The Delta-4 Approach to Dependability in Open Distributed Computing Systems

David Powell (LAAS-CNRS, FR), Gottfried Bonn (IITB-Fraunhofer, DE), Douglas Seaton (Ferranti Computer Systems Ltd., UK), Paulo Verissimo (INESC, PT), François Waeselynck (BULL-MTS, FR)

in Proc's of 18th IEEE Int'l Symp. on Fault-Tolerant Computing (FTCS), pp. 246-251, June 1988.

(Commented analysis of paper and architecture, referring to sections when appropriate)

The Delta-4 project, which ran for 5 years (1986-91) with around 40 Mio Euro funding, is considered one of the most relevant European projects to date, having launched seminal research in fault tolerance in open distributed computing, using modular COTS systems, a pioneering achievement that left an indelible mark in the way F/T systems are designed today. This paper introduces the main features of DELTA-4: the innovative architecture, dependability model and implementation approaches.

Two main facts should be outlined. First, by introducing '*user-transparent fault-tolerance, on open systems*' (sec.Intro), DELTA-4 achieved a very significant quantum leap towards feasibility and vulgarization of F/T, i.e. "F/T systems where you can log on". In previous F/T systems, applications could not be agnostic from the system structure and served very specialized, closely-coupled systems, like e.g. TANDEM, or the seminal Jean-Claude Laprie Award winners SIFT or FTMP control-oriented architectures. Second, by introducing the fundamental '*fail-silence*' fault model, with its '*weak-fail-silence'* variation, Delta-4 allowed protocols to work under assumptions nearer reality and thus be extremely robust: nodes either worked correctly, or silently crashed, after a bounded number (≥1) of omission or timing faults. Previous models, such as the equally seminal Jean-Claude Laprie Award winner Fail-Stop Processor model --- with which fail-silence is often confused and miss-cited --- depended on additional abstractions of difficult implementation and enforcement, such as the need for a node that has failed, to inform all others of its failure, or to allow access to its storage after failure.

As an architectural paper, rather than a single mechanism/protocol one, its huge impact goes well beyond direct citations, but more fundamentally is indirect, by the change in the design culture of F/T systems that it has operated. In fact, the paper introduced many concepts and laid down fundamental advancements in the theory and practice of fault-tolerant computing, which are now commonly used in many current dependable systems. First, by inspiring further seminal research inside the project consortium itself, leading to an impressive impactful publication record by the several teams, during 5 years, and second by inspiring as well many young researchers and practitioners, who are now well established in our domain, having fostered many designs and developments along these lines.

In Delta-4, solutions were developed at many levels from architectural principles, hardware systems and communication protocols, up to software and administration layers.

For example, the paper presents the first fully-fledged architecture supporting '*F/T based on the distributed replication of software components on heterogeneous, COTS hosts'*. The only similar effort known contemporarily is the equally successful but more focused approach, ISIS (Birman,et.al), offering less complete options, since it was essentially a software toolbox running at application level, not supporting real-time, or arbitrary faults, whereas DELTA-4 supported '*a wider set of fault assumptions, with extended timing properties'* (sec.Intro).

With its innovative replication models, such as the '*semi-active replication'*, DELTA-4 gave the necessary leap forward to ensuring active replication in systems that are not necessarily lock-step and where replicas can lag, but remain synchronised in practice (sec.1.1.4.1). The technique was reused often and later even re-hashed towards the other side of the spectrum, as semi-passive replication (Défago,Schiper,et.al).

The DELTA-4 architecture was as complete as reaching out to real-time, extra-performance, and security, and went as far as devising a F/T object model, *'DELTASE'* (sec.3), which was featured as "*the recommended approach to achieve F/T*" in the contemporary ANSA ODP model manual, precursor of the well-known OMG CORBA model.

DELTA-4 invented the 'propagate-before-validate' replication management technique (sec.1.1.4.b), whereby a subset of replicas were allowed to progress faster in absence of errors, defining correctness conditions for the validation points, error confinement and recovery. This inspired later speculative execution replication techniques in general, and in particular executions with a minimum quorum of replicas until the first vote mismatch, inspiring further works and re-hashed recently in dual-mode BFT-SMR protocols like CheapBFT (Kapitza,Cachin,et.al).

DELTA-4 invented sophisticated protocols for live re-instantiation or migration of replicas, or '*cloning'* (sec.1.2.3), actually pioneering automatic reactive recovery of failed replicas, a problem which would be often "re-solved" in the coming years.

Introducing rudiments of '*architectural hybridization*' and '*trusted computing'* which saw the light many years later, in DELTA-4, the fail-silence assumption of the communication layer was enforced (i.e. substantiated) by construction of '*trusted network attachment controllers*' (sec.2). This made it easy to support any host faulty behavior, from crash to arbitrary, and any kind of COTS, obviating the need for 3f+1 hosts to achieve arbitrary F/T: only 2f+1 were needed. This DELTA-4 invention is nothing else than the '*separation of agreement from execution*', and was later nicely cast into asynchronous BFT systems (Yin,Alvisi,et.al).

In DELTA-4, the communication layer correctness relied on an atomic multicast protocol (*'AMp',* sec2.1.2) distributing messages to replica groups. Portability was ensured by AMp relying on a set of '*abstract network'* properties, which were then

implemented for each actual network DELTA-4 was ported to. AMp introduced two crucial innovations w.r.t. predecessors, which were either asynchronous but assumed a loss-free network (ISIS), or were synchronous and timed (clocked), and assumed perfect timeliness of the network (SIFT, AAS). AMp eliminated these two causes of brittleness leading to unexpected coverage failures, gaining a robustness which was verified by formal verification and fault injection.

First, AMp made network faults visible, introducing the '*bounded omission degree*' technique, whereby a bounded number of omission or timing faults were allowed before fail-silence (the weak-fail-silence hypothesis). Second, AMp had a time-free, round-based structure, and through a technique which would be later re-hashed and nicely called *'immersion'* (LeLann), it inherited its real-time behavior as a function of the timeliness parameters of the raw abstract network it was implemented over.