4th International Workshop on Computational Antifragility and Antifragile Engineering (ANTIFRAGILE 2017)

Towards Antifragile Software Architectures

Daniel Russo\textsuperscript{a,b,c,*}, Paolo Ciancarini\textsuperscript{a,b,c}

\textsuperscript{a}Department of Computer Science & Engineering - University of Bologna, Mura Anteo Zamboni 7, 40126 Bologna, Italy
\textsuperscript{b}Consiglio Nazionale delle Ricerche (CNR), Institute of Cognitive Sciences and Technologies
\textsuperscript{c}Consorzio Interuniversitario Nazionale per l’Informatica (CINI)

Abstract

Antifragility is a rising issue in Software Engineering. Due to pervasiveness of software in a growing number of mission critical applications, traditional resilience and recovery systems may not be sufficient. Software has taken over many functionalities which are of vital interest in today and future world. We rely a lot on software applications which may be faulty and cause immense damages. To cope with this scenario, claiming to develop better software is not enough, since unexpected events a.k.a. Black Swans, may disrupt and overcome our system. The purpose of this paper is to propose a new architectural design that responds to the need to build antifragile systems for contemporary complex scenarios. We suggest a system which enhances its strength through experience and errors. It is a self adaptive system architecture improving when facing errors. The most relevant aspect of this approach is that architectures are not only resilient, they extract the intrinsic value of faults. This paper suggests that a fine grained architecture is the key issue to build antifragile systems.

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.

Keywords: Complex Systems; Software Engineering; Antifragility; Architecture.

1. Introduction

The Jeep hacked remotely through the internet was not, per se a big event\textsuperscript{24}. However, the awareness that it was actually possible to compromise familiar mission critical devices, rose crucial questions. If someone can compromise remotely, just with an internet connection, a mission critical application, like a self driving car, potential impact on human life could be tremendous. Any car, and any mission critical application, could be a potential weapon which endangers people. This is leveraged by the fact that such kind of applications are rising, due to the wider use of software in nearly all human-related domains. However, also traditional critical industry, like the banking one, expressed concerns regarding the reliability of their systems\textsuperscript{23}.

The rise of ever tougher challenges in the domain of open and complex systems deserves solutions which are not built with a traditional incremental approach. There is the clear need for disruption. Technology in general has incredibly increased its capability in the last decades. Nevertheless, there are still big issues (both engineering and

* Corresponding author. Tel.: +39 051 2094861 ; fax: +39 051 20 94510.
E-mail address: daniel.russo@unibo.it
philosophical) that need to be tackled. Among them, the costs in money and time of designing, testing, delivering, operating and maintaining large systems is accelerating at an unsustainable rate. Moreover, these systems often do not perform as intended. Even if they perform as expected for some time, when failures occur, they are often difficult and expensive to fix. The architectural complexity increases also the probability of intermittent faults. There is the need for new engineering and philosophical designs to develop new systems to achieve those performances that traditional methods are unable to guarantee.

According to this view, mission critical organizations (e.g., NASA) are actively working to solve this big challenge. Instead of designing systems to meet known requirements that will always lead to fragile systems at some degree, systems should be designed wherever possible to be antifragile: designing cognitive cyberphysical systems that can learn from their experience, adapt to unforeseen events they face in their environment, and grow stronger in the face of adversity. Apparently, US federal research and engineering organizations are actively searching new solutions, since “addressing future challenges [...] is not simply to do what we know how to do now better: we need to do things we currently do not know how to do”.

The rest of this paper is outlined as follows. Section 2 explains the emergence of the new antifragile paradigm with respect to the differences with resilience. In section 3 major insights about design and implementation of such systems is given. Finally, a roadmap for an antifragile architecture is proposed in section 4 and in section 5 we briefly sum up our main findings.

2. A new emerging paradigm

The emergence of a new paradigm is the direct reaction to give the community a reference set for Antifragility. Etymologically, resilience (from the Latin resilire, to rebound) means springing back. Traditionally, in computer science and, in particular in the field of dependable computing systems, resilience has been intended as fault tolerance. The definition evolved to self-adaptive and automated systems, where the “resilience of a system as its ability to maintain service delivery that can justifiably be trusted in spite of changes in its internal and external contexts or in the interface between these (i.e., the avoidance of failures that are unacceptably frequent or severe, even when facing changes)”.

“Antifragility is beyond resilience or robustness. The resilient resists shocks and stays the same; the antifragile gets better.” In general, an antifragile system is able to evolve its identity in order to improve itself systematically in the context.

Main characteristics of antifragility are:

- Systems are not necessarily more “resilient” than change tolerant or elastic objects and systems. As it is the case for those entities, also antifragile systems are characterized by a yielding point (i.e., a point which beyond they would fail or become untrustworthy).
- The identity of systems itself may change over time. This means that the behaviors of antifragile software may outlay the system’s specifications.
- Systems may have some form of awareness of their current and past environment fitness. In particular, their risk model of adapt and evolve aspects of their system identity may be considered.

3. Design & Implementation

From an architectural perspective, by fragility is meant that any changes to a system, either during the integration of the system (e.g. software errors, changes in requirements), or during the maintenance of the system when additional functionality is to be implemented, will generally produce the result that some part of the application will not comply with the original architectural design.

Taleb extended such definition to any possible condition, where:
دریافت فوری متن کامل مقاله

امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات