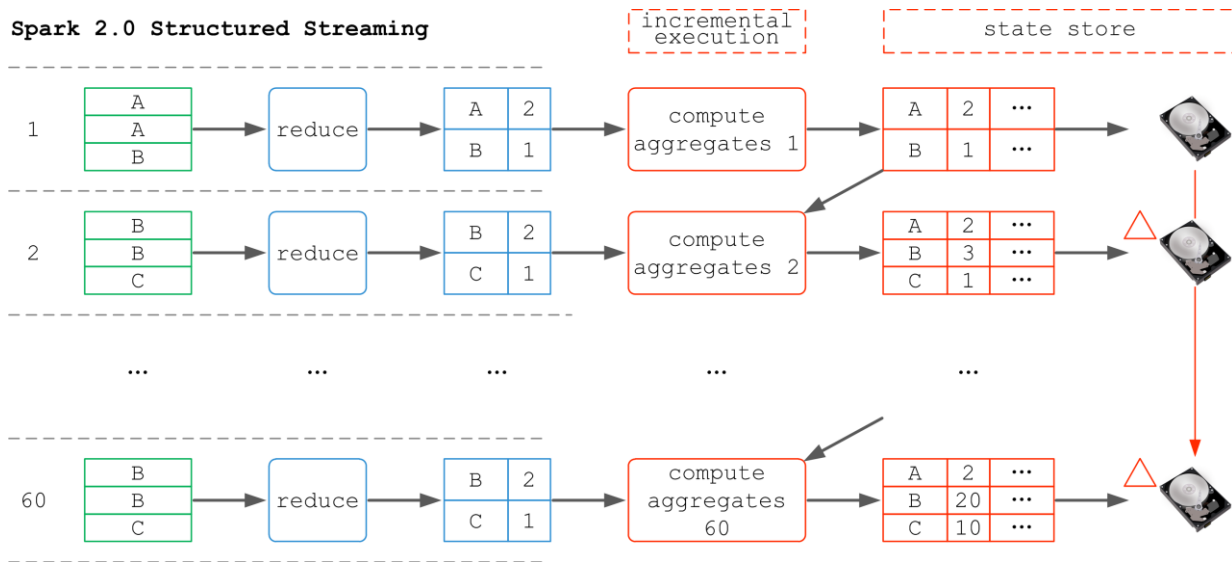
应用：跨 batch 聚合 (pv/uv 计算，记录去重，微额记账等)

Our approach over Spark 1.5/1.6



# 特性与应用-(2) 可靠状态

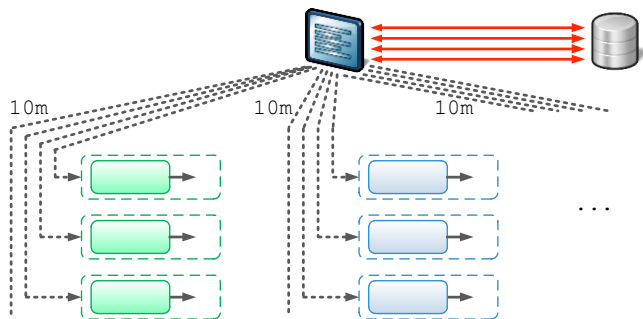
应用：跨 batch 聚合 (pv/uv 计算，记录去重，微额记账等)



# 特性与应用-(3) 快速 batch 调度

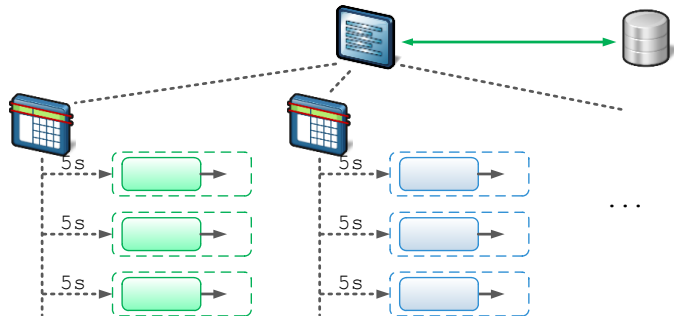
## MapReduce 的实例调度

- 一级调度系统 (oozie 等)
  - 最小间隔 10 min
  - 进程调度
  - 需要一定启动时间



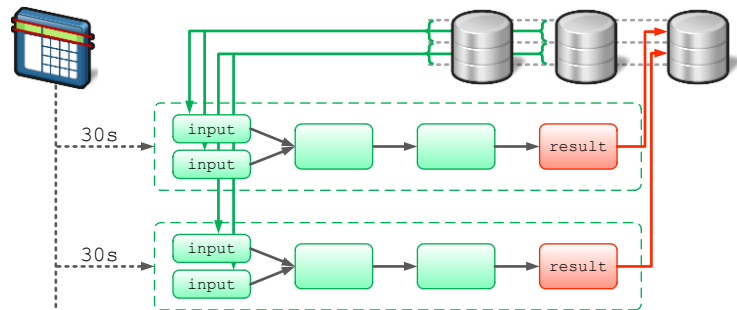
## Spark Streaming 的 batch 调度

- 一级调度系统 (oozie 等)
- 二级调度系统
  - Driver / JobScheduler 调度
  - 间隔 1s ~ 60s
  - 进程常驻+线程调度、无启动时间

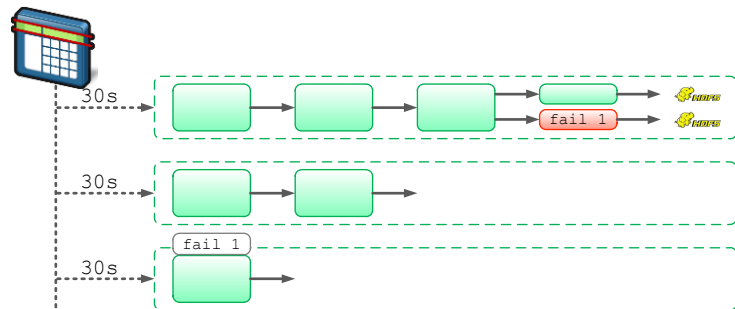


# 特性与应用-(3) 快速 batch 调度

应用：数据指标监控



应用：复杂 pipeline 的  
未成功数据唯一快速重试



# Agenda

概述

特性与应用

优化经验

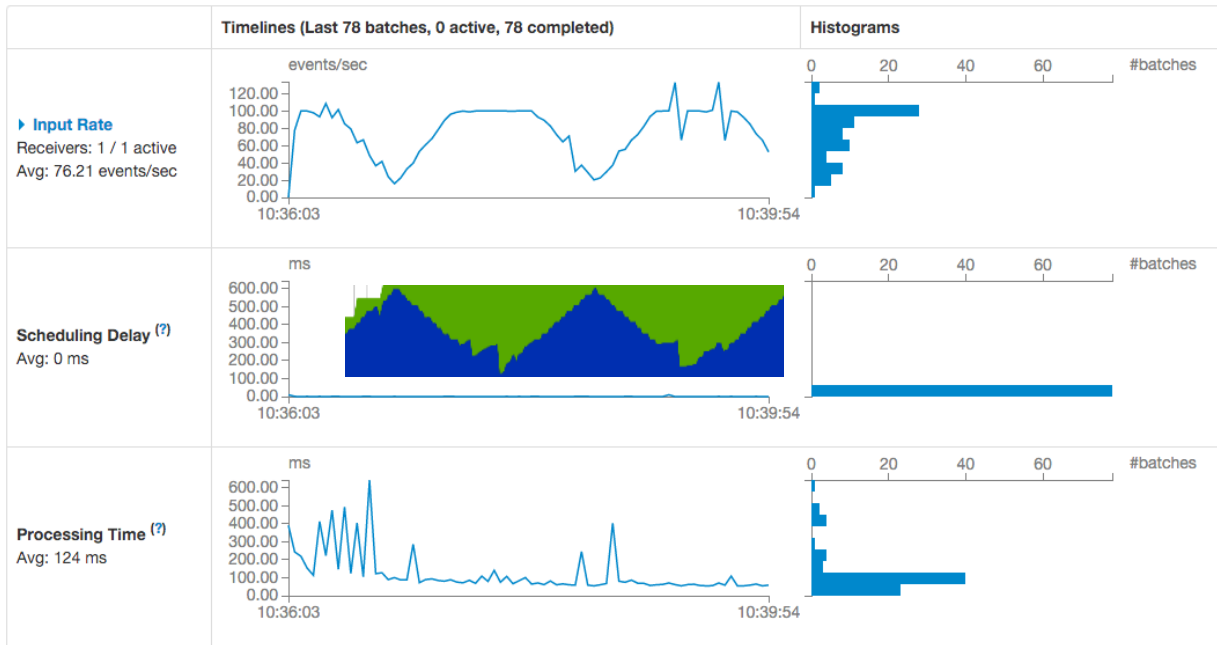
- (1) 增加 Memory Back Pressure
- (2) 为 Spark 增加新特性（无需编译 Spark 优化）
- (3) SparkSQL API > RDD API
- (4) async execution within a task
- (5) try-cath
- (6) concurrentJobs 开启
- (7) Spark 远程调试



## 优化经验

## (1) 增加 Memory Back Pressure

- Receiver 的接收速率随 Executor 的内存使用动态放缩, 避免 OOM



Line 1 (tab)4.36.48

File

Blame

History

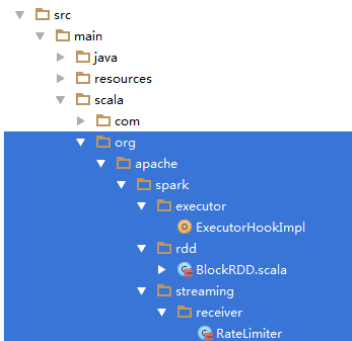
Q

```
1 // Licensed to the Apache Software Foundation (ASF) under one or more
2 // contributor license agreements. See the NOTICE file distributed with
3 // this work for additional information regarding copyright ownership.
4 // The ASF licenses this file to You under the Apache License, Version 2.0
5 // (the "License"); you may not use this file except in compliance with
6 // the License. You may obtain a copy of the License at
7
8 // http://www.apache.org/licenses/LICENSE-2.0
9
10 // Unless required by applicable law or agreed to in writing, software
11 // distributed under the License is distributed on an "AS IS" BASIS,
12 // WITHOUT WARRANTIES OR REPRESENTATIONS OF ANY KIND, either express or implied.
13 // See the License for the specific language governing permissions and
14 // limitations under the License.
15
16 //
17 package org.apache.cassandra.thrift;
18
19 import org.apache.cassandra.config.DatabaseDescriptor;
20 import org.apache.cassandra.config.DatabaseDescriptor;
21 import org.apache.cassandra.config.DatabaseDescriptor;
22
23 // Provides methods to limit the rate at which requests come into
24 //
25 // and also allows other user messages have been pushed to backlog,
26 //
27 //
28 // The spark configuration option streaming.requested.messages gives the maximum number of messages
29 // or more than that request will accept.
30
31 //spark.cassandra.streaming.requested.messages
32
33 //
34 //
35 //
36 //
37 //
38 //
39 //
40 //
41 //
42 //
43 //
44 //
45 //
46 //
47 //
48 //
49 //
50 //
51 //
52 //
53 //
54 //
55 //
56 //
57 //
58 //
59 //
60 //
61 //
62 //
63 //
64 //
65 //
66 //
67 //
68 //
69 //
70 //
71 //
72 //
73 //
74 //
75 //
76 //
77 //
78 //
79 //
80 //
81 //
82 //
83 //
84 //
85 //
86 //
87 //
88 //
89 //
90 //
91 //
92 //
93 //
94 //
95 //
96 //
97 //
98 //
99 //
100 //
101 //
102 //
103 //
104 //
105 //
106 //
107 //
108 //
109 //
110 //
111 //
112 //
113 //
114 //
115 //
116 //
117 //
118 //
119 //
120 //
121 //
122 //
123 //
124 //
125 //
126 //
127 //
128 //
129 //
130 //
131 //
132 //
133 //
134 //
135 //
136 //
137 //
138 //
139 //
140 //
141 //
142 //
143 //
144 //
145 //
146 //
147 //
148 //
149 //
150 //
151 //
152 //
153 //
154 //
155 //
156 //
157 //
158 //
159 //
160 //
161 //
162 //
163 //
164 //
165 //
166 //
167 //
168 //
169 //
170 //
171 //
172 //
173 //
174 //
175 //
176 //
177 //
178 //
179 //
180 //
181 //
182 //
183 //
184 //
185 //
186 //
187 //
188 //
189 //
190 //
191 //
192 //
193 //
194 //
195 //
196 //
197 //
198 //
199 //
200 //
201 //
202 //
203 //
204 //
205 //
206 //
207 //
208 //
209 //
210 //
211 //
212 //
213 //
214 //
215 //
216 //
217 //
218 //
219 //
220 //
221 //
222 //
223 //
224 //
225 //
226 //
227 //
228 //
229 //
230 //
231 //
232 //
233 //
234 //
235 //
236 //
237 //
238 //
239 //
240 //
241 //
242 //
243 //
244 //
245 //
246 //
247 //
248 //
249 //
250 //
251 //
252 //
253 //
254 //
255 //
256 //
257 //
258 //
259 //
260 //
261 //
262 //
263 //
264 //
265 //
266 //
267 //
268 //
269 //
270 //
271 //
272 //
273 //
274 //
275 //
276 //
277 //
278 //
279 //
280 //
281 //
282 //
283 //
284 //
285 //
286 //
287 //
288 //
289 //
290 //
291 //
292 //
293 //
294 //
295 //
296 //
297 //
298 //
299 //
300 //
301 //
302 //
303 //
304 //
305 //
306 //
307 //
308 //
309 //
310 //
311 //
312 //
313 //
314 //
315 //
316 //
317 //
318 //
319 //
320 //
321 //
322 //
323 //
324 //
325 //
326 //
327 //
328 //
329 //
330 //
331 //
332 //
333 //
334 //
335 //
336 //
337 //
338 //
339 //
340 //
341 //
342 //
343 //
344 //
345 //
346 //
347 //
348 //
349 //
350 //
351 //
352 //
353 //
354 //
355 //
356 //
357 //
358 //
359 //
360 //
361 //
362 //
363 //
364 //
365 //
366 //
367 //
368 //
369 //
370 //
371 //
372 //
373 //
374 //
375 //
376 //
377 //
378 //
379 //
380 //
381 //
382 //
383 //
384 //
385 //
386 //
387 //
388 //
389 //
390 //
391 //
392 //
393 //
394 //
395 //
396 //
397 //
398 //
399 //
400 //
401 //
402 //
403 //
404 //
405 //
406 //
407 //
408 //
409 //
410 //
411 //
412 //
413 //
414 //
415 //
416 //
417 //
418 //
419 //
420 //
421 //
422 //
423 //
424 //
425 //
426 //
427 //
428 //
429 //
430 //
431 //
432 //
433 //
434 //
435 //
436 //
437 //
438 //
439 //
440 //
441 //
442 //
443 //
444 //
445 //
446 //
447 //
448 //
449 //
450 //
451 //
452 //
453 //
454 //
455 //
456 //
457 //
458 //
459 //
460 //
461 //
462 //
463 //
464 //
465 //
466 //
467 //
468 //
469 //
470 //
471 //
472 //
473 //
474 //
475 //
476 //
477 //
478 //
479 //
480 //
481 //
482 //
483 //
484 //
485 //
486 //
487 //
488 //
489 //
490 //
491 //
492 //
493 //
494 //
495 //
496 //
497 //
498 //
499 //
500 //
501 //
502 //
503 //
504 //
505 //
506 //
507 //
508 //
509 //
510 //
511 //
512 //
513 //
514 //
515 //
516 //
517 //
518 //
519 //
520 //
521 //
522 //
523 //
524 //
525 //
526 //
527 //
528 //
529 //
530 //
531 //
532 //
533 //
534 //
535 //
536 //
537 //
538 //
539 //
540 //
541 //
542 //
543 //
544 //
545 //
546 //
547 //
548 //
549 //
550 //
551 //
552 //
553 //
554 //
555 //
556 //
557 //
558 //
559 //
560 //
561 //
562 //
563 //
564 //
565 //
566 //
567 //
568 //
569 //
570 //
571 //
572 //
573 //
574 //
575 //
576 //
577 //
578 //
579 //
580 //
581 //
582 //
583 //
584 //
585 //
586 //
587 //
588 //
589 //
590 //
591 //
592 //
593 //
594 //
595 //
596 //
597 //
598 //
599 //
600 //
601 //
602 //
603 //
604 //
605 //
606 //
607 //
608 //
609 //
610 //
611 //
612 //
613 //
614 //
615 //
616 //
617 //
618 //
619 //
620 //
621 //
622 //
623 //
624 //
625 //
626 //
627 //
628 //
629 //
630 //
631 //
632 //
633 //
634 //
635 //
636 //
637 //
638 //
639 //
640 //
641 //
642 //
643 //
644 //
645 //
646 //
647 //
648 //
649 //
650 //
651 //
652 //
653 //
654 //
655 //
656 //
657 //
658 //
659 //
660 //
661 //
662 //
663 //
664 //
665 //
666 //
667 //
668 //
669 //
670 //
671 //
672 //
673 //
674 //
675 //
676 //
677 //
678 //
679 //
680 //
681 //
682 //
683 //
684 //
685 //
686 //
687 //
688 //
689 //
690 //
691 //
692 //
693 //
694 //
695 //
696 //
697 //
698 //
699 //
700 //
701 //
702 //
703 //
704 //
705 //
706 //
707 //
708 //
709 //
710 //
711 //
712 //
713 //
714 //
715 //
716 //
717 //
718 //
719 //
720 //
721 //
722 //
723 //
724 //
725 //
726 //
727 //
728 //
729 //
730 //
731 //
732 //
733 //
734 //
735 //
736 //
737 //
738 //
739 //
740 //
741 //
742 //
743 //
744 //
745 //
746 //
747 //
748 //
749 //
750 //
751 //
752 //
753 //
754 //
755 //
756 //
757 //
758 //
759 //
760 //
761 //
762 //
763 //
764 //
765 //
766 //
767 //
768 //
769 //
770 //
771 //
772 //
773 //
774 //
775 //
776 //
777 //
778 //
779 //
780 //
781 //
782
```

# 优化经验

## (2) 为 Spark 增加新特性（无需编译 Spark 工程）

- A. 直接改源文件 `***.scala`
  - src 原包名下，如 `src/o/a/s/streaming/receiver/RateLimiter.scala`
  - 运行参数：`spark.driver/executor.userClassPathFirst=false; spark.driver/executor.extraClassPath=app.jar`
- B. 直接改字节码 `***.class`
  - resources 原包名下，如 `resources/o/a/s/executor/Executor.class`
  - 运行参数：`spark.driver/executor.userClassPathFirst=false; spark.driver/executor.extraClassPath=app.jar`



A

```
38 08 astore 5
39 10 new #12, <org/apache/spark/executor/Executor$Bound1>
40 13 dup
41 14 aload_0
42 15 aload_1
43 16 aload_2
44 17 aload_3
45 18 load 4
46 20 aload 5
47 82 invokespecial #565, <org/apache/spark/executor/Executor$Bound1.<init>()V>(<org/apache/spark/executor/Executor$Bound1>:Lorg/apache/spark/executor/Executor$Bound1;)V
48 85 astore 6
49 87 aload 6
50 89 iconst_1
51 90 invokevirtual #568, <java/lang/Thread.setDaemon(Z)V>(<java/lang/Thread>:Ljava/lang/Thread;)V
52 93 aload 6
53 95 ldc_w #570, <Driver Heartbeater>
54 98 invokevirtual #574, <java/lang/Thread.setName(Ljava/lang/String;)V>(<java/lang/Thread>:Ljava/lang/Thread;)V
55 101 aload 6
56 103 invokevirtual #577, <java/lang/Thread.start()V>(<java/lang/Thread>:Ljava/lang/Thread;)V
57 106 return
```

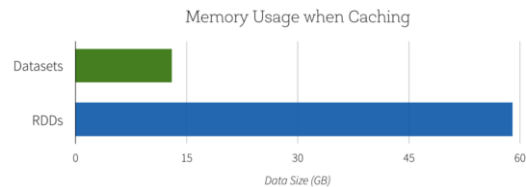
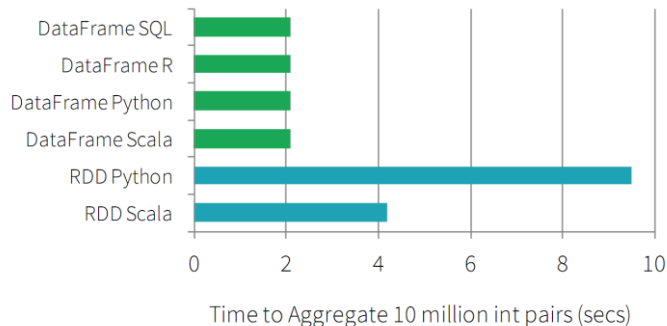
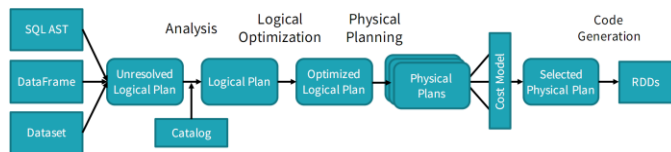
```
38 08 astore 5
39 10 new #12, <org/apache/spark/executor/Executor$Bound1>
40 13 dup
41 14 aload_0
42 15 aload_1
43 16 aload_2
44 17 aload_3
45 18 load 4
46 20 aload 5
47 82 invokespecial #565, <org/apache/spark/executor/Executor$Bound1.<init>()V>(<org/apache/spark/executor/Executor$Bound1>:Lorg/apache/spark/executor/Executor$Bound1;)V
48 85 astore 6
49 87 aload 6
50 89 iconst_1
51 90 invokevirtual #568, <java/lang/Thread.setDaemon(Z)V>(<java/lang/Thread>:Ljava/lang/Thread;)V
52 93 aload 6
53 95 ldc_w #570, <Driver Heartbeater>
54 98 invokevirtual #574, <java/lang/Thread.setName(Ljava/lang/String;)V>(<java/lang/Thread>:Ljava/lang/Thread;)V
55 101 aload 6
56 103 invokevirtual #577, <java/lang/Thread.start()V>(<java/lang/Thread>:Ljava/lang/Thread;)V
57 106 getstatic #578, <java/lang/System.out:Ljava/io/PrintStream>
58 109 ldc_w #571, <Inside org.apache.spark.executor.Executor.startDriverHeartbeater() before invoke hook>
59 112 invokevirtual #576, <java/io/PrintStream.println(Ljava/lang/String;)V>(<java/io/PrintStream>:Ljava/io/PrintStream;)V
60 115 getstatic #579, <org/apache/spark/executor/ExecutorHook:HOOK_HF:Ljava/lang/Runnable;V>
61 118 aload_0
62 119 invokevirtual #574, <org/apache/spark/executor/ExecutorHook.hook(Lorg/apache/spark/executor/Executor;)V>(<org/apache/spark/executor/ExecutorHook:Ljava/lang/Runnable;V>:Ljava/lang/Runnable;)V
63 122 getstatic #578, <java/lang/System.out:Ljava/io/PrintStream>
64 125 ldc_w #572, <Inside org.apache.spark.executor.Executor.startDriverHeartbeater() after invoke hook>
65 128 invokevirtual #576, <java/io/PrintStream.println(Ljava/lang/String;)V>(<java/io/PrintStream>:Ljava/io/PrintStream;)V
66 131 return
```

B

# 优化经验

## (3) SparkSQL API > RDD API

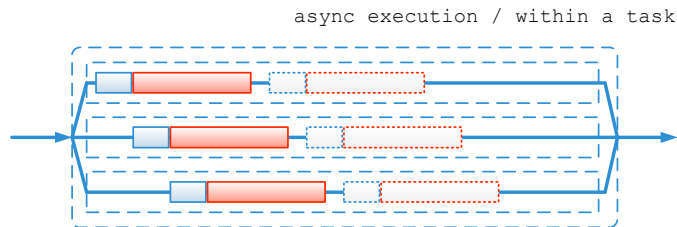
- Run faster
  - Catalyst Optimizer
  - Tungsten Engine
    - Memory Management
    - Cache-aware Algorithms
    - Whole Stage Codegen
- Spark 1.x
  - `dstream.foreachRDD { rdd => rdd.toDF().select...`
- Spark 2.x: Structured Streaming
  - `spark....stream....startStream()`



# 优化经验

## (4) async execution within a task

- 应对外部操作：线程池+异步



## (5) try-catch

- task 的错误，会在 driver 端抛出
- 屏蔽 "Could not compute split" 等问题

```
val inputDStream = ssc.fileStream("...")
inputDStream.foreachRDD(rdd => {
  try {
    // do something
  } catch {
    case NonFatal(e) => errorHandler(e)
  }
})
```

## (6) concurrentJobs 开启

- 内部参数，同时执行 n 个 output
  - 一般 1 个 batch 对应 1 个 output
- spark.streaming.concurrentJobs = n

Spark Jobs Stages Storage Environment Executors Streaming

Active Batches (10)

Batch Time	Input Size	Scheduling Delay <sup>(1)</sup>	Processing Time <sup>(1)</sup>	Status
2015/01/01 00:00:55	50 events	-	-	queued
2015/01/01 00:00:50	50 events	-	-	queued
2015/01/01 00:00:45	50 events	-	-	queued
2015/01/01 00:00:40	50 events	-	-	queued
2015/01/01 00:00:35	49 events	-	-	queued
2015/01/01 00:00:30	50 events	5 ms	-	processing
2015/01/01 00:00:25	50 events	1 ms	-	processing
2015/01/01 00:00:20	50 events	4 ms	-	processing
2015/01/01 00:00:15	50 events	1 ms	-	processing
2015/01/01 00:00:10	6 events	9 ms	-	processing

# 优化经验

## (7) Spark 远程调试

- your profiler 按需分发



```
73 private def cmds() = {  
74     val LS = "ls -trhl";  
75     Array(  
76         s"mkdir ${dirPath};",  
77         LS,  
78         s"cd ${dirPath};",  
79         s"cp -L ../${srcJarPath} ${desJarPath};",  
80         LS,  
81         s"jar -xf ${desJarPath};",  
82         LS,  
83         s"tar zxf ${resourcesPath};",  
84         LS,  
85         s"cmd to enable your profiler..."  
86     )  
}
```



# 总结与展望

## 总结

特性	应用
(1) exactly-once	实时准确数据转移 反作弊 + 计费 !!!
(2) 可靠状态	跨 batch 聚合 (pv/uv 计算, 记录去重, 微额记账等)
(3) 快速 batch 调度	数据指标监控 复杂 pipeline 的未成功数据唯一重试

## 展望

- From Lambda Architecture(MR + Storm) to Spark Streaming
- Spark 2.0: Structured Streaming
  - High-level streaming API built on Spark SQL engine
  - Event time, windowing, sessions, sources & sinks



Thanks!