Spark Streaming , Graphx & ML
پردازش داده های جریانی، گراف و یادگیری ماشین در اسپارک

عادل قاضی خانی
هیئت علمی گروه کامپیوتر دانشگاه امام رضا (ع)

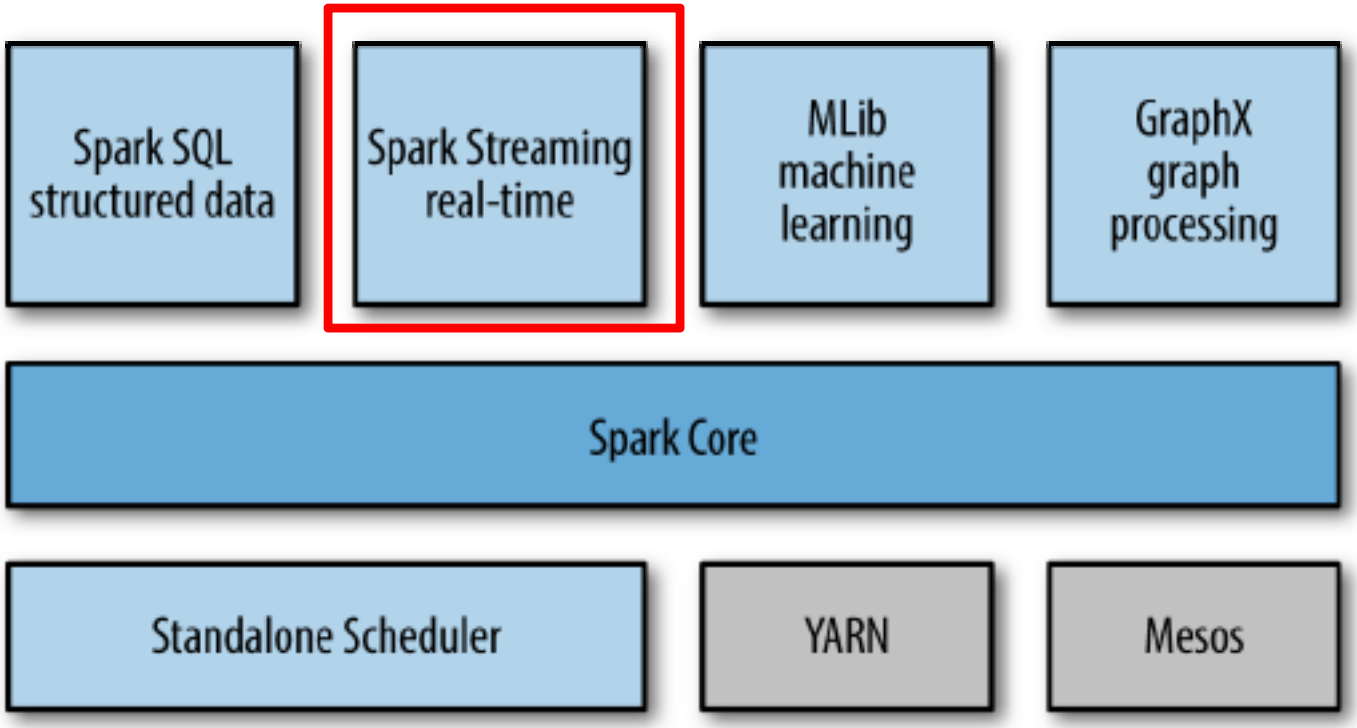
پردازش داده های جریانی

Spark Streaming

Spark Streaming

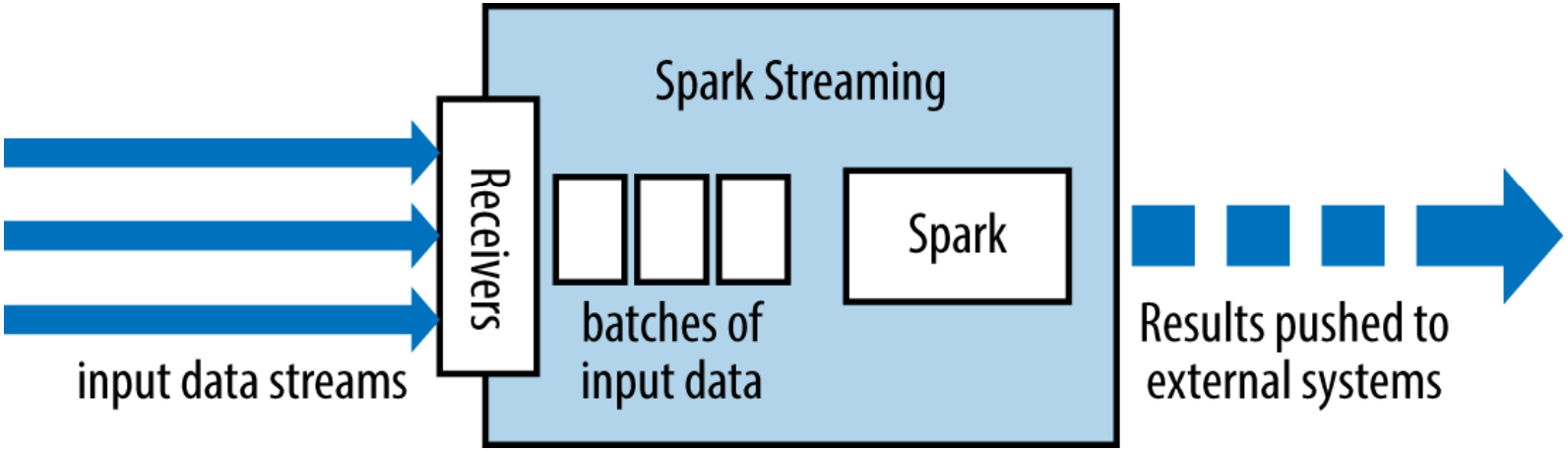
مقدمه

- یکی از بخشهای اسپارک پردازش داده های جریانی است



پردازش داده های جریانی

- ساختار کلی



پردازش داده های جریانی

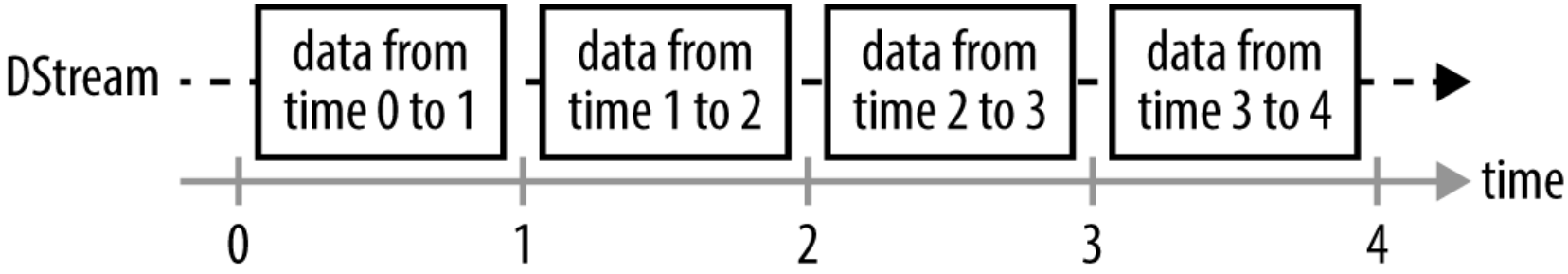
- در اسپارک **داده‌های جریانی** ورودی که ممکن است از منابع مختلف باشند **بصورت تکه های کوچک*** پردازش می شوند

- هر تکه داده در بازه های زمانی مشخص دریافت می شود

micro batch

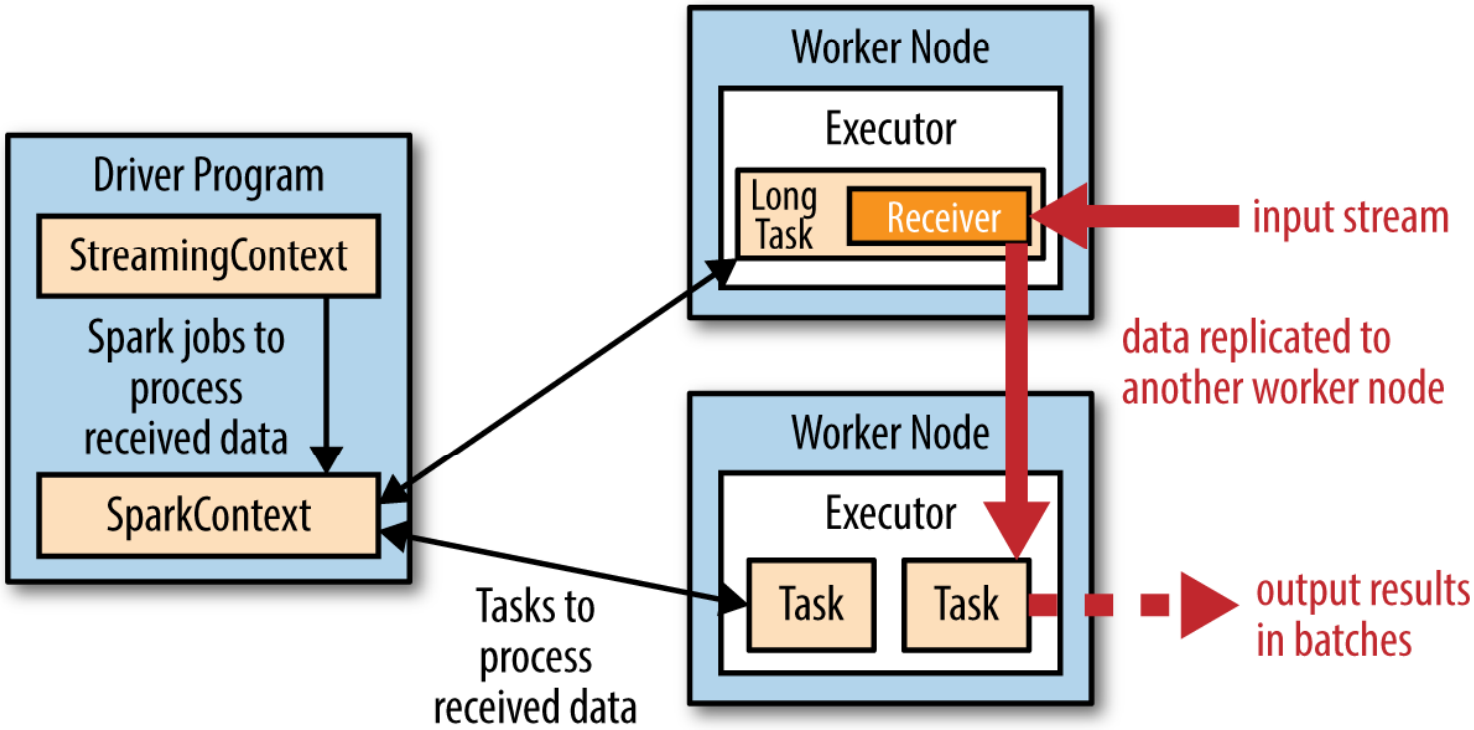
پردازش داده های جریانی

- در اسپارک ساختار داده Dstream وجود دارد



پردازش داده های جریانی

- ساختار اجرای برنامه در اسپارک



پردازش داده های جریانی

- **تبدیلات:**

❖**بدون حالت:** پردازش هر تکه داده وابسته به تکه های قبلی نیست

❖**با حالت :** پردازش هر تکه داده وابسته به تکه های قبلی هست

Stateless
Stateful

Spark Streaming

تبدیلات بدون حالت

Function name	Purpose	Scala example	Signature of user-supplied function on DStream[T]
map()	Apply a function to each element in the DStream and return a DStream of the result.	ds.map(x => x + 1)	f: (T) → U
flatMap()	Apply a function to each element in the DStream and return a DStream of the contents of the iterators returned.	ds.flatMap(x => x.split(" "))	f: T → Iterable[U]
filter()	Return a DStream consisting of only elements that pass the condition passed to filter.	ds.filter(x => x != 1)	f: T → Boolean
repartition()	Change the number of partitions of the DStream.	ds.repartition(10)	N/A
reduceByKey()	Combine values with the same key in each batch.	ds.reduceByKey((x, y) => x + y)	f: T, T → T
groupByKey()	Group values with the same key in each batch.	ds.groupByKey()	N/A

تبدیلات با حالت

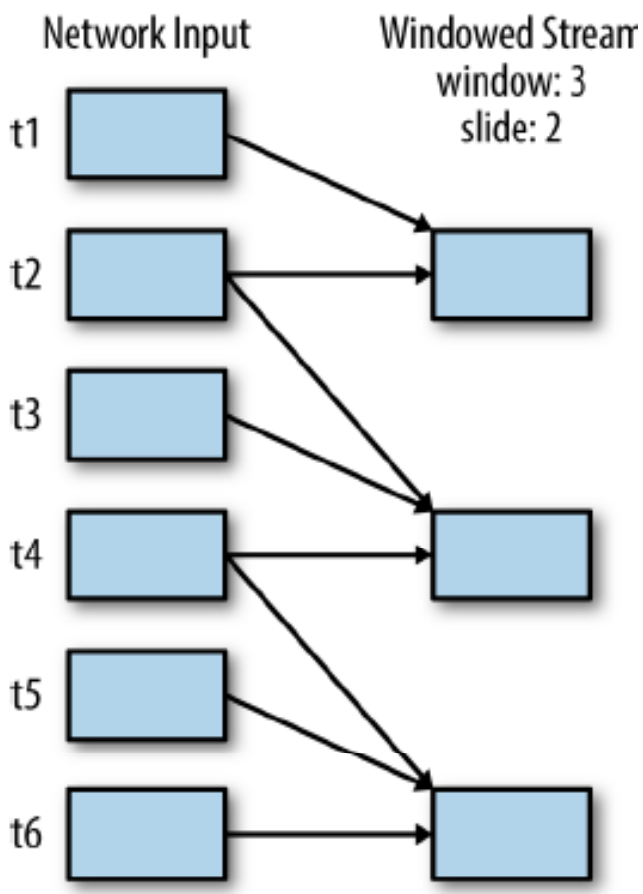
- **دو نوع اصلی:**

۱- تبدیلات پنجره ای

- در این تبدیلات دو پارامتر وجود دارد: **زمان پنجره و زمان سر خردن**

تبدیلات با حالت

- دو نوع اصلی:
  - ❖تبدیلات پنجره ای – مثال



Spark Streaming

تبدیلات با حالت

- **دو نوع اصلی:**

۲- تبدیلات بروزرسانی حالت با کلید

- در این تبدیلات یک زوج (کلید، حالت) وجود دارد که در آن با یک تابع مشخص می شود که حالت چگونه بروزرسانی شود

Spark Streaming

تبدیلات با حالت

- **دو نوع اصلی:**

۲- تبدیلات بروزرسانی حالت با کلید

- در این تبدیلات یک زوج (کلید، حالت) وجود دارد که در آن با یک تابع مشخص می شود که با داشتن کلید حالت چگونه بروزرسانی شود

Spark Streaming

انواع ورودیها

- Stream of files
- Akka actor stream
- Apache Kafka
- Apache Flume
- Push-based receiver
- Pull-based receiver

Spark Streaming

عملیات 24/7

- تضمین **همیشگی پردازش داده ها**
- مستلزم *Checkpointing* ؟

Spark Streaming

Checkpointing

- تکنیکی که در آن در بازه هایی اطلاعات در جایی مطمئن ذخیره سازی می شوند (snapshot) تا بتوانیم تحمل پذیری در برابر خطا داشته باشیم

- در Spark باید HDFS یا یک منبع مطمئن دیگر برای این منظور مشخص گردد

```
ssc.checkpoint("hdfs://...")
```

- بعد از فعال کردن امکان تحمل پذیری خطا برای راه انداز، اجرا کننده و دریافت کننده داده های جریانی وجود دارد

Spark Streaming

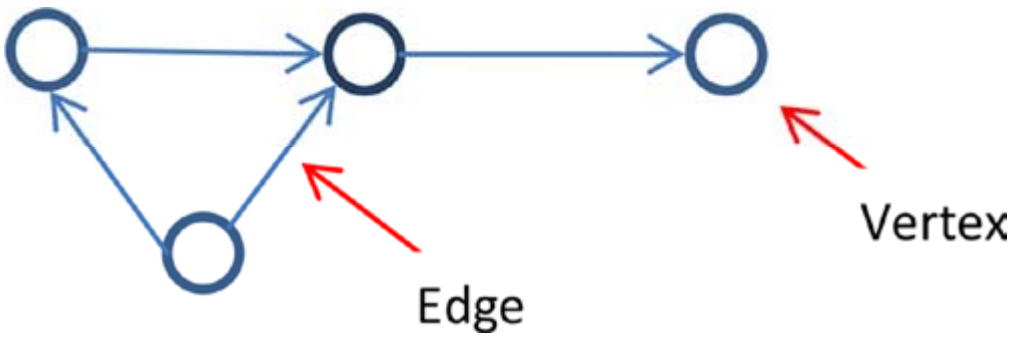
عملکرد پردازشی

- تعیین اندازه تکه های داده ها
- تعیین اندازه پنجره
- میزان موازی سازی

پردازش گراف در اسپارک

مقدمه

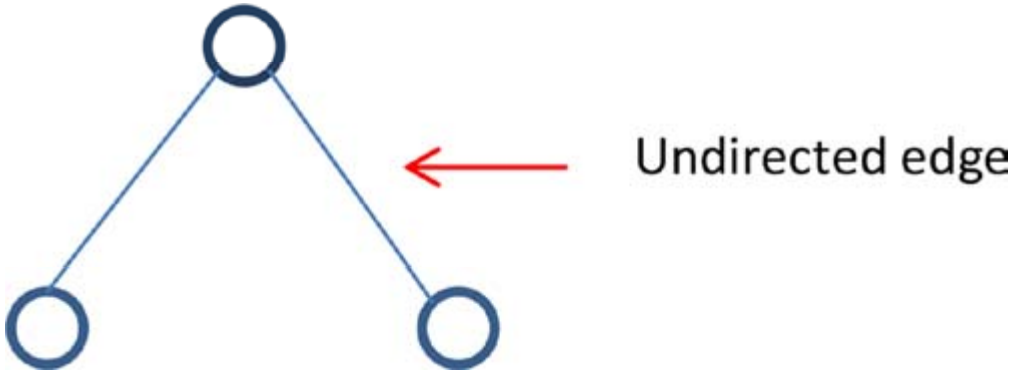
- بسیاری از داده ها را می توان با گراف به شکل بهتری بیان کرد



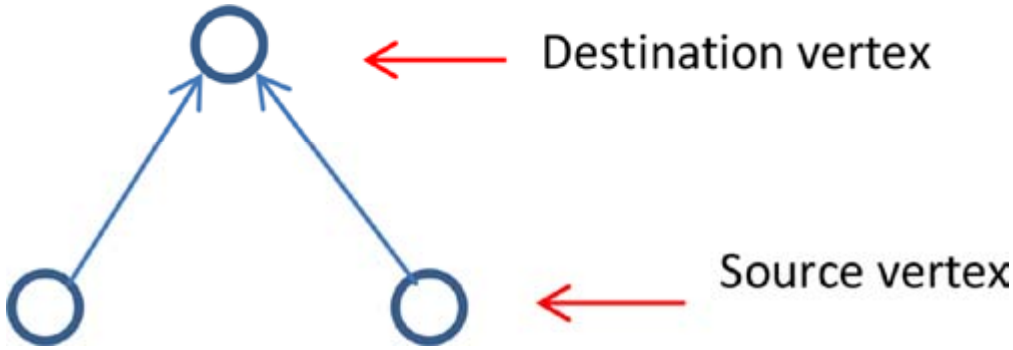
- **مثلاً**: شبکه اجتماعی

مقدمه

- گراف بدون جهت

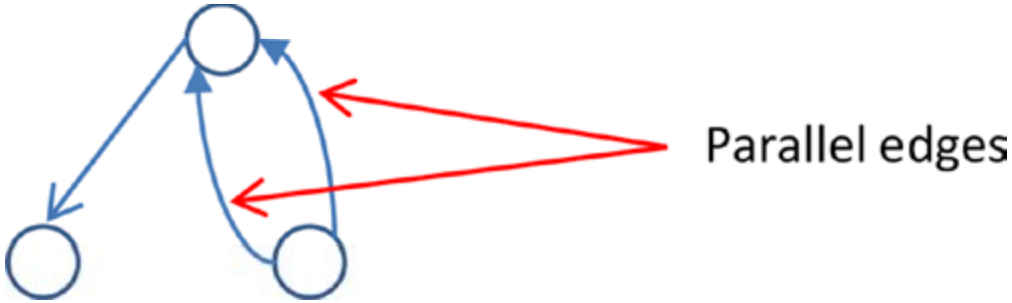


- گراف جهتدار

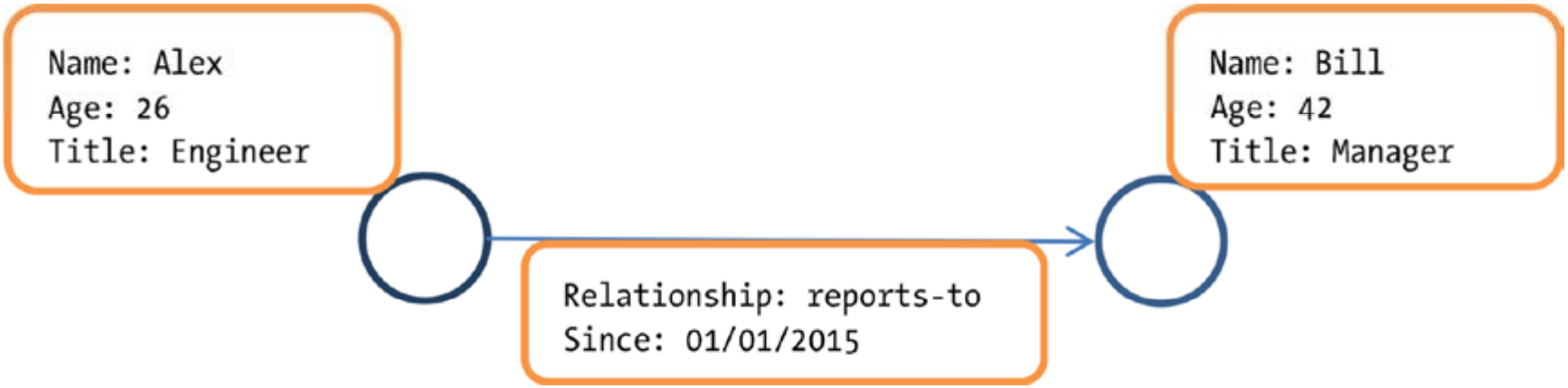


مقدمه

- چند گراف جهتدار



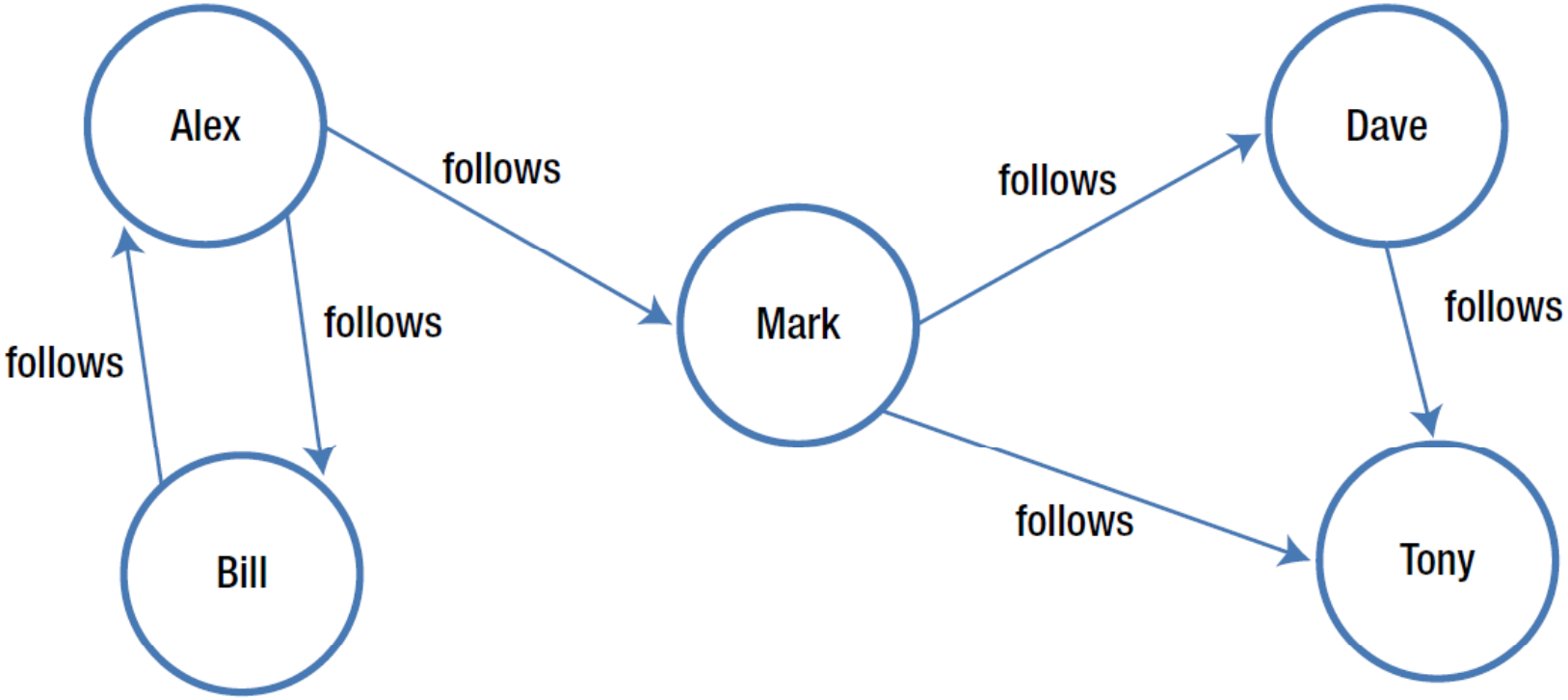
- گراف خاصیت



Property graph

مقدمه

- **مثال:** گراف خاصیت



مقدمه

- **Graphx:**
- بستری برای پردازش توزیع شده گراف. شامل عملیات پایه ای و پیشرفته روی گراف، همچنین الگوریتمهای پردازش گراف

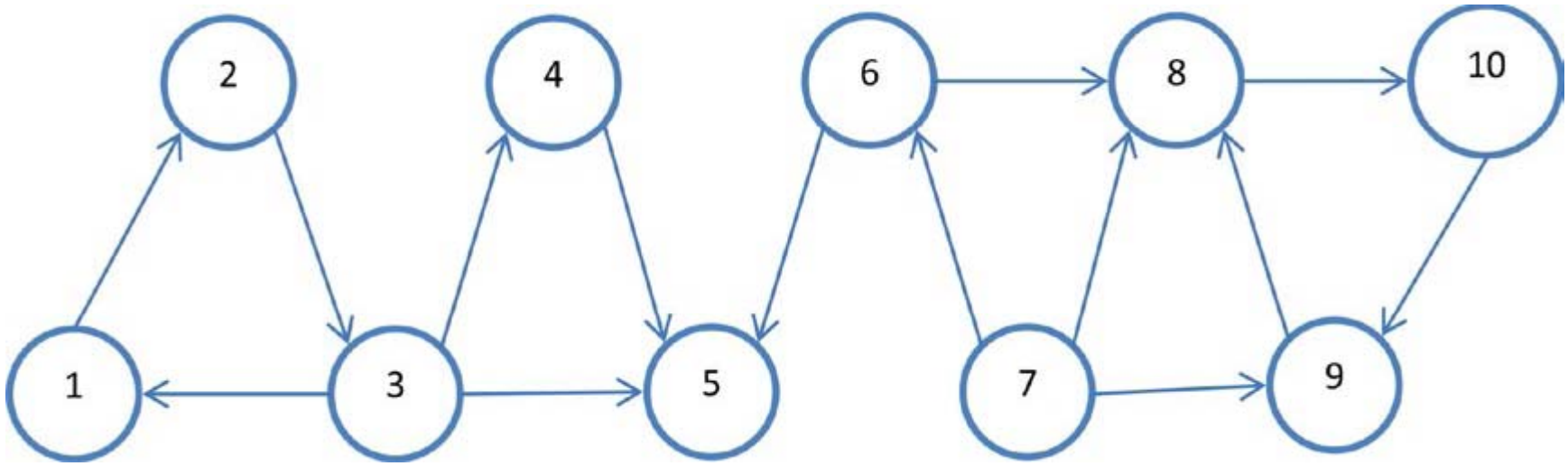
مراحل پردازش گراف

- خواندن داده ها
- پیش پردازش
- استخراج رئوس و لبه ها برای ایجاد گراف خاصیت
- انتخاب یک زیرگراف
- اجرا الگوریتم پردازش
- تحلیل نتایج

انواع داده ها

- **VertexRDD:** Distributed set of vertex
- **Edge:** Single Edge
- **EdgeRDD:** Distributed Edge
- **EdgeTriplet:** Edge and vertices it connects
- **Graph:** property graph

ایجاد گراف مثال



ایجاد گراف مثال

```
val users = List((1L, User("Alex", 26)), (2L, User("Bill", 42)), (3L, User("Carol", 18)),
                 (4L, User("Dave", 16)), (5L, User("Eve", 45)), (6L, User("Farell", 30)),
                 (7L, User("Garry", 32)), (8L, User("Harry", 36)), (9L, User("Ivan", 28)),
                 (10L, User("Jill", 48))
               )

val usersRDD = sc.parallelize(users)

val follows = List(Edge(1L, 2L, 1), Edge(2L, 3L, 1), Edge(3L, 1L, 1), Edge(3L, 4L, 1),
                   Edge(3L, 5L, 1), Edge(4L, 5L, 1), Edge(6L, 5L, 1), Edge(7L, 6L, 1),
                   Edge(6L, 8L, 1), Edge(7L, 8L, 1), Edge(7L, 9L, 1), Edge(9L, 8L, 1),
                   Edge(8L, 10L, 1), Edge(10L, 9L, 1), Edge(1L, 11L, 1)
                  )

val followsRDD =  sc.parallelize(follows)

val defaultUser = User("NA", 0)


val socialGraph = Graph(usersRDD, followsRDD, defaultUser)
```

عملیات گراف

- تبدیل خاصیت
- تبدیل ساختار
- پیوند
- محاسبات موازی گراف
- الگوریتمهای پردازش گراف

عملیات گراف

- **تبدیل خاصیت**
- در این عملیات صفات یالها یا رئوس گراف را تغییر می کند
- **مثلاً**
- mapVertices
- برای تغییر صفات رئوس

عملیات گراف

- **تبدیل ساختار**
- در این عملیات ساختار گراف تغییر می کند
- **مثلاً**
- reverse
- برای تغییر جهت تمام یالهای گراف

عملیات گراف

- **پیوند**
- در این عملیات امکان بروزرسانی صفات رئوس یا اضافه کردن صفت به رأس
- **مثلاً**
- joinVertices
- برای بروزرسانی صفات رئوس

عملیات گراف

- **محاسبات موازی گراف**
- امکان پیاده سازی الگوریتمهای تکراری برای پردازش گرافهای بزرگ

- pregel

عملیات گراف

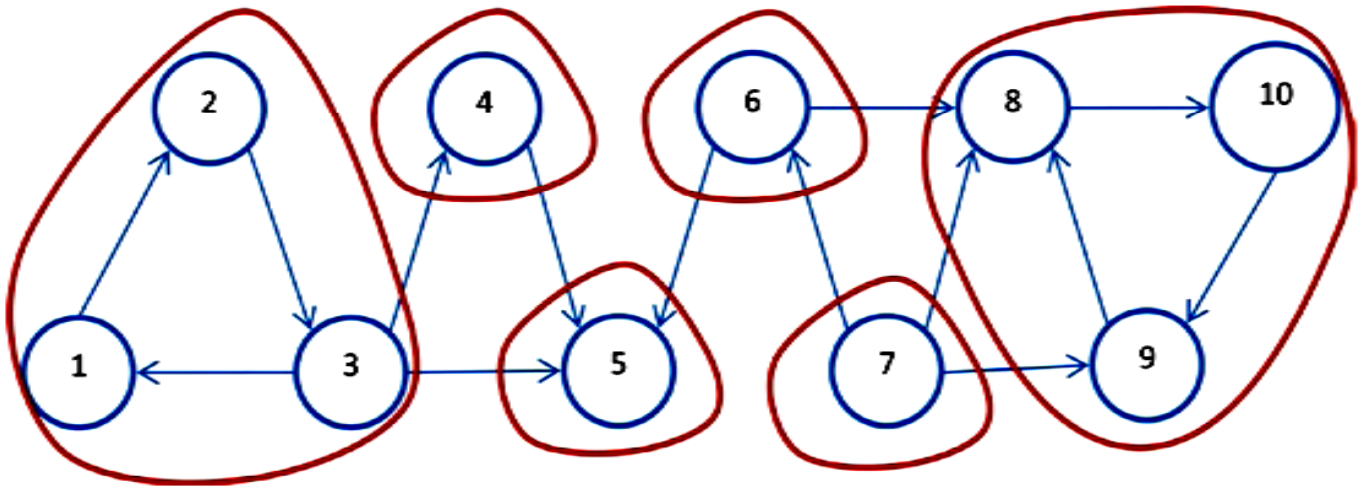
- **الگوریتمهای پردازش گراف**
- الگوریتمهایی که عملیاتی خاصی روی گراف انجام می دهند

- **مثلاً**

- stronglyConnectedComponents
- زیرگرافی که شامل رئوسی است که توسط تمام رئوس دیگر زیرگراف قابل دسترسی باشد

عملیات گراف

- مثلاً
- stronglyConnectedComponents
- زیرگرافی که شامل رئوسی است که توسط تمام رئوس دیگر زیرگراف قابل دسترسی باشد



یادگیری ماشین در اسپارک

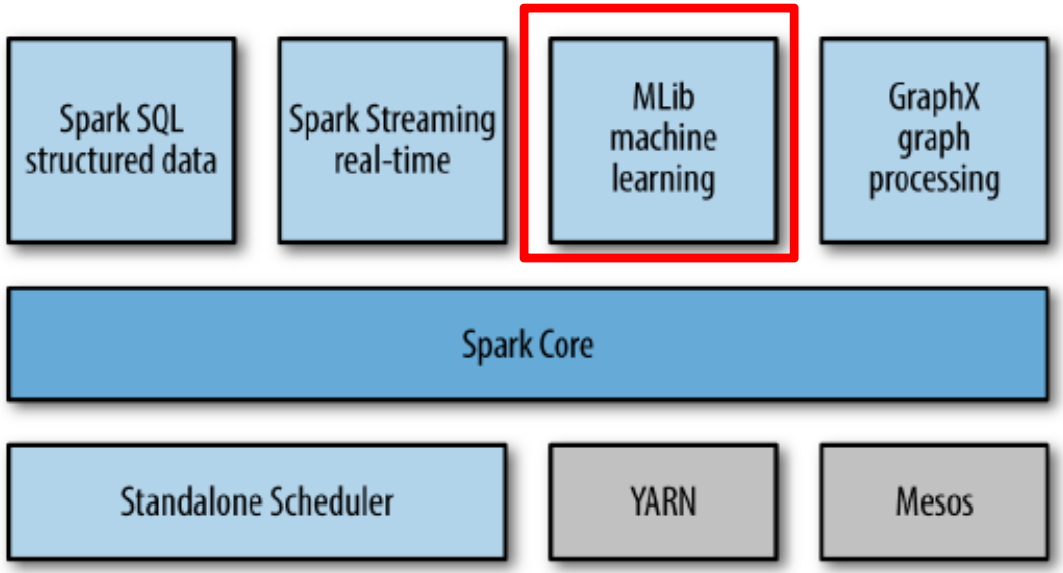
Spark

MLlib

Spark
MLlib

مقدمه

- یکی از بخشهای اسپارک کتابخانه ای برای یادگیری ماشین، که در آن مجموعه ای از الگوریتمهای یادگیری ماشین پیاده سازی شده است



انواع داده ها در اسپارک

- در اینجا هم از بردار استفاده می شود

```
denseVec1 = array([1.0, 2.0, 3.0])  # NumPy arrays can be passed directly to MLlib
denseVec2 = Vectors.dense([1.0, 2.0, 3.0]) # .. or you can use the Vectors class
```

```
sparseVec1 = Vectors.sparse(4, {0: 1.0, 2: 2.0})
sparseVec2 = Vectors.sparse(4, [0, 2], [1.0, 2.0])
```

Spark MLlib

استخراج ویژگی

- الگوریتم TFIDF برای داده های متنی
- امکان مقیاس گزاری و نرمال سازی ویژگی ها
- تبدیل کلمات به بردار (Word2Vec)

Spark MLlib

ابزارهای آماری

- محاسبه آماره ها (حداقل، حداکثر، میانگین، واریانس)
- محاسبه همبستگی
- آزمونهای آماری

Spark MLlib

الگوریتمهای یادگیری

- رگرسیون (رگرسیون خطی و لگاریتمی)
- طبقه بندی (ماشین بردار پشتیبان، بیز، درخت تصمیم)
- خوشه بندی (k-means,..)
- کاهش ابعاد (PCA,SVD)