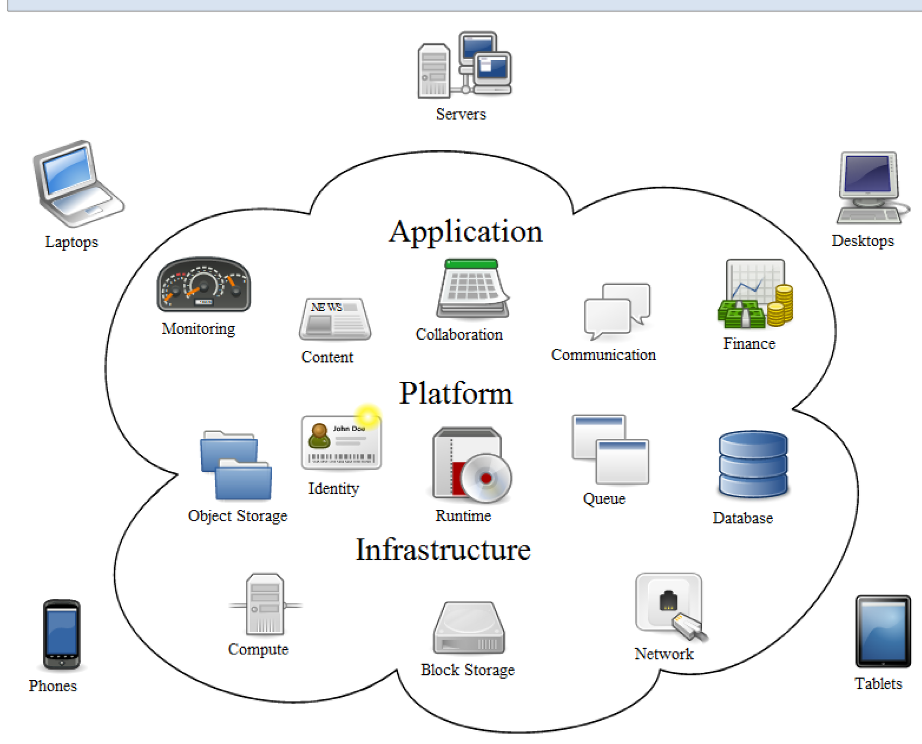
Modeling and Control of MapReduce Systems

BEREKMERI, Mihaly, GIPSA-lab, mihaly.berekmeri@gipsa-lab.grenoble-inp.fr
BOUCHENAK, Sara, LIG, sara.bouchenak@imag.fr
BOGDAN, Robu, GIPSA-lab, bogdan.robu@gipsa-lab.fr
MARCHAND, Nicolas, GIPSA-lab, nicolas.marchand@gipsa-lab.fr

Systems based on the MapReduce programming model are emerging as a central tool for deploying jobs that process large datasets in parallel. However the configuration of MapReduce systems is a complex process and at the moments it's left up to the user. These ad-hoc configuration methods make it difficult for small companies to take advantage of the growth of cloud computing solutions that provide resources as a service. Furthermore, the definition of SLAs becomes a complicated process for the user and the service provider as well. We propose a control theoretical approach to solving these problems. This implies the development of a general model that captures the dynamics of MapReduce systems. Finally, we intend to provide novel control methods that ease the configuration process and guarantee service level objectives such as constraints on system performance (execution times) and dependability (latency, availability) while optimizing resource consumption.

I. Context and Problem Statement.

As our world is becoming ever more digitalized companies have more and more data to process. This **steep increase of the unstructured data available is asking for a shift of perspective** from the traditional database approach to an efficient distributed computing platform specifically designed for handling petabytes of unstructured information available in jobs such as: personalized advertising, advanced data mining or classification.



Cloud computing, using the pay as you go model, offers an enticing new approach to deploying such task**.** The dominant new perspective emerging, in large scale data processing, is the programming model called MapReduce.

II. Cloud computing

Cloud Computing is **next milestone** of **IT evolution**. It incorporates the long standing dream of providing **computing as a service** over the internet. Cloud computing can offer many computing forms as a service such as: hardware, platform, software, storage**. Infrastructure as a service** is one of the most wide spread service models at the moment and it offers computers as a service. **Platform as a service** models offer computing frameworks such as programming models and languages, operating systems**. Software as a service** provides an environment for running end user applications in the cloud.

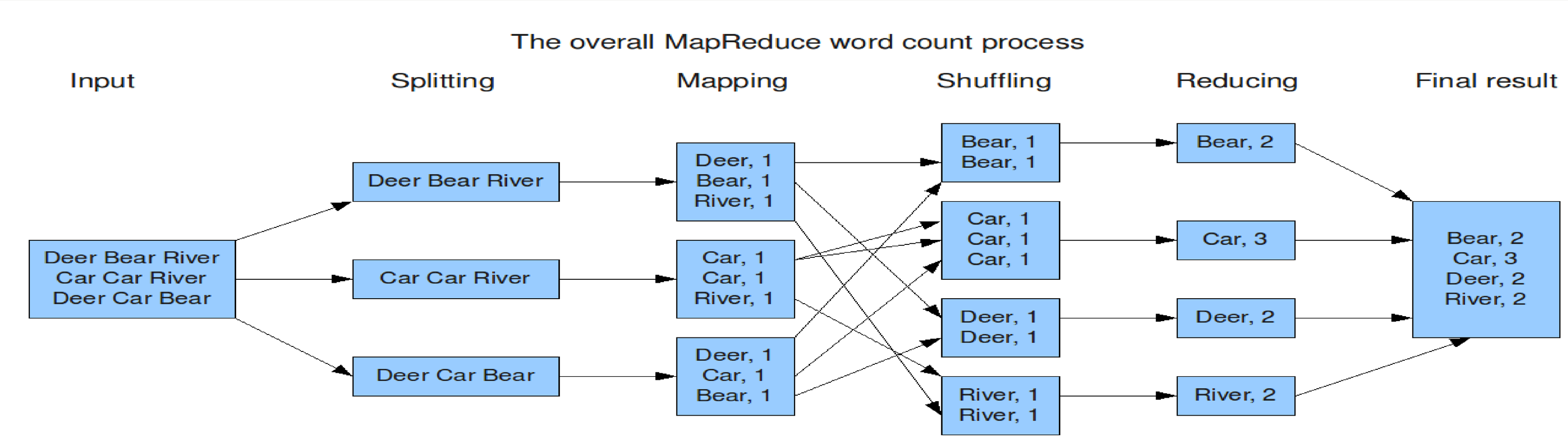
Scalability and **elasticity** are a few of the key aspects of Cloud computing. Seemingly resources scale up infinitely on demand. In the background, cloud computing takes care of such non trivial problems as data security and availability, multitenancy issues, virtualization, hardware and software maintenance.

III. MapReduce

MapReduce is a programming model especially **designed for large scale unstructured data processing**. Its success lies in its **simplicity, scalability and fault-tolerance**. The runtime environments, which build upon the MapReduce framework, automatically take care of:

- data partitioning
- replication, data consistency
- load balancing and fault tolerance
- task distribution and scheduling

Only two functions have to be implemented: the **Map function** and the **Reduce function**. The Map function takes an input set of (key,value) pair and outputs an intermediate (key,value) pair. The MapReduce framework automatically groups all the values associated with the same keys and forwards it to a Reduce function. The Reduce function processes these values and usually gives as output a reduced set of values. Finally, from a user perspective, the MapReduce framework hides all the messy overhead of parallel computing and lets us focus on the task at hand.



IV. Current problems and development perspectives

Modeling issues: the models are specific to certain job metric, such as job finish time and don't consider latency, availability metrics.

- Most of the models don't account for any kind of faults.
- Randomness of such systems is not incorporated into current models.
- Ad-hoc configuration of MR framework poses many challenges.

SLA problems: a general framework for defining the terms of the Service Level Agreements, for MapReduce jobs, is needed.

- It is very difficult for the user to check that the constraints of the service level agreements are kept, difficult to differentiate between providers.
- Abusive jobs are becoming increasingly difficult to handle in virtualized environments where multiple jobs use the same cluster.

Control Insertion: novel control laws are needed that can enforce the Service Level Agreements, both performance and dependability metrics.

- More energy efficient control laws are desired.

V. Our goals

I. MapReduce Dynamic Model

- **Transfer the problems into the control theory framework**
- **The model has to ease the definition of service level objectives**
- **Account for uncertainties**
- **Include faults**

II. MapReduce Control Laws

- **Provide guarantees in terms of performance, dependability and cost**
- **Optimize the system configuration**
- **Minimize deployment costs**
- **Handle multitenancy**
- **Test it on real life systems, such as AmazonEc2, Grid500**

Grenoble Images Parole Signal Automatique
UMR CNRS 5216 – Grenoble Campus
38400 Saint Martin d'Hères - FRANCE