# What Would Cloud Computing Learn from Quantum Mechanics

**Nilo Serpa**
*UNIP - Universidade Paulista, Brazil*

## INTRODUCTION

*We still do not know the "physical reasons" responsible for the Hilbert-space structure of quantum mechanics. Therefore, deriving quantum mechanics from a reduced set of primitives or axioms with a clear physical content remains an open problem for the foundations of physics. (Adán Cabello, 2005)*

The number of applications, systems and repositories that cohabit a corporate environment has grown at a rate never seen before. However, the fragmentation of information that usually follows this growth brings a lot of drawbacks, such as redundancies, inconsistencies, interoperability constraints and maintenance problems. In several occasions, systems are developed in response to specific requirements of certain sectors, whose managers take actions having absolute control on their areas without any concerns to an inclusive and democratic management that enables global organizational progress and compliance.

On one hand, the heterogeneity of computing environments is actually quite common in Brazil and other countries, especially in the public area, creating barriers to strategic planning and governance in general. The reasons for that surpass the usual scarcity of resources coming to the lack of specialized technical people, the lack of political will of the leaders and the lack of business strategic vision. On the other hand, technology speeds the improvement of business processes in a progression whose limits are difficult to predict. The degree of autonomy and intelligence of the systems to which it is possible to reach is an exciting matter and, at the same time, scary and inevitable.

This paper comments the convergence between SOA and cloud computing, discussing the future of these technologies with the birth of quantum computing. I intend to explain how cloud computing can work in a kind of beehive effect based on quantum entanglement of the states of information of quantum servers.

## BACKGROUND

### The Concept of Service

A glance over the literature that deals with the concept of service is sufficient to see no consensus on this notion. In non-academic literature, for instance, the term "service" appears in a very unusual way as in Sonera (2002), which associates services and XML Web Services within a context of marketing without any technical motivation. In most cases, it was associated with "service" an entry that merely references a software component established on a contract (Bieber & Carpenter, 2001). For another perspective, the academic literature is faintly alluding to the term "service" replacing it in most cases by "component" or "contract". From my point of view, services are cybernetic replicas of human practices, being evoked by well-established environmental motivations.

SOA is an architecture that integrates in a standard manner several service units, each of them sending their features as sets of tasks over the network. Only service interfaces are exposed to consumers as exported methods (Nakamura *et al.*, 2004). Therefore, when services are requested,

DOI: 10.4018/978-1-4666-5202-6.ch243

SOA seeks the best responses to those environmental motivations according to the internal logic of each service. In particular, this architecture is now strongly linked to the theme of "enterprise application integration" (EAI) in contexts where legacy applications already established are performed on different platforms.

The literature on SOA comprises several milestone contributions as the works of Nakamura et al. (2004), Erl (2005), Anderson & Ciruli (2006), Natis (2007), Sha (2007) and, markedly, Frenken *et al.* (2008) about device-level service deployment. On this latter subject, it is noteworthy that, in the process of architectural development, devices which access legacy applications are created and interact using a protocol defined by the system. In turn, the system returns the aggregated information from the various legacy applications, preferably without any additional code. The architectural development also takes care of the service interface, prescribing the information required to access the competences of that. It is worth remembering that the existence of interfaces and descriptions of accessibility is *sine qua non* for the implementation of SOA.

## Foundations of SOA Implementation

Before any effective action planning, it is necessary to establish what we call "reference model". This is an abstract framework to understand the ways in which occur the significant relationships between the constituent elements of a given environment, adopting obviously a systemic view. Such a model enables the implementation of specific architectures founded on consistent patterns that support this environment. The reference model consists of a minimal set of unifying concepts in a certain semantic topology, in addition to axioms and relationships with the private domain of the problem faced, being independent of specific technologies or applications. In short, the reference model aims to establish a top-level domain in which we recognize entities applicable to all the SOA; it defines the essence of the service-oriented architecture, offering a vocabulary and promoting a common understanding of SOA while it provides a robust basis enough to survive to the technological developments that will influence the SOA projects. The Enterprise Service Bus (ESB) is thought to be the main component of the infrastructure layer. It is the mediator between provider and service consumers, and its responsibility is to provide integration and interoperability between different systems. Embedded in this responsibility is also the mission of cleaning the databases by a service that tracks and recognizes all of them which will be linked; here is done a survey of the applied technologies, their limitations, fragmentation levels and other technical certifications. After this step, connectors are created in the databases feeding a new datawarehouse completely normalized, such that any updates made on the original basis are automatically computed and reflected in the standardized repository.

## The Cloud Computing Adventure

Following current ideas, we can say that cloud computing is a cybernetic implementation by which all IT resources (hardware, software, networking, storage, etc.) are provided as services on-demand to consumers via Internet, remaining managed to ensure fast delivery, high availability, security and quality. In short, cloud computing is a model of computation by which those IT resources are randomly dispersed in the network, being offered as services paid as they are consumed. Although this subject promotes a lot of controversy about information security, everything suggests that the process of agglomeration of servers in clouds is irreversible.

Cloud computing and SOA have contributed significantly with one another and should remain so for a long time. In the words of David Linthicum, "SOA can be used as a key technology-enabling approach to leverage cloud computing" (Linthicum, 2009). Thus, the use of SOA can be galvanized by the cloud structure, since it allows on-demand delivery beyond the limitations imposed by the