**Review of Storm**

# **Summary**

Storm is currently being used to run various critical computation in Twitter ar scale, and its-real time.

Considering twitter users making billions of complicated decisions in real-time, many modern data processing environments require processing complex computation on streaming data. Although Hadoop is the most popular one to deal with big data problems, it just exists for batch processing. Storm focus on dealing with stream data flow, providing a new data computation model.

It is currently being used to run various critical computations in Twitter at scale and in real-time. It is for distributed scale-out and fault-tolerance. Storm powers the real-time stream data management tasks that are critical to provide Twitter service.

Storm is an open-source and a computation system. It doesn’t have storage conception, which make it possible to handle date in real time no matter what kind of data is put in.

**Contributions**

1. **Scalable :** easily add or remove nodes from the Storm cluster without disrupting existing data flows.
2. **Resilient :** the storm cluster must continue processing existing topologies with a minimal performance impact.
3. **Extensible:** storm topologies may call arbitrary external functions and thus needs a framework that allows extensibility.
4. **Efficient :** used in real-time applications, storm must keep all its storage and computational data structures in memory.
5. **Easy to administer** : quickly point out the source problems as they arise. Thus, easy-to-use administration tools are a critical part of the requirements.

**Comments**

**Limitations: ①**Though if the Nimbus service fails, the workers still continue to make forward progress, users cannot submit new topologies.②lack of declarative query paradigm for storm that still allows easy extensibiliy.