Discretized Streams: Fault-Tolerant Streaming Computation at Scale Abstract Many “big data” applications must act on data in real time. Running these applications at ever-larger scales	Các luồng rời rạc: Điện toán luồng dữ liệu có khả năng kháng lỗi cao hoạt động trên mọi kích thước hệ thống Tóm tắt Nhiều ứng dụng “dữ liệu lớn” phải xử lý dữ liệu trong thời gian thực. Việc chạy những ứng dụng trên những hệ thống có kích thước

requires parallel platforms that automatically handle faults and stragglers. Unfortunately, current distributed stream processing models provide fault recovery in an expensive manner, requiring hot replication or long recovery times, and do not handle stragglers. We propose a new processing model, discretized streams (D-Streams), that overcomes these challenges. D-Streams enable a parallel recovery mechanism that improves efficiency over traditional replication and backup schemes, and tolerates stragglers. We show that they support a rich set of operators while attaining high per-node throughput similar to single-node systems, linear scaling to 100 nodes, sub- second latency, and sub-second fault recovery. Finally, D-Streams can easily be composed with batch and interactive query models like MapReduce, enabling rich applications that combine these modes. We implement D-Streams in a system called Spark Streaming.	ngày càng tăng cần các platform song song tự động xử lý lỗi và các straggler (các nút chậm). Tuy nhiên, các mô hình xử lý luồng phân tán hiện nay có cơ chế khắc phục lỗi đắt tiền, cần sao chép nóng hoặc thời gian khắc phục lâu, và không thể xử lý được các straggler. Chúng tôi đề xuất một phương pháp xử lý mới, các lường rời rạc (D-Streams), có khả năng khắc phục được những nhược điểm trên. D-Streams có cơ chế phục hồi song song nên hiệu suất cao hơn các phương pháp sao chép và phục hồi truyền thống, và có thể chịu được các straggler. Chúng tôi thấy rằng phương pháp này hỗ trợ rất nhiều phép toán nhưng vẫn duy trì được lưu lượng tin trên mỗi nút cao tương đương với các hệ đơn nút, có khả năng mở rộng tuyến tính đến 100 nút, độ trễ dưới một giây, thời gian khắc phục lỗi dưới một giây. Cuối cùng, D-Streams có khả năng kết hợp với các mô hình truy vấn theo lô và truy vấn tương tác chẳng hạn như MapReduce, tạo điều kiện triển khai nhiều ứng dụng kết hợp các mô hình này. Chúng tôi thực thi D-Streams trong hệ phân tích thời gian thực.
1 Introduction Much of “big data” is received in real time, and is most valuable at its time of arrival. For example, a social network may wish to detect trending conversation topics in minutes; a search site may wish to model which users visit a new page; and a service operator may wish to monitor program logs to detect failures in seconds. To enable these low-latency processing applications, there is a need for streaming computation models that scale transparently to large clusters, in the same way that batch models like MapReduce simplified offline processing.	1.Giới thiệu Đa phần “các dữ liệu lớn” được tiếp nhận trong thời gian thực, và dữ liệu này có giá trị nhất tại thời điểm nhận. Ví dụ, mạng xã hội cần phát hiện chủ đề trò chuyện yêu thích trong vài phút gần đây; một trang web tìm kiếm cần biết cách thức truy cập trang mới của những người dùng; và một nhà điều hành dịch vụ muốn theo dõi các bản ghi chương trình để phát hiện lỗi trong vài giây. Để kích hoạt các ứng dụng xử lý độ trễ thấp này, chúng ta cần các mô hình tính toán luồng có thể dễ dàng mở rộng cho các cụm lớn, hoạt động theo cơ chế giống như các mô hình xử lý theo lô chẳng hạn như mô hình xử lý offline đơn giản hóa MapReduce.

Designing such models is challenging, however, be-cause the scale needed for the largest applications (e.g., realtime log processing or machine learning) can be hundreds of nodes. At this scale, two major problems are faults and stragglers (slow nodes). Both problems are inevitable in large clusters [12], so streaming applications must recover from them quickly. Fast recovery is even more important in streaming than it was in batch jobs: while a 30 second delay to recover from a fault or straggler is a nuisance in a batch setting, it can mean losing the chance to make a key decision in a streaming setting.	Tuy nhiên, việc thiết kế các mô hình như thế gặp nhiều khó khăn vì nhiều ứng dụng lớn (chẳng hạn như xử lý bản ghi thời gian thực hoặc ứng dụng học máy) có thể cần đến hàng trăm nút. Với quy mô lớn như vậy, hai khó khăn chính sẽ phát sinh là lỗi và các straggler (các nút chậm). Đây là hai vấn đề không thể tránh được trong các cụm lớn [12], vì vậy các ứng dụng luồng phải có khả năng phục hồi sau những lỗi này một cách nhanh chóng. Phục hồi nhanh trong phương pháp luồng thậm chí còn quan trọng hơn trong phương pháp xử lý theo lô: chậm phục hồi lỗi hoặc straggler 30 giây trong phương pháp xử lý theo lô có thể gây phiền toái, còn trong phương pháp luồng độ trễ như vậy có thể làm mất cơ hội đưa ra quyết định quan trọng.