On the Exploration of Equal Length Cellular Automata Rules Targeting a MapReduce Design in Cloud

Arnab Mitra, Computer Innovative Research Society (CIRS), Howrah, India
Anirban Kundu, Netaji Subhash Engineering College, Kolkata, India
Matangini Chattopadhyay, School of Education Technology, Jadavpur University, Kolkata, India
Samiran Chattopadhyay, Department of Information Technology, Jadavpur University, Kolkata, India

ABSTRACT

A MapReduce design with Cellular Automata (CA) is presented in this research article to facilitate load-reduced independent data processing and cost-efficient physical implementation in heterogeneous Cloud architecture. Equal Length Cellular Automata (ELCA) are considered for the design. This article explores ELCA rules and presents an ELCA based MapReduce design in cloud. New algorithms are presented for i) synthesis, ii) classification of ELCA rules, and iii) ELCA based MapReduce design in Cloud. Shuffling and efficient reduction of data volume are ensured in proposed MapReduce design.

KEYWORDS

Cloud Computing, Equal Length Cellular Automata, Equal Length Data Blocks, Load-Reduced Data Processing, Mapreduce

INTRODUCTION

Modern days’ many of our daily life applications are based on Cloud Computing. Cloud computing is described as “computing as a service over the Internet” at “pay-for-use basis” (in “http://www.ibm.com/cloud-computing/what-is-cloud-computing.html”) with facility of on-demand delivery of requested service(s). Large volume of data (often in ‘terabyte’ (TB) volume) are processed in clouds and are alienated into smaller data chunks for distribution among processing units (http://www.cloudcomputingpatterns.org/Distributed_Application and “http://www.cloudcomputingpatterns.org/#cloud_application_architectures”). MapReduce computation is increasingly used in clouds for faster and parallel processing of large volume data (Shafer et al., 2010; Yang et al., 2011; Gangeshvari et al., 2012; Srirama et al., 2012; “http://searchcloudcomputing.techtarget.com/definition/MapReduce” and “http://www.cloudcomputingpatterns.org/Map_Reduce”). Input data is distributed parallelly among available computing units. MapReduce is typically composed of following components (“http://searchcloudcomputing.techtarget.com/definition/MapReduce”).

- **JobTracker (JT):** A node (server) is responsible for managing all jobs and resources in a cluster;
- **TaskTrackers (TT):** Agents positioned at each node (client) in cluster to execute map and trim down tasks;

DOI: 10.4018/IJCAC.2018040101
• **JobHistoryServer (JHS):** A component that keeps track of completed modules of job; typically implemented as a separate function or along with JT.

Fault-tolerance is an essential criterion for MapReduced computing; each node periodically updates its condition to a master node; otherwise, master node re-assigns jobs to other available computing units. Applications of MapReduce (Zhou et al., 2011; Koundinya et al., 2012; Padhy et al., 2013; and Yang et al., 2014), ‘Hadoop’ (Yamamoto et al., 2012; Yang et al., 2012; Lin et al., 2013; Liu et al., 2015; and Río et al., 2015) were described for big data processing in clouds. ‘Hadoop’ is a popular third-party deployment of MapReduce in distributed computing. ‘Hadoop’ provides a distributed computing architecture for big data. “Hadoop Distributed File System” (HDFS) is provided to store multiple copies of data blocks in available clusters of computing nodes. A file in HDFS is decomposed into chunks; each chunk is assigned to different working machine. Equal load distribution is an important criterion in MapReduced computation. Hence a cost-effective approach towards equally load distributed MapReduce may be beneficial in clouds.


Summary of the contributions in this research are as follows:

1. ELCA based MR design having low-cost physical implementation capability, is presented;
2. ELCA rules are explored for efficient MR design;
3. Equal load distribution and efficient data volume reduction are ensured;
4. New algorithms are designed for synthesis, classification of ELCA rules, and design for ELCA based MR.

Rest of the paper is organized as follows: brief discussions on CA and lexicographic combination are in preliminaries on CA and lexicographic combination section; related works are presented in next section; all ELCA rules are explored in set of ELCA rules section; ELCA based MR design and detailed discussions are presented thereafter; experimental results are followed next; finally, conclusions are drawn.

**PRELIMINARIES ON CA AND LEXICOGRAPHIC COMBINATION**

CA (Wolfram, 1986; Chaudhuri et al., 1997) are used as a discrete mathematical model for modeling of dynamic system. Number of cells are organized in the form of lattice, which evolve in discrete space and time. CA evolution may be achieved in several dimensions. One-dimensional three-neighborhood (dependency with self-cell with left and right neighbor) CA scenario either in periodic boundary or null boundary conditions is known as elementary CA (ECA). Next state of a ECA is determined by a mathematical function (Chaudhuri et al., 1997):

\[
S_{i+1} = f_i(S_{i-1}, S_i, S_{i+1}) \tag{1}
\]

where:

- \(f_i\) is next state function and termed as rule;
- \(S_i\) is current state value of \(i^{th}\) cell at time \(t\).
Examining Behavioral Intention Toward Mobile Services: An Empirical Investigation in Greece
[www.igi-global.com/article/examining-behavioral-intention-toward-mobile/53531?camid=4v1a](www.igi-global.com/article/examining-behavioral-intention-toward-mobile/53531?camid=4v1a)

Key Distributed Components for a Large-Scale Object Storage
[www.igi-global.com/chapter/key-distributed-components-large-scale/77428?camid=4v1a](www.igi-global.com/chapter/key-distributed-components-large-scale/77428?camid=4v1a)