Handling fault tolerant of large scale streaming system

Background

After the first study of “partitioning techniques for CEP”, I understand the basic techniques to partition the CEP, or to scale out the CEP, but I want to know more about what people are trying to do with these techniques. The first hot topic is "Fault-tolerant", which is the focus of this report.

> comments:
>
> Is it valuable to add RDD into CEP engine? If yes, then how?

Parallel recovery

stragglers

reference material:

1. [MapReduce:Simplified data processing on large clusters, OSDI'04]
2. [MapReduce:Simplified Data Processing on Large Clusters.08]

If a node is available but is performing poorly,a condition that we call a straggler, MapReduce runs a **speculative copy** of its task (also called a “backup task”) on another machine to finish the computation faster. Without this mechanism of speculative execution, a job would be as slow as the misbehaving task. Stragglers can arise for many reasons, including faulty hardware and mis-configuration. Google has noted that speculative execution can improve job response times by 44% [1] [2].

Due with stragglers on Streaming

1. [Discretized