*Originally published in French (14/11/2017), this post is a translation made with the help of: www.DeepL.com/Translator*

# BIG DATA: A NEW FORM OF COLLECTIVE INTELLIGENCE (3B)

## Mini-Series of Posts #3 : From search engines to Big Data. And, what about conflict resolution?

## Post #3B – *Hadoop - a distributed architecture, a new infrastructural paradigm*!

The genesis of Big Data lies in the initial need to improve search engines efficiency. This link, named Hadoop, constitutes the core of this mini-series of Posts #3, whose objective is double: to define the notion of Big Data, on the one hand, and to clarify the impact of these innovations on conflict resolution methods, on the other hand.

### Distributed architecture vs. centralized architecture

Hadoop is exclusively installed on a *cluster computing*[i]. A cluster is first and foremost a kind of distributed architecture, as opposed to an architecture centralized around a server. Here lies precisely the paradigm shift. Our architecture model is being changed from centralized architectures to distributed architectures.

Several reasons explain the need for this change. Notably, the fact that in centralized architectures bottlenecks in the network are frequent. Furthermore, these architectures are not suitable for analytical processing. The growth in data volume increases the computing load at the central server level, and therefore the latency time, i.e. the time taken by a processing task to complete; the latter turns out to be prejudicial for uses based on quasi-real-time responses[ii].

## Distributed architectures

Juvénal CHOKOGOUE, our author of reference, makes a distinction among the types of distributed architectures. He considers that there are "false" distributed architectures: client/server architectures (or off-site architectures). They do not represent "real" distributed architectures because they *"simply transfer the processing load to another workstation, while in a distributed architecture the processing is divided into subtasks, then these subtasks are distributed for processing to a set of computers that are seen as a single central server. The philosophy is really not the same philosophy*"[iii]. He deems that cluster computing (several computers addressing the same problem) and grid computing (one or more different problems are shared between several computers dealing with them independently) form the "real" distributed architectures[iv]. Since Hadoop must be installed on a cluster computing architecture, we will focus our attention on this model and its specificities in order to understand the search engines improvement, its link with Big Data, and ultimately the possible sociological impact of this invention, in particular on the conflict resolution modes.

A computer cluster is defined as the grouping of tens, hundreds, or even thousands of traditional PCs interconnected by network cables using switches (physical link). Several racks, each containing between 8 and 64 nodes (computers), form the cluster, which can be huge as shown, for example, in the photo opposite[v]. Node resources, differently shared depending on the cluster, are the following: RAM memory, hard disk, CPU processor. In the Hadoop cluster, in terms of resources sharing, each node has its own hard disk, its own memory, its own processor. We are talking about *shared-nothing* cluster architecture. As its name suggests, "nothing is shared," each node is self-sufficient and uses its own resources to do its job. There are other ways to share resources within a cluster, but presenting them would exceed the objectives of this mini-series of Posts #3, especially since *shared-nothing* is the default mode for resources sharing of a cluster, a mode used in many Big Data problems[vi].

If I dare the comparison, in cafeterias it is frequent to find racks where to deposit used dishes. The racks represent the racks, the trays would be the nodes, and all the racks would form the cluster.

## What communication inside the cluster: is there a pilot in the Hadoop plane?

*Shared-nothing* means that cluster nodes do not share their resources. They are independent from one another. Since the cluster is seen by the user as a single computer, how do the nodes communicate with one another? There is a conductor, a task orchestrator which plans and manages the cluster's activities, all the communication that takes place there. This software layer (virtual link) placed above the nodes effectively allows the user to work with the cluster as if it were a single computer. Ingenious, isn't it ?! In the Hadoop model, a master node (the reference node) distributes the tasks between cluster nodes (slaves) and manages redundancy/replication of data storage via the Hadoop Distributed File System (HDFS)[vii]. This is why this communication model is called master-slave, and differs from the peer-to-peer model in which the load of executing the request in the cluster belongs to all the nodes[viii].

In summary, the appropriate Hadoop architecture is a master/slave shared-nothing cluster[ix].

## The four characteristics and advantages of distributed cluster infrastructures compared to centralized architectures.

### 1) Horizontal scalability

The scalability of a computer system is defined as its ability to scale up, in other words to maintain its functionality and performances in the event of high demand. In application of Moore's famous law[x], which quite accurately predicted how the computing power of computers and the complexity of computer hardware would evolve, adding a node to a cluster (horizontal scalability) turns out to be less expensive than increasing the capacity of a central server (vertical scalability, uprizing)[xi]. Expressed for the first time in 1965 by engineer Gordon E. Moore, co-founder of microprocessor manufacturer Intel, Moore's law foresees, in its 1975 adjusted version, that *"the number of transistors of microprocessors (and no longer of simple less complex integrated circuits) on a silicon chip doubles every two years"*[xii]. And, Wikipedia to conclude: *"As a result, electronic machines have become less and less expensive and more and more powerful"*[xiii]. It is therefore understandable that opting for a cluster architecture is not an insignificant choice for a company. This architecture supports massively parallel processing, because it has the advantages of intensive computing, and also makes it possible to take advantage of the returns to scale generated by lower computer costs; this contributes to the adoption of Hadoop in companies. Thus, managing the growing volume of data, is done simply by increasing the nodes in the computer cluster[xiv].

"Scaling Up"; key concept to understand the evolution going from search engines improvement towards Big Data, as well as the popularity and choice of clusters to handle the data explosion we are witnessing today. *"Building a data center of 1000, 10000 machines or more is now much cheaper than uprizing a server that centralizes data management"*[xv].

### 2) Linear scalability

A computer system is said to be linearly scalable when its performance increases proportionally with the addition of new components. This definition also applies to a cluster. Linear describes the performance of linearly scalable architectures growing at a constant rate as the number of processors increases. The progressive addition of further nodes in the cluster (horizontal scalability) makes it possible to stabilize, over time, the processing time induced by the growth in data volume. The linear scalability of cluster computing, unique model of clustering computers that allows linear scalability, has an indisputable competitive advantage over non-scalable or non-linearly-scalable architectures whose performance eventually caps after adding a certain number of processors[xvi].

It is the sharing of shared-nothing resources that allows the cluster on which Hadoop is based to do intensive computing, because it guarantees the almost unlimited horizontal and linear scalability of the cluster, as well as fault tolerance and high availability[xvii].

### 3) Fault tolerance

A cluster's fault or failure tolerance refers to its ability to function despite failures, especially node failures. We also talk about resilience. It is obtained thanks to the redundancy of data in the cluster: each file block processed by each node is replicated in the cluster; in the case of

Hadoop, replication takes place three times by default, via the Hadoop Distributed File System (HDFS). Thus, when a node fails, the system hands over the baton to another node that contains a duplicate of the data processed by the failing node, and when resources are lacking, another computer with more resources takes over[xviii].

**4) High availability**

The availability of a system means that it is operational and able to respond to users' requests. Unlike centralized architectures in which the availability of the entire system relies entirely on the central server (single point of failure), distributed architectures offer the ability for the system to continue to operate despite failures. We have seen above the relay mechanism activated in case of failure. As all systems being in themselves somewhat available (hopefully), high availability is not a per se cluster characteristic, but rather the measurement of a characteristic. Hadoop is a high availability system. When it comes to a cluster, high availability consists in minimizing its level of interruption, but minimizing to what extent? The answer varies from field to field. Obviously there are contexts in which availability can have a life or death dimension: hospitals, aviation, for example. These business requirements are formalised in an SLA (*Service Level Agreement*). The following two principles govern the achievement of high system availability: the elimination of single points of failure in the system, already seen above, on the one hand, and the automatic detection of failures, on the other hand[xix]. If you want to delve deeper into this subject, are interested in the technical aspects and the five different ways of building a cluster to make it highly available, I invite you to read pages 41 to 43 of Juvénal CHOKOGOUE's book, *Hadoop: become operational in the world of Big Data.* Since this topic largely exceeds the questions raised in thisPost #3B, I will not dwell on it.

## Societal impact and conflict resolution?

Data flood processing requires a special type of architecture, notably to benefit from linear scalability: a cluster computing, a form of distributed architecture. This is an architectural paradigm shift. Former centralized architectures are outperformed by massive data processing architectures. Given the increasing and varied uses made of this data, technologies based on a distributed architecture underpin our digital society. Change is technological, but it is also deeply societal. So, are these new technologies horizontalizing, decentralizing forces? How might the modes of conflict resolution, in particular the vertical, centralized, asymmetrical, by definition, judicial mode, be affected by these decentralizing forces? How will the sociological principle linking justice and society - a model of society generating a corresponding mode of regulation[xx] - find application in the digital age? Beginning of answer in the last Post of this mini-series #3. Let's move step by step towards this discussion.

## There is also a software definition of a Hadoop cluster

In this Post #3B, I defined a Hadoop cluster from a mainly architectural, physical point of view. It is also possible to give a software definition of the Hadoop cluster; here it is: a Hadoop cluster is simply a cluster on which Hadoop has been installed, that is two elements : first, the set of JAVA implementation classes of the MapReduce algorithm and, second, the Hadoop Distributed System File (HDSF) as a file system on all the hard disks of the cluster nodes[xxi]. These notions will be the subject of the following post.

See you soon for Post #3C : *Hadoop - softwares : MapReduce and the Hadoop Distributed File system - a new programming paradigm !*

Anne-Sylvie Weinmann
[www.medialien.ch](http://www.medialien.ch)

*References & Notes :*

[i] CHOKOGOUE Juvénal, *Hadoop : devenez opérationnel dans le monde du Big Data*, St-Herblain, ENI, 2017, p. 63 (https://m.editions-eni.fr/livre/hadoop-devenez-operationnel-dans-le-monde-du-big-data-9782409007613#).

[ii] CHOKOGOUE Juvénal, op. cit., pp. 31, 33, 34, 117.

[iii] Translation from CHOKOGOUE Juvénal, op. cit., p. 36.

[iv] CHOKOGOUE Juvénal, op. cit., pp. 28, 31, 37, 63, 65, 71.

[v] Source : Grappe de serveurs, https://fr.wikipedia.org/wiki/Grappe_de_serveurs

[vi] CHOKOGOUE Juvénal, *op. cit.*, pp. 31, 35, 45ss (notamment 45, 46, 49), 72.

[vii] See Billet #3C (to come).

[viii] CHOKOGOUE Juvénal, *op. cit.*, pp. 34, 50, 51, 63.

[ix] CHOKOGOUE Juvénal, *op. cit.*, p. 65.

[x] Nota bene: Post #3B English version differs about Moore's Law from the French version.

[xi] CHOKOGOUE Juvénal, *op. cit.*, p. 39.

[xii] Translation of : *Loi de Moore*, https://fr.wikipedia.org/wiki/Loi_de_Moore

[xiii] Translation of : *Loi de Moore*, https://fr.wikipedia.org/wiki/Loi_de_Moore

[xiv] CHOKOGOUE Juvénal, *op. cit.*, pp. 21, 37-39, 63.

[xv] CHOKOGOUE Juvénal, *op. cit.*, p. 39.

[xvi] CHOKOGOUE Juvénal, *op. cit.*, pp. 38-39.

[xvii] Translation from CHOKOGOUE Juvénal, *op. cit.*, p. 63.

[xviii] CHOKOGOUE Juvénal, *op. cit.*, pp. 33, 40, 41, 111.

[xix] CHOKOGOUE Juvénal, *op. cit.*, pp. 33, 41, 43.

[xx] BONAFE-SCHMITT Jean-Pierre, *La médiation : une justice douce.* Paris, Syros-Alternatives, 1992, p. 180.

[xxi] CHOKOGOUE Juvénal, *op. cit.*, p. 63.