

Processing Information in the Clouds [†]

Mark Burgin ¹, Eugene Eberbach ² and Rao Mikkilineni ^{3,*}

¹ Dept. of Math., University of California Los Angeles; Los Angeles, CA 90095, USA; mburgin@math.ucla.edu

² Dept. of Eng. and Sci., Rensselaer Polytechnic Institute; Hartford, CT 01609, USA; eeberbach@gmail.com

³ School of Business, Golden Gate University, San Francisco, CA 94105, USA

* Correspondence: rmikkilineni@ggu.edu

[†] Conference Morphological, Natural, Analog and Other Unconventional Forms of Computing for Cognition and Intelligence (MORCOM), Berkeley, CA, USA, 2–6 June 2019.

Published: 7 May 2020

Abstract: Cloud computing makes the necessary resources available to the appropriate computation to improve scaling, resiliency, and the efficiency of computations. This makes cloud computing a new paradigm for computation by upgrading its artificial intelligence (AI) to a higher order. To explore cloud computing using theoretical tools, we use cloud automata as a new model for computation. Higher-level AI requires infusing features of the human brain into AI systems such as incremental learning all the time. Consequently, we propose computational models that exhibit incremental learning without stopping (sentience). These features are inherent in reflexive Turing machines, inductive Turing machines, and limit Turing machines.

Keywords: sentience; artificial intelligence; knowledge structure; oracle; structural machines

1. Introduction

When Eugene Wigner discussed the role of mathematics in physical theories, he emphasized the unreasonable effectiveness of mathematics in the natural sciences. He wrote:

“The first point is that mathematical concepts turn up in entirely unexpected connections. Moreover, they often permit an unexpectedly close and accurate description of the phenomena in these connections. Secondly, just because of this circumstance, and because we do not understand the reasons of their usefulness, we cannot know whether a theory formulated in terms of mathematical concepts is uniquely appropriate.” [1]

In this paper, we examine a few new theoretical insights based on advanced mathematical theories, and their application to understanding the information processing structures in cloud computing. The term “cloud computing” refers to a variety of Internet-based computing services. The difference between cloud-based and traditional software is that the cloud provides elastic, on-demand computing resources for executing information processing structures. Most cloud computing services are accessed through a web browser or a dedicated mobile app or through a browser on a smartphone or tablet. Therefore, cloud services do not require users to have sophisticated computers that can run specialized software. Despite its popularity and rich applications, cloud computing is not a de facto choice. Its formal model(s), and therefore, some properties of cloud computing, cannot be properly studied to exploit its full richness. In this paper, we discuss the ability of the cloud automata model [2] to harness the power of distributed computing in clouds. These results are new and original, and by stressing the emergence of cloud computing as a new computational paradigm, it is possible to use cloud computing for problems that either cannot be solved at all or not to be solved effectively by traditional computers.

Cloud automata are constructed using the mathematical theory of Oracles [3], which include Oracles of Turing machines as its special case. We develop a hierarchical approach based on Oracles [4–7] with different ranks, which encompass Oracle AI [8] as a particular case. In the context of the named-set approach, we describe an implementation of a high-performance edge cloud using hierarchical name-oriented networking and Oracle AI-based orchestration. We demonstrate how cloud automata with a control overlay allows for microservice network provisioning, monitoring, and reconfiguration to address non-deterministic fluctuations affecting their behavior without interrupting the overall evolution of computation. A high-level AI requires infusing features that mimic human functioning into AI systems. One of the central features is that humans learn all the time and the learning is incremental. Consequently, for AI, we need to use computational models, which reflect incremental learning without stopping (sentience). This means that conventional models of computation, such as Turing machines, are insufficient for AI—which needs more advanced models, such as inductive Turing machines [9]. In our model, we also use the Oracle AI agents as hierarchical cognizing agents. These assist in modeling, monitoring and executing computational structures that process information, and manage their evolution even in the face of non-deterministic fluctuations in the availability or demand of resource.

2. Oracles in Clouds and Cloud Automata

Many scientists are convinced that computations going beyond Turing machines are possible, while contemporary and future computers can compute, and will compute, beyond the Turing limit using—for example, reactive programs, such as operating systems or client/server Internet computing [10,11]. Cloud computing also belongs to this category, and cloud automata are aimed at defining more powerful models of computation and computing networks. Current cloud computing model is mainly addressing how to make the right resources available to the proper computation to improve scaling, resiliency and efficiency of the computation. It is possible to define three stages of networking a general and cloud technology, in particular:

1. The *address-oriented networking* when the connection to systems and resources is based on their addresses in the network;
2. The *name-oriented networking* when the connection to systems and resources is based on their names;
3. The *hierarchical name-oriented networking* when the connection to systems and resources is organized through connection to the relevant host, which provides necessary systems and resources by their names.

Currently, all three types of cloud computing and networking coexist using the Internet as their communication space. However, today, the Internet architecture supports and is exclusively oriented only at address-oriented networking. It is possible to use two approaches to achieve the transition from address-oriented networking to name-oriented networking in clouds. One of them demands a change in the basic Internet architecture. The other approach does not demand such big transformations, because it utilizes network agents, such as described in *DIME* (Distributed Intelligent Managed Element) *Network Architecture - DNA* cloud [2,5,6] or *middleware CORBA* (Common Object Request Broker Architecture) or *Java RMI* (Java Remote Method Invocation) [10,11], to convert names to addresses. Sets and operations are named using an analytical description of these procedures [12].

Cloud automata are designed using the hierarchical approach in artificial intelligence, based on the mathematical theory of oracles [3,4]. This approach includes Oracle AI [8] as a special case providing new tools for the exploration of artificial intelligence in general and Oracle artificial intelligence in particular. In computer science, oracles appeared in the first half of the 20th century when Turing introduced into computing, an oracle machine or o-machine. It is a Turing a-machine that pauses its computation at state “o”, while to complete its calculation, it “awaits the decision” of “the oracle”—an unspecified entity “apart from saying that it cannot be a machine” [13]. Later, the concept of an oracle was utilized by different researchers. In his mathematical theory of Oracles [4],

Burgin describes an organizational hierarchy of Oracles, which consists of three levels of agent oracles:

1. Controlled agent oracles;
2. Regulated agent oracles;
3. Autonomous agent oracles.

A hierarchy of oracles is a useful structure for AI when machines (systems) are organized in layers so that a machine (system) in the layer n can be an oracle only for machines (systems) in the layer $n - 1$ and lower. Naturally, when such a hierarchy exists, machines (systems) in the lowest layer are not oracles for machines at higher levels in this hierarchy. This structure ascribes ranks to oracles in the following way. An oracle of the rank 1 is an oracle for one or several basic systems [3,4]. Assuming that functioning of machines is organized according to the hierarchical structure, an oracle of the rank n is an oracle for one or several oracles of the rank $n - 1$. Note that, in general, it is possible that a machine A plays the role of an oracle for a machine B in some situations, while in other situations the machine B plays the role of an oracle for a machine A. Hierarchic intelligence forms a hierarchy of oracles over the base information processing system. The oracles can be used as cognizing agents that configure, monitor and provide addressing, alerting, mediation and supervision functions for down-stream information processing systems based on global knowledge acquired from a peer or superior oracles. To build a high-level artificial intelligence, it would be useful to incorporate certain features of the human brain. One of the central features is that the brain learns all the time and learning is incremental. Consequently, for artificial intelligence, we need to use computational models, which reflect incremental learning without stopping (sentient systems). Oracles as cognizing agents coupled with knowledge structures [14] and structural machines [15] provide a framework to implement such sentient machines.

In this context, a *cloud automaton* is a three-component structure (M, C, O) , in which M is the total user (basic) automaton, C is the communication space and O is the Oracle O in a Cloud.

The *total user automaton* is a distributed system, which consists of *unit user automata*, each of which is utilized for computation or more general, for solving problems by the cloud user. The most convenient and flexible mathematical model of a user automaton is a grid automaton [9].

In a cloud automaton, the *Oracle* O can serve as a provider of information, services and/or virtual systems, e.g., computing and storage facilities, virtual software and hardware, data, and infrastructures, to the user automaton via the *communication space* [16].

When the unit user automata are either recursive, such as Turing machines or Minsky machines [12], or sub-recursive automata, such as finite automata, then a super-recursive automaton, such as an inductive Turing machine or inductive cellular automaton [17], can be the Oracle in a cloud automaton.

The structural machine framework relates knowledge structures (made up of named objects, their relationships and behaviors in the form of algorithms) and their processing to glean information using computing processes [15]. The knowledge structures and their evolution can be thought of as executable microservices and their evolution in cloud computing. An implementation of non-stop web service using a set of oracles managing a web server, application server and a database on multiple clouds is described in Reference [6]. High availability using auto-failover, auto-scaling, and live migration of application components from one cloud to another, which are separated geographically is demonstrated without disrupting the service transactions. The oracles manage the downstream application components based on policies that are executed by oracle agents.

3. Conclusions

The foundation for an information processing entity (IPE) comes from the stored program implementation of the Turing machine. However, IPE networking architecture augments the information processing power by facilitating communication among IPEs and allowing shared resources for computation within a computing machine or a network of these machines. The first phase of information networking used their host (or logical) address (usually an IP address in private

or public networks). A domain name server (DNS) provides the mapping between the location and the IPE. Inside each host, multiple IPEs are distinguished by various ports. During the past five+ decades, IP networking using IP addresses has been the dominant protocol providing the information networking services establishing links between the IPE nodes.

More recently, the focus has been shifting to IPEs as microservices that provide specific business (or other) functions, with each acting as an “active” node—hiding the details of implementation (which itself could be a subnetwork of IPEs). This provides a new class of hierarchical name-oriented microservice networking, where each node executes a specific behavior based on the input. A control overlay allows for microservice network provisioning, monitoring, and reconfiguration, as required to address fluctuations in their behavior.

In essence, such a hierarchical named network of IPEs allows the composition of micro-services to create a managed process workflow by enabling dynamic configuration and reconfiguration of the micro-service network. Examples of this are Kubernetes orchestrated microservice networks and DIME (distributed intelligent managed element) networks, as discussed in the literature. Both these examples use hierarchical named micro-service networks (and subnetworks) to provision, monitor, and control information processing structures and their resources, while addressing non-deterministic fluctuations in the demand or availability of needed computing resources. In this context, cloud automata provide theoretical tools for modeling, exploration, and improvement of information processing in clouds.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wigner, E. The Unreasonable Effectiveness of Mathematics in the Natural Sciences. *Commun. Pure Appl. Math.* **1960**, *13*, 1–14.
2. Burgin, M.; Eberbach, E.; Mikkilineni, R. Cloud Computing and Cloud Automata as A New Paradigm for Computation. *Comput. Rev. J.* **2019**, *4*, 113–134.
3. Burgin, M. Inaccessible Information and the Mathematical Theory of Oracles. In *Information Studies and the Quest for Transdisciplinarity: Unity through Diversity*; World Scientific: New York, NY, USA; London, UK, Singapore, 2017; pp. 59–114.
4. Burgin, M. On the power of oracles in the context of hierarchical intelligence. *J. Artif. Intell. Res. Adv.* **2016**, *3*, 6–17.
5. Mikkilineni, R.; Morana, G.; Burgin, M. Oracles in Software Networks: A New Scientific and Technological Approach to Designing Self-Managing Distributed Computing Processes. In Proceedings of the 2015 European Conference on Software Architecture Workshops, Dubrovnik/Cavtat, Croatia, 7–11 September 2015; ACM, pp. 11:1–11:8.
6. Burgin, M.; Mikkilineni, R. Cloud computing based on agent technology, super -recursive algorithms, and DNA. *Int. J. Grid Util. Comput.* **2018**, *9*, 193–204.
7. Mikkilineni, R.; Morana, G. Post-Turing Computing, Hierarchical Named Networks and a New Class of Edge Computing. In Proceedings of the 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Napoli, Italy, 12–14 June 2019, pp. 82–87.
8. Armstrong, S.; Sandberg, A.; Bostrom, N. Thinking Inside the Box: Controlling and Using an Oracle AI. *Minds Mach.* **2012**, *22*, 299–324.
9. Burgin, M. *Superecursive Algorithms*; Springer: New York, NY, USA, 2005.
10. Tanenbaum, A.; Bos, H. *Modern Operating Systems*, 4th ed.; Pearson: London, UK, 2015.
11. Tanenbaum, A.; Wetherall, D. *Computer Networks*, 5th ed.; Pearson: London, UK, 2011.
12. Burgin, M. *Superecursive Algorithms*; NovaScience: New York, NY, USA, 2011.
13. Turing, A.M. Systems of logic defined by ordinals. *Proc. Lond. Math. Soc. Ser. 2*, **1939**, *45*, 161–228.
14. Burgin, M. *Theory of Knowledge: Structures and Processes*; World Scientific Books: Singapore, 2016.
15. Burgin, M.; Adamatzky, A. Structural Machines as a Mathematical Model of Biological and Chemical Computers. *Theory Appl. Math. Comput. Sci.* **2017**, *7*, 1–30.

16. Burgin, M.; Liu, D.; Karplus, W. The Problem of Time Scales in Computer Visualization. In Proceedings of the International Conference on Computational Science, San Francisco, CA, USA, 28–30 May 2001; Lecture Notes in Computer Science; Volume 2074, part II, pp. 728–737.
17. Burgin, M. Inductive Cellular Automata. *Int. J. Data Struct. Algorithms* **2015**, *1*, 1–9.



© 2020 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).