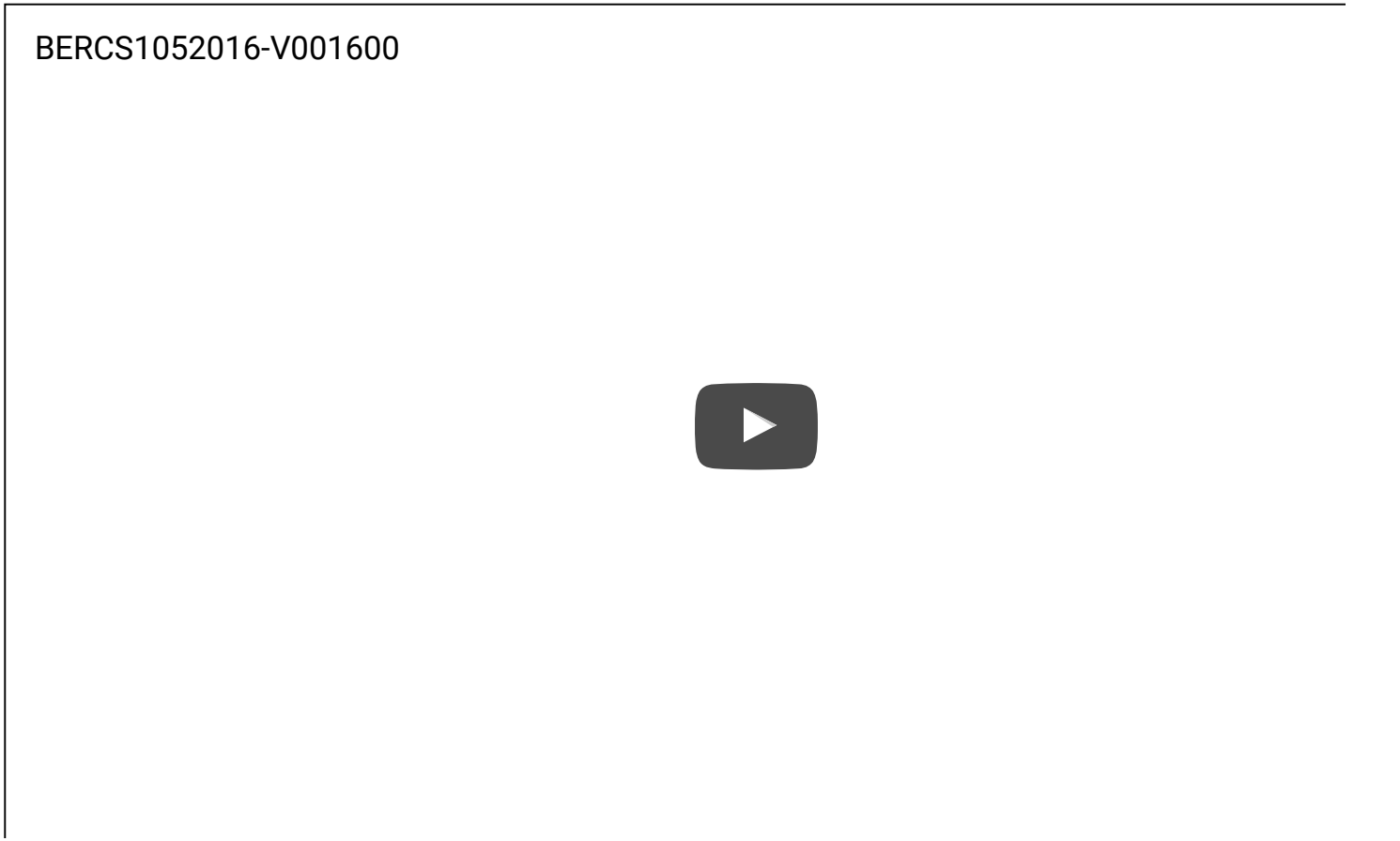
Bookmark

Cluster Computing Challenges and the Map Reduce Programming Paradigm

BERCS1052016-V001600



0:00 / 7:33 1.0x

Download video Download transcript .srt

Using Cluster Computing for Big Data

(1/1 point)

Which of the following properties does modern cluster computing have:

- [ ] Uses premium hardware
- [x] Uses consumer grade hardware ✔
- [ ] Uses complex hardware
- [x] Uses complex software ✔
- [x] Easy to add capacity ✔

✔

Note: Make sure you select all of the correct options—there may be more than one!

EXPLANATION

Modern cluster computing is based on less expensive, consumer grade hardware which makes it easy to grow capacity. Complex software is used to handle any problems, instead of hardware.

You have used 1 of 4 submissions

Using Divide and Conquer

(1/1 point)

What are some of the challenges of using divide and conquer:

- [x] Moving data is very expensive ✔
- [ ] Using a single machine is faster than multiple machines
- [x] Having many machines means having to deal with many failures ✔
- [x] Having many machines means having to deal with slow machines ✔
- [ ] Using hash tables for very large documents works well

✔

Note: Make sure you select all of the correct options—there may be more than one!

EXPLANATION

When using divide and conquer, you have to consider network and data locality because moving data between machines is expensive. Even with a low per-machine failure rate, using many machines means that several will fail per day. As machines age, they may fail in ways that cause slow performance (e.g., a failing disk drive that retries each read or write operation multiple times before successfully completing). With divide and conquer-based computing, the minimum time to complete a computation will depend on the slowest machine.

You have used 1 of 4 submissions

Map Reduce deals with failures and slow tasks by re-launching the tasks on other machines. This functionality is enabled by the requirement that individual tasks in a Map Reduce job are *idempotent* and have *no side effects*. These two properties mean that given the same input, re-executing a task will always produce the same result and will not change other state. So, the results and end condition of the system are the same, whether a task is executed once or a thousand times.

MapReduce

(1/1 point)

Which of the following problems does a MapReduce implementation handle?

- [x] Recovering from machine failures ✔
- [x] Shuffling data between the Map and Reduce functions ✔
- [x] Running the Map and Reduce functions on many machines ✔
- [ ] Automatically parallelizing an algorithm
- [x] Recovering from slow machines ✔

✔

Note: Make sure you select all of the correct options—there may be more than one!

EXPLANATION

Map Reduce handles the execution of Map and Reduce functions on many machines including the shuffling of data between Map and Reduce functions. It also automatically recovers from both machine failures and slow machines. However, it is up to you to parallelize your algorithm.

You have used 2 of 4 submissions