Abstract

Nowadays, the development of reliable and efficient distributed systems is a must. In the last years, our group has been involved in the development of different solutions to real-world problems using off-the-shelf computer clusters managed by distributed control systems implemented using functional programming and, in particular, Erlang/OTP.

Despite the relevant advantages of the functional paradigm, we identified some lacks that avoid this programming model extend against other mainstream approaches. The goal or this project is to overcome those lacks improving final product quality —availability and performance— and decreasing its cost. In order to accomplish that, we explore different support tools in the whole software development life cycle, identify good practices materialized as a collection of design patterns, and build libraries and tools to solve general recurrent problems improving the Erlang/OTP framework.

Finally, as case study, the results are applied to the evolution of the VoDKA video streaming server as well as to other distributed functional applications deployed in real-world conditions.

Keywords: functional programming, distributed computing, cluster computing, design patterns, model checking, theorem provers, performance evaluation.

1 Introduction and Goals

1.1 Project Overview

The design and development of reliable, high available and efficient distributed systems is an important challenge. Convergence in communication networks requires at the present time and for the foreseeable future, more and more services to be added to these merging networks. What is more, these services are becoming more complex, both in themselves and in their interactions with each other and their end users.

*Email: gulias@udc.es
For performance reasons, and to be cost-effective, commodity off-the-shelf computer clusters have been suggested as a highly flexible and scalable architecture for some problems. As high speed networks and processor have become commodity hardware, affordable and reasonably efficient clusters are flourishing everywhere. However, the programming of such systems is still a difficult task. We believe that a declarative programming approach, such as the proposed by functional programming, plays an important role in the development. Moreover, the combined use of design patterns and distributed functional programming has been pointed out as a key factor to quickly produce correct distributed systems running on a cluster of computers with a high degree of adaptability, fault tolerance and scalability.

Despite the relevant advantages of the functional paradigm to develop systems with a high level of concurrency due to the high-level of abstraction and the (almost) absence of side-effects, some lacks are identified that avoids this programming model extend against other more conventional approaches (such as the proposed by the Java family).

One important aspect of the project is the emphasis on technological transfer to society. We think that distributed functional computing and all the related techniques can be useful for industry and one of our goals is to demonstrate it. In the last years, we have been working in the design, implementation and deployment of several distributed solutions for real environments by combining clusters built from commodity hardware and control systems implemented using the distributed functional paradigm. Among our experiences, we highlight the video-on-demand server VoDKA, a system for the distribution and streaming of media contents that has been almost completely developed using Erlang. Nowadays, VoDKA has left the laboratory environment to successfully compete with other existing products, being industrialized through the technology-based spinoff company LambdaStream (http://www.lambdastream.com). The main goal of this project is to overcome the mentioned lacks using as case study the evolution of the VoDKA streaming server as well as other distributed functional applications in real-world deployments, with thousands of concurrent users in 24x7 conditions.

1.2 Objectives

The concrete goal of the project is to design and develop an architecture for the analysis, design, implementation and performance evaluation of distributed functional applications, which includes:

- The identification and development of tools to support the software development life cycle: diagram tools, code generators from models, model simulators, and so on.

- The collection of good practices as a collection of design patterns.

- The development of frameworks and libraries that extend and improve Erlang/OTP to tackle common and recurrent problems: fault tolerance, high availability, on-line code update and change, process and service mobility, etc.

- The study of the use of formal techniques (mainly model checking and theorem proving) as part of the software development life cycle for distributed functional systems.
2 Success Level

2.1 Achievements

The public project results are shown in http://madsgroup.org/web/projects/farmhands. The achievements so far, according to the goals in section 1.2, are:

- A complete framework to develop distributed functional applications is being defined; a collection of design patterns and a library of recurrent functions has been identified and extracted from the analysis of VoDKA streaming server, including reflective generic servers, distributed scheduling using responsibility chains, guarded suspensions, data movement abstractions, model-view architecture based on observers, and so on.

- VoDKA server has been refactored focusing on the identified design patterns and recurrent libraries. The quality of the system has improved notably in terms of reliability.

- A new TCP/IP stack was designed and implemented in Erlang/OTP as part of Javier Paris Ph.D. thesis. It provides support for distributed applications to move their TCP connections to other servers in a way that is transparent to the connection peer. Two ways of replication are supported: moving the state from the old server or recovering it from the client. Both techniques are transparent to the client (peer uses a standard TCP/IP stack).

- An Erlang/OTP library implementing a distributed hash table (based on Chord algorithm) has been developed in order to create a distributed and decentralized global state dissemination and replication layer. This software layer is a powerful building block in decentralized P2P behavior design and can be applied in several recurrent problems. A simulator is in development as part of Carlos Abalde Ph.D. thesis.

- Different formal tools have been tested and compared along the development process (PVS and Coq), some help has been provided in the definition of others (McErlang) and even a new one has been developed (VoDKAV, as part of Juan J. Sánchez Ph.D. Thesis). Continuing our previous project (VRDADER, TIC2002-02859), those have been applied to VoDKA system for proving the correctness of relevant properties.

- A prototype of a model-based code generator tool for interactive applications has been designed and implemented. Based on a state diagram representing the interactive application, its output is all the source code and communications interfaces with integration web services required in order to deploy the application on different user terminals (web, FRESCO-based set-top boxes, MHP-based set-top boxes, etc.)

- A tool (JWMTool, Java Watermarking Tool) to empirically evaluate limits of digital media (particularly, video) watermarking processes was implemented in collaboration with Dr. Ernst Leiss (Univ. Houston) as part of a visit of Laura Castro to University of Houston.

- The aforementioned results from VoDKA evolution are also being applied to new complex distributed applications, deployed in real-world scenarios: the backend of an Electronic Service Guide for Mobile TV (Antares, http://lambdastream.com/products/antares), a playout system for DVB and 3GPP (BMX, http://lambdastream.com/products/bmx), massive advertisement distribution to public screens (Pulsar, http://lambdastream.com/products/pulsar), a clustered system for advertisement insertion in digital TV (for local cable company R Cable y Comunicaciones de Galicia), a distributed risk management information system for a large company (Armistice, http://www.alfa21.com/soluciones/armistice.html). All these developments were funded by project’s EPOs as a clear expression of interest on distributed functional programming and the results of this project.
3 Result Indicators

3.1 Formation Activities

During the project, one Ph.D. thesis have been finished:

- Juan J. Sánchez. From software architecture to formal verification of a distributed system. Directed by Thomas Arts (IT University of Gothenburg) and Víctor M. Gulías (University of A Coruña). University of A Coruña, 2006.

Juan J. Sánchez, former FPU recipient (2001-2004) and nowadays assistant lecturer at the University of A Coruña (since late 2004), has been working on applying model checking techniques to relevant patterns of VoDKA and, in particular, to the scheduler of the video-on-demand server. The software architecture of VoDKA is flexible and complex, and better tools are needed in order to increase the confidence of the system architects and improve the overall system quality. He studied how to use formal verification for that purpose. Therefore, using several tools from the area of formal methods, he proposed an innovative method for automatically extracting performance information about the system. As input to the method, the system source code and the system configuration (the description of the components and how they interact) are received. As output, feedback information about the system performance and architectural bottlenecks are provided. He extensively applied the method for analyzing the VoDKA system as a case study and showed how it can be reused with other tools and for other similar distributed systems.

There are three additional people in the project team working on its Ph.D. thesis:

- Javier Paris (late 2007), working on fault-tolerance and transparent migration of TCP services for distributed functional applications.
- Carlos Abalde (early 2008), working on P2P algorithms for massive information distribution.
- Laura Castro (late 2008), working on distributed systems for risk and digital rights management.

There are two additional people working on their Ph.D. which were part of the group in the past and now they are hired by one of the EPO of the project (LambdaStream) in its R&D department:

- Samuel Rivas, working on systems developed using distributed functional programming for digital video distribution.
- Carlos Varela, working on a unified ESG architecture implemented using the framework for distributed functional application development proposed by this project.

Moreover, many master thesis and DEA projects related to FARMHANDS were presented in the last two years. They represent preliminar work that we expect to consolidate in the following months.

3.2 International Publications

To summarize, the international publications (up to now) are:

<table>
<thead>
<tr>
<th>Type</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>papers in journals in JCR (not LNCS)</td>
<td>2</td>
</tr>
<tr>
<td>chapters in books (one on formal methods, three on multimedia technology)</td>
<td>4</td>
</tr>
<tr>
<td>papers in conferences with proceedings in Lecture Notes in Computer Science (all selected for extended and revised version)</td>
<td>4</td>
</tr>
<tr>
<td>paper in conference with publication in IEEE Computer Society Press</td>
<td>1</td>
</tr>
<tr>
<td>paper in conference sponsored by IFIP/IEEE Computer Society</td>
<td>1</td>
</tr>
<tr>
<td>papers in international conferences</td>
<td>5</td>
</tr>
<tr>
<td>papers in national conferences</td>
<td>2</td>
</tr>
</tbody>
</table>
A brief summary of the published results:

• The project proposes distributed functional programming, and in particular Erlang/OTP, as a tool to build reliable distributed systems for real-world scenarios and works to improve its weaknesses. Our experience with this framework includes both teaching and researching [15], being the VoDKA project one of our major achievements [10]. In [11], an overview of the VoDKA project is presented making emphasis on the role of declarative programming in the development of such a real-world application. In the project, we work towards a methodology and architecture which could be reused in further developments, significantly reducing the development life cycles of future versions [3].

• The work in [9], an extended version of [7], presents the certification of properties of an efficient functional program in the area of symbolic computation (the calculation of Gröbner basis of a set of multivariate polynomials), which shows several hints that can be applied to other case studies. To verify the properties, two approaches are explored: manual proofs that reason directly over the source code of the algorithms, applying techniques like equational reasoning, and theorem provers that are used as a tool to help us certify a model of the real system. In [8], the same approach is shown to prove properties about the block allocator of VoDKA’s distributed cache algorithm. The chapter in [5] summarizes some of the strategies to verify properties of a distributed functional program, using two different theorem provers (Coq and PVS) to show the method.

• A different approach to formal methods applied to the distributed functional program development life cycle is undertaken in [4]. In this case, McErlang, a verification tool (model checker) that takes as input an Erlang program and checks it against safety correctness properties also specified in Erlang, is shown as a useful tool to extract performance properties of VoDKA. The results are compared with those obtained using a different set of tools as part of Sanchez’s thesis.

• One of the identified recurrent components in large distributed systems is the presence of a decentralized peer-to-peer infrastructure for information distribution among nodes. In the case of VoDKA deployments, for example, this is necessary for media object pre-fetches in satellite networks with no broadband access. The work in [2] presents the extension of VoDKA architecture to include a generic P2P coordination algorithm built on top of a distributed hash table (Chord) and its integration inside the streaming server. This approach is applied in [1] to build a P2P video content distribution network developed as an extension of the VoDKA server, showing the experimental results that the design is quite appropriate for highly-correlated mass-scale content distribution, with interesting scalability and availability features. This feature is now the core of LambdaStream’s Pulsar product (http://lambdastream.com/products/pulsar).

• One of the more challenging difficulties in reliable distributed systems is to provide transparent service migration when it is not possible to deploy fault-tolerance components at client side. For example, in our case study of a video-on-demand system, user terminals (set-top boxes, mobile devices, regular video players, etc.) are shipped with somehow standard protocols that cannot be changed to provide alternative recovery measures in the event of a server crash. Some set-top boxes, for instance, relies on a RTSP session (a TCP control connection with the video-on-demand server) to control video playback. In [10] several TCP connection migration techniques are presented based on takeover/failover mechanisms defined in Erlang/OTP framework with the help of our own implementation of a TCP/IP server stack, that solves problems related to TCP state synchronization and IP address conservation after connection migration, among others. This alternative TCP/IP stack is general enough to be used as a component by several distributed applications with special high availability requirements. As an example, [17] shows a large distributed filesystem to store media assets.
• Different aspects of multimedia systems related to our case study (a video-on-demand server) are explored in several publications. In [18], different aspects related to multimedia stream management are covered. We analyze the different standards involved, the technical issues and the implications of actual multiplexing techniques. The main challenges of a multiplexer are meeting the bandwidth constraints and, at the same time, keeping reasonable buffering requirements for the receivers. In [19], we outline the technical challenges involved in the implementation of VCR capabilities for digital multimedia systems. Current video compression techniques based on motion compensated predictive coding and lossy image compression algorithms are a must to implement digital multimedia systems. However, they impose severe restrictions on how we can play the coded data. The inter-frame dependencies prevent a straightforward implementations of backwards playback and limit the points where the video can be accessed.

• In [20], the architecture explored by the project is applied to a new system, an electronic service guide (Antares, http://lambdastream.com/products/antares) for mobile TV (DVB-H). The ESG architecture must be flexible and scalable, and it should let integrate information coming from several heterogeneous sources, as well as to be able to broadcast information in multiple standards (CBMS, OAI, OMA...) simultaneously. In the work, we describe such architecture based on a decorators/plugins inspired in the VoDKA architecture.

• Based on the experience in the development of Erlang-based systems, a methodology and a reusable architecture is presented, reducing the development effort. In [3], this architecture is applied to the distributed risk management information system ARMISTICE, which was previously studied and formalized in [12]. Due to the importance of integrating existing applications, in [14], a reviewed and extended version of [13], we present a design that allows to reuse the logic of existing applications in a new execution environment following SOA principles. To achieve this purpose, the architecture developed follows the architectural patterns Model-View-Controller (MVC) and Layers.

3.3 Technological Transfer to Society

A key aspect of the project is to make possible that the explored research can be useful to industry. The main collaborations with the companies that supported our proposal have been:

• LambdaStream Servicios Interactivos. Its R&D department participates actively in the definition of the technological framework to develop distributed functional applications, in the evolution of VoDKA, and in the definition, design and implementation of new distributed applications (Pulsar, BMX and Antares) using this paradigm and applying the results of the project. As a sign of the compromise of exploiting the results of the project, it funded several private contracts in the last years that helped core FARMHANDS research: (a) Availability of Distributed Applications in Emergent Television Environments (2007), (b) Development of a Highly-Available Distributed ESG server for interactive digital television (2006-2007), (c) Adapting VoDKA video-on-demand server for DVB-C digital television (2005-2007).

MADS members apply some of project’s results in two PROFIT projects, funded by Spanish Ministry of Industry, in close collaboration with LambdaStream: (a) Hierba III: High Tech and Innovation for High Bandwidth Services in Rural Areas (FIT-350301-2006-14) (2006-2007) in collaboration with MAAT KNOWLEDGE, DS2, Multicanal de cable TVM, DESETECH, LambdaStream and UPM and Univ. of Castilla-La Mancha, and (b) HERMES, Convergence scenarios for mobile television (FIT-330220-2007-49) (2007-2008) with Red de Banda Ancha de Andalucia, LambdaStream, Sogetel, RTVE, TMira and Univ. of Castilla-La Mancha.

With the collaboration of MADS group, the company joins a STREP European project to applied property-based testing (ProTest project, see [7]), a European version of FARMHANDS to make cheaper software reliable and to make realable cheap software.
• **R Cable y Comunicaciones de Galicia S.A.** Besides starting commercial deployments of VoDKA, this cable company is interested in designing and implementing a clustered server for insertion of advertisements for digital television using the methodology proposed in this project.

• **Alfa21 Outsourcing S.L.** It funded the initial development of ARMISTICE risk management information system, nowadays deployed in a large multinational enterprise. The group has recently registered both server and client application.

An additional company, **Altea Consultores S.A.**, requested to apply FARMHANDS development methodology and experience in distributed services for interactive digital television with a private contract: *Platform for electronic administration services for interactive digital television (2007-2008).*

### 3.4 Collaboration with Other Groups

During the project, several interactions have been carried out with the external observers of the project: J.Aguilar (Univ. Merida, Venezuela), T.Arts (IT Univ. of Gothenburg, Sweden) and L.Fredlund (UPM, formerly at SICS). J.Paris visited Prof. Aguilar for three months; members of the project visited Prof. Arts and L. Fredlund several times; both Arts and Fredlund visited our group in Coruña several times; Professor Arts co-directed Sanchez’s thesis and there are some common publications.

Additional contacts with other groups: (a) Dr. V. Brabeman, Univ. Buenos Aires, dependable systems research group, visited for one month by J.Sanchez to present our methodological framework; (b) Dr. E. Leiss (collaborator of Aguilar), Univ. of Houston, multimedia and parallel systems group, visited for three months by L.Castro as a first approach to the application of watermarking techniques; (c) Dr. A. Ghodsi, Swedish Institute of Computer Science (SICS, thanks to Fredlund), due to the experience of this group in P2P systems and distributed algorithm verification; (d) J. Kangasharju, Univ. of Darmstadt (Germany), who focuses his research on content distribution and P2P technologies.

With the help of Arts and Fredlund, we got in touch with Prof. J.Derrick (Univ. of Sheffield, UK), who visited us in 2006. One of the key decisions of this meeting was to write an STREP proposal, ProTest, *property-based testing*, to develop software engineering approaches to improve reliability in pervasive and trusted network and service infrastructures (ICT-2007.1.2: Service and Software Architecture, Infrastructure and Engineering), inspired in FARMHANDS goals and also in a failed attempt in 2004 for a formally-based tool support for Erlang development STREP project. The consortium is composed by several companies (Ericsson, Sweden; Erlang Training and Consulting, UK; Quviq, Sweden; LambdaStream, Spain) and universities (Sheffield, UK; Kent, UK; UPM, Spain; Gothenburg, Sweden; Chalmers, Sweden). LambdaStream, headed by V.Gulia and with support of FARMHANDS team, will use Antares ESG as case study for property-based testing using innovative tools such as QuickCheck. The evaluation summary has been recently reported (a promising 14.5 out of 15).

### References


Make sure that this monobehaviour script should be attached to any of GameObjects. We already have Example Assets GameObject, so let's use this.

```csharp
using System.Collections.Generic;
using UnityEngine;

public class AddressablesController : MonoBehaviour {
    [SerializeField] private string _label;
    private Transform _parent;
    private List<_createdObjs> _createdObjs = new List<_createdObjs>

    The resulting system is highly scalable and can handle failing systems. All the data models and functions implemented in Secondo can be used in a scalable way without changing the implementation. Many aspects of the distribution are hidden from the user. An application developer is only required to implement these two functions; all aspects about the distribution and fault-tolerance are hidden. With Hadoop an open source implementation of this framework exists. MapReduce uses the Google file system (GFS) to provide a highly available and distributed data storage. New data can be added to GFS but it is impossible to modify existing data. Farmhand Feature Help 1. Select a Farm/Garden Select a Farm or Garden in your bag to use 2. Select Plants/Livestock Select... Do not make threads just to point out to your support ticket #. Do not ask for support with your Glyph account, and/or ArcheAge account. Please instead use Trion Worlds support for that. Do not distribute exploits, hacks, or RWT. Directly linking to and distributing said exploits, hacks, or cheats is not permitted. Proxy links are not allowed. Real money trades are not allowed in game nor on the sub-Reddit. Discussion about hacks and exploits are allowed. The detailed version of these rules is available by clicking here, otherwise, hover over the rules to get a simplified version of them. Cont