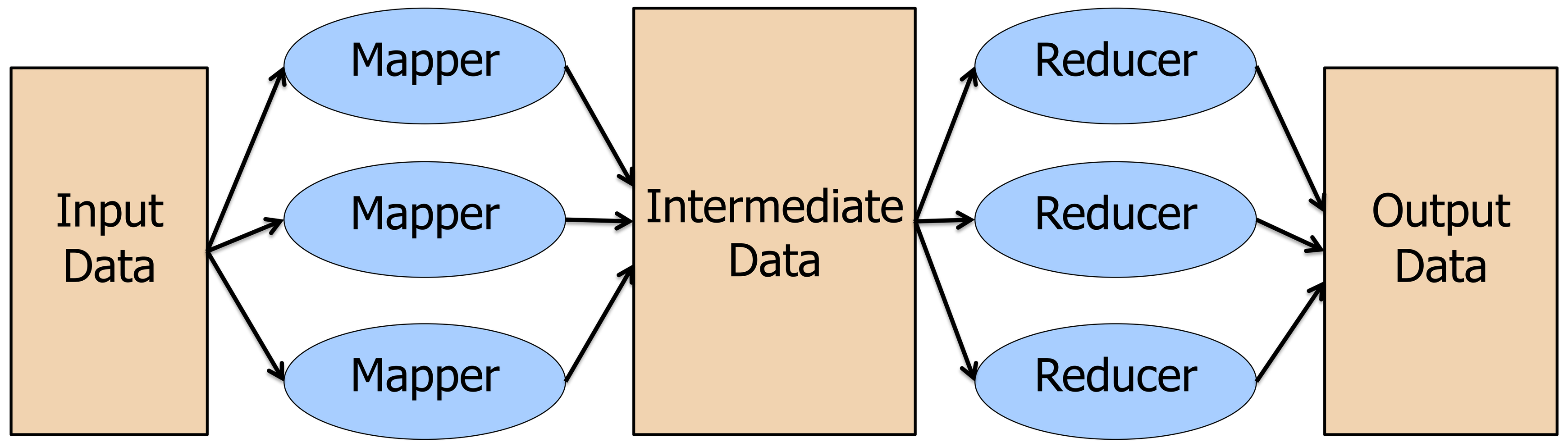
# Bringing Elastic MapReduce to Scientific Clouds

Pierre Riteau [1,2], Kate Keahey [3,4], Christine Morin [2]
[1] Université de Rennes 1, IRISA [2] INRIA Rennes – Bretagne Atlantique
[3] Argonne National Labs [4] University of Chicago Computation Institute

## Introduction

- The **MapReduce** programming model proposed by Google offers a simple way to perform distributed computation over large data sets. Input data is split in chunks serving as input for a map function. The intermediate data produced by the map function is reassembled by a reduce function to produce the result of the computation.




- The Apache **Hadoop** project develops a **free** and **open-source** implementation of the **MapReduce framework** together with the **HDFS distributed file system** used to store data.

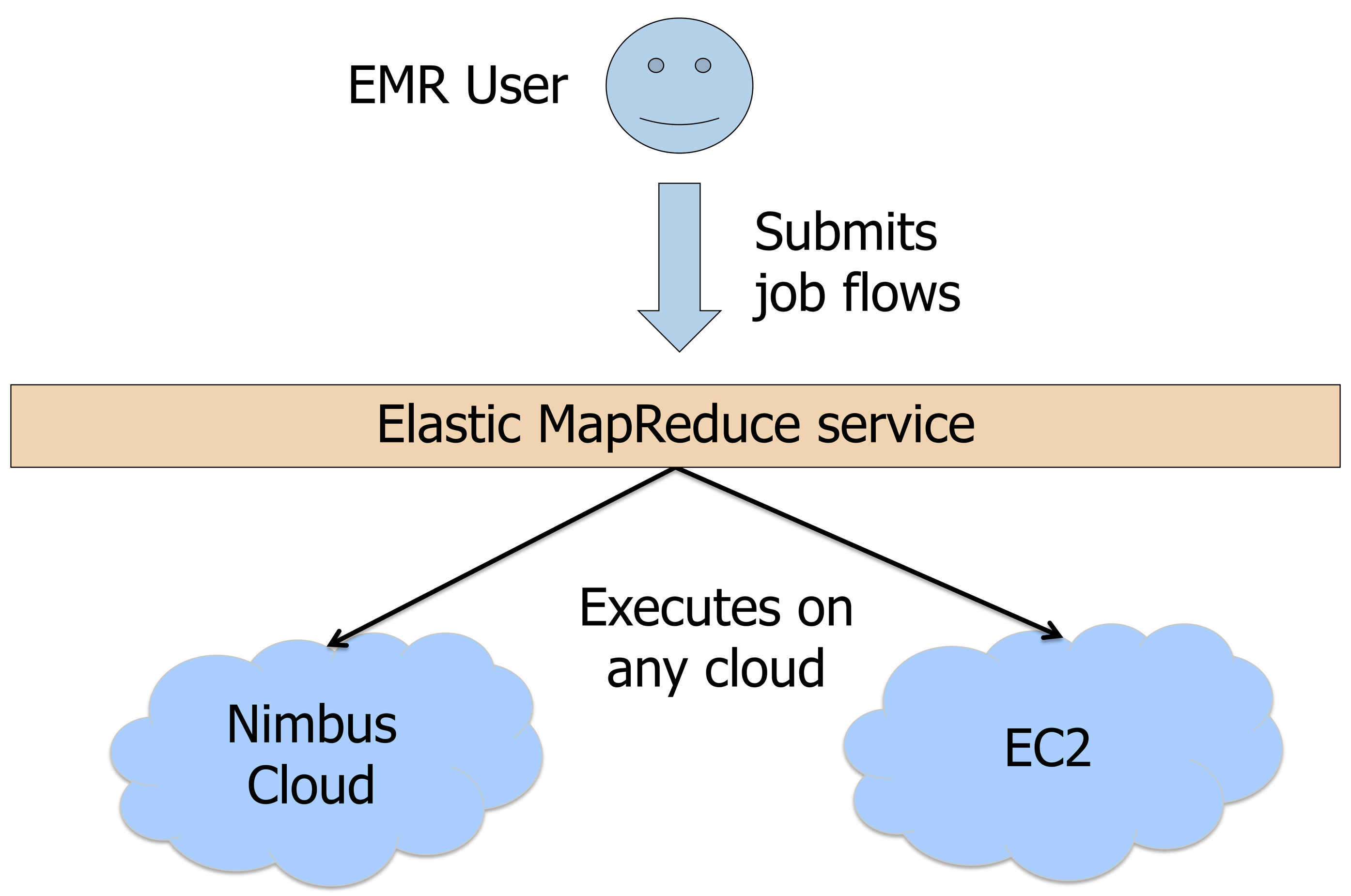
## Amazon Elastic MapReduce

- Amazon **Elastic MapReduce** (EMR) is a service offered by the Amazon Web Services platform. It allows users to submit sequences of MapReduce jobs called **job flows**, using a web interface, a command line tool or an API.


- Input and output data is stored in **Amazon S3**, a fully redundant data storage infrastructure. Elastic MapReduce takes care of **provisioning** a Hadoop cluster on Amazon EC2, performs **configuration** and **tuning**, execute job flows and improves **fault tolerance** by monitoring virtual machines and restarting failed ones. It also supports **dynamically resizing** Hadoop clusters.

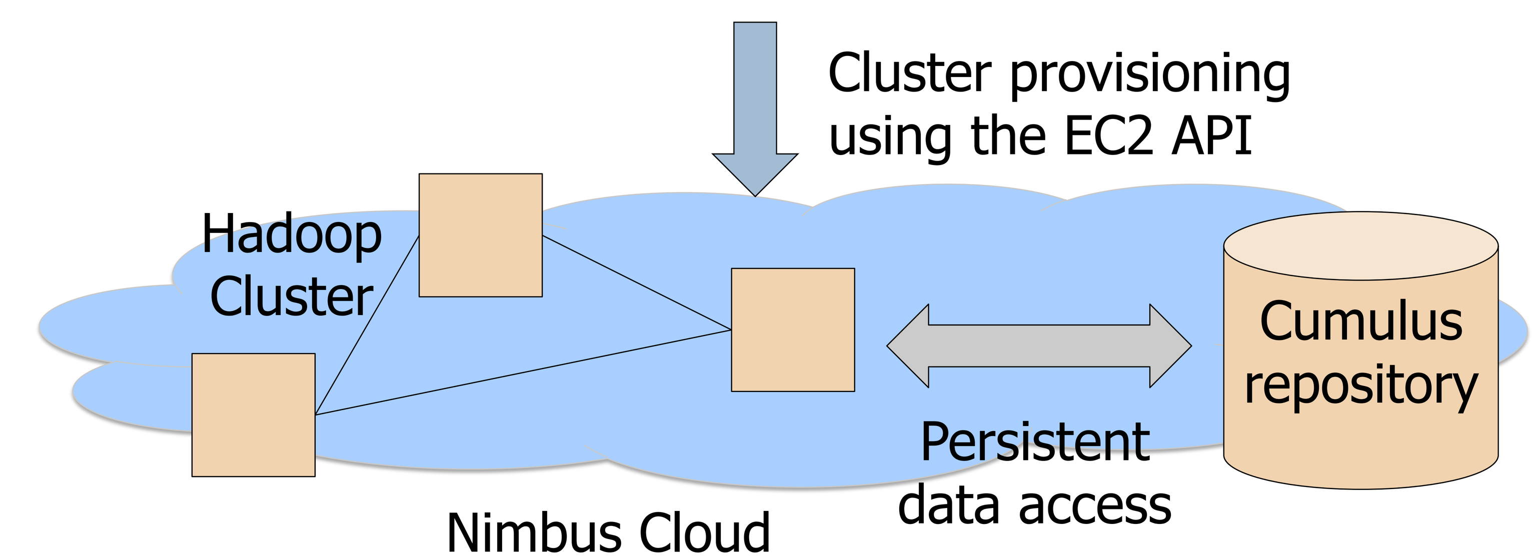
## Our Elastic MapReduce

- Amazon Elastic MapReduce is a powerful and useful tool, but it is a **closed** platform **restricted to Amazon EC2** resources.
- We aim to bring an Elastic MapReduce platform **to scientific clouds** compatible with Amazon EC2, such as those based on open-source implementations like **Nimbus**, **OpenNebula**, and **Eucalyptus**. It will of course work with EC2 as well.



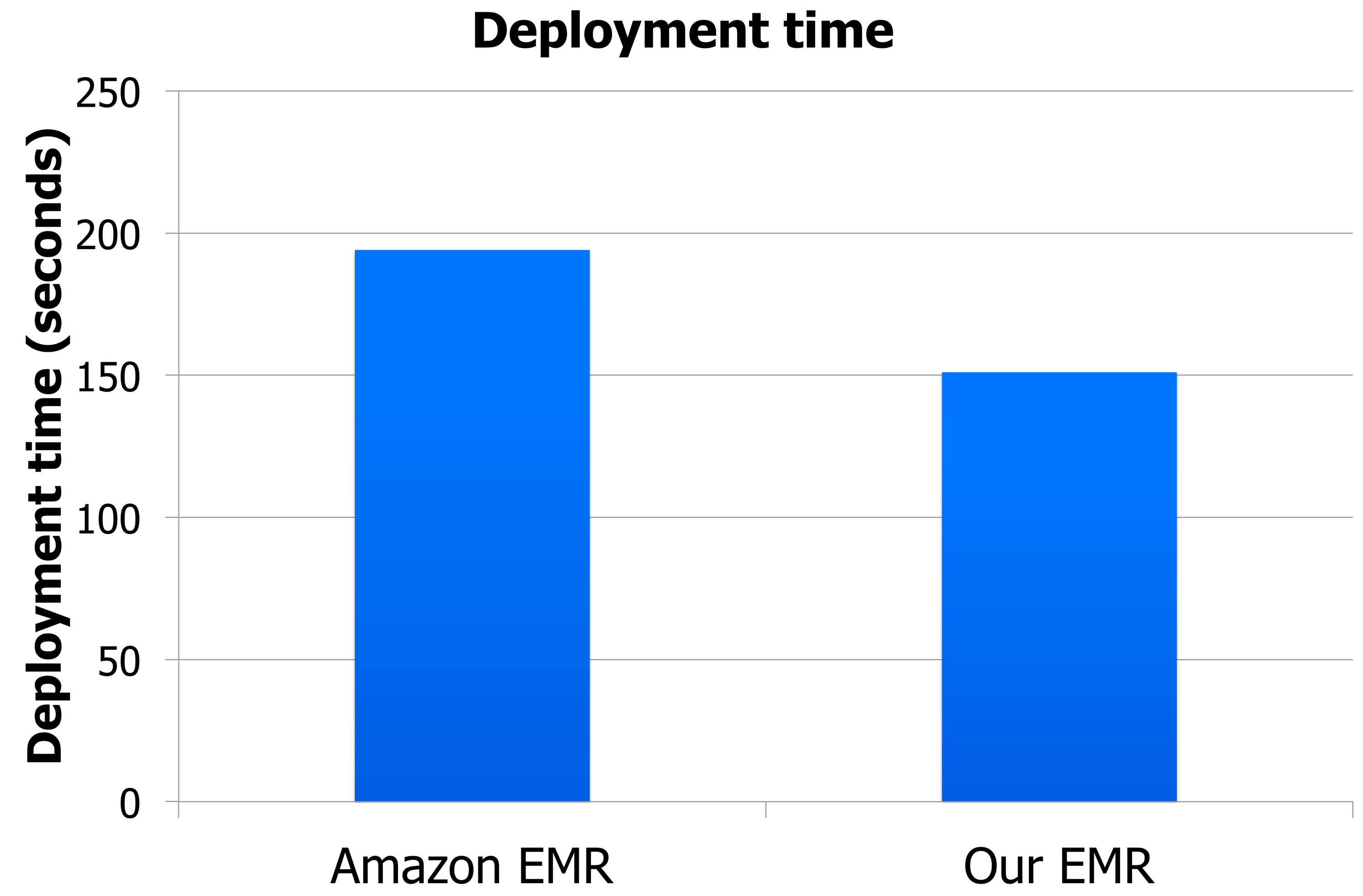

## Implementation

- Our Elastic MapReduce implementation is written in **Ruby** using the **Sinatra** web framework. It uses the **EC2 API** to provision machines on an EC2-compatible cloud.
- We modified **Hadoop** to add support for accessing **Cumulus** storage. Cumulus is an **implementation of S3** developed in the Nimbus project.




## Evaluation

- To evaluate our Elastic MapReduce implementation, we use a scientific computation based on the **CloudBurst** algorithm.
- CloudBurst is a new parallel read-mapping algorithm optimized for mapping next-generation sequence data to the human genome and other reference genomes.
- We execute a CloudBurst sample job flow using 3 c1.medium Amazon EC2 instances (one master and two slaves), and compare deployment time with 3 VMs provisioned from a Nimbus cloud using resources from the Grid'5000 testbed.




## Conclusion

- Elastic MapReduce allows users to process **massive amounts of data** in the cloud without taking care of cluster provisioning, configuration, data staging: **they can focus on solving their problem**.
- Our Elastic MapReduce implementation will not be limited to replicate Amazon EMR functionality. Future work includes:
  - ✓ **performance comparison** of different types of instances,
  - ✓ **dynamic resizing** of clusters according to job flow **deadlines**,
  - ✓ **spot instances** usage on EC2 and Nimbus,
  - ✓ **multi-cloud** MapReduce job flows.

Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).