# CLOUD COMPUTING APPLICATIONS

## Motivation for Spark

Roy Campbell & Reza Farivar

# Motivation

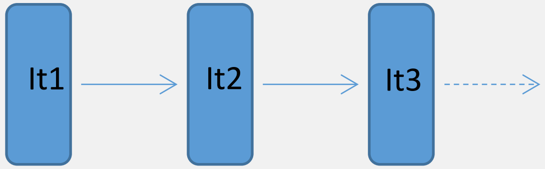
- Iterative algorithms and interactive data exploration are commonly used in many domains
- Traditional MapReduce and classical parallel runtimes cannot solve iterative algorithms efficiently
  - Hadoop: Repeated data access to HDFS, no optimization to data caching and data transfers
  - MPI: no natural support of fault tolerance and programming interface is complicated

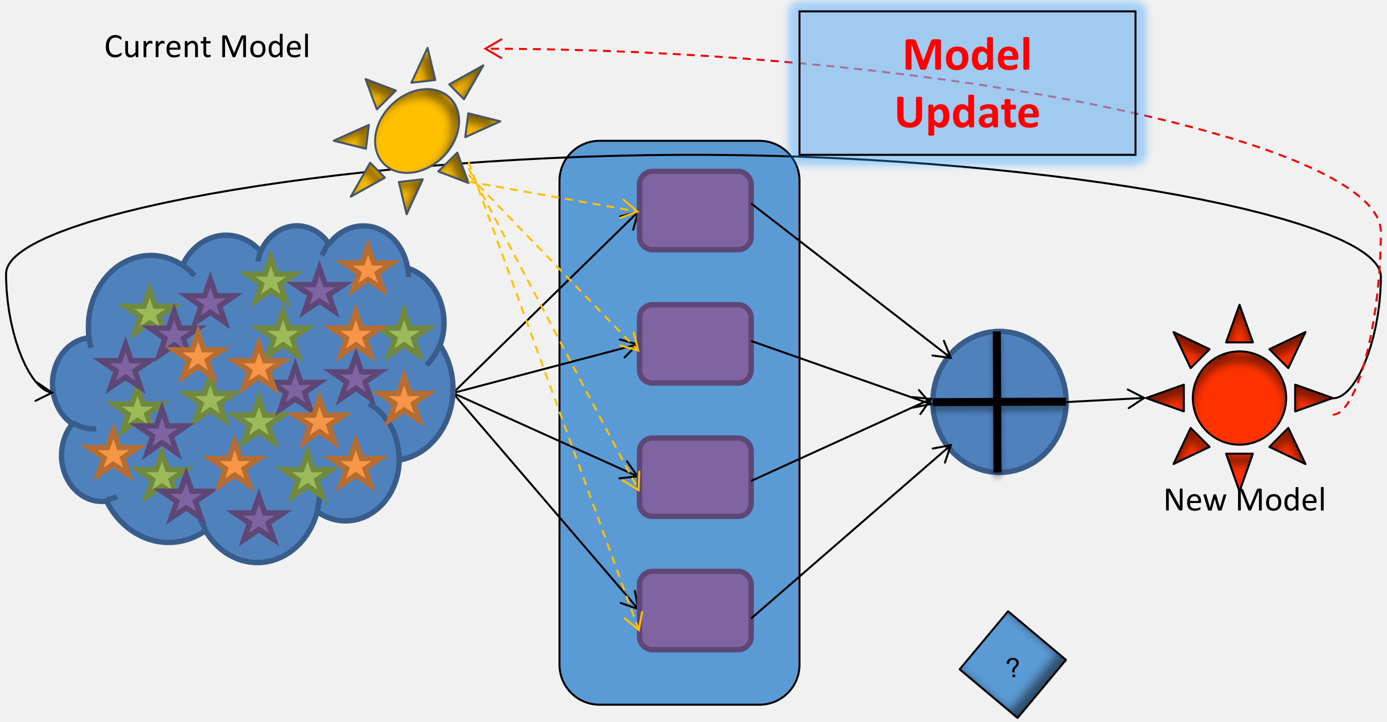
# Retrofitting Iterations on MR

- MR does not support iteration out of the box
- But we still want Page Rank, clustering etc.
  - Mahout
  - Nutch
- The “obvious” solution: Split iteration into multiple MapReduce jobs. Write a driver program for orchestration.

# A MapReduce Implementation of I.C.

```
Iterate {
   Map: for (each) i = 1 to M
      Compute();
   Reduce();
} Until converged();
```





- Repeated reads of constant (input) data in each iteration
- Run-time overheads in each iteration
- Intermediate Communication resulting from model updates
- Model Update Traffic
- Granularity of parallelism limited by iteration