Proving causality in distributed systems

**Keywords:** distributed system, trace of events, causality, dependency, proof.

**Required skills:** The study requires a taste for logic, semantic of concurrent systems, some programming skills and of course a great motivation.

**Bibliography:** It is advisable to read the following papers:

**Context** The LISE project aims at bringing the means to demonstrate in a court trial the responsibility of a software. Our domain of application is distributed system made of several communicating components (e.g., cellular phones and servers). More and more critical operations are performed using a cellular (signature of a contract, payment, bank transfer, ...). Such operations involve many components: the end users, the provider and many intermediate servers. During a critical operation the activity of each components is recorded into a log that shall be use to determine the responsibility of each component in a case of failure (for instance your bank account has been debited but the provider pretend having received neither your command nor your payment).

Lawyers are not experts in information system and networking. Moreover, they only accept evidence that cannot be contested by any party. The goal of the LISE project is to provide proof of responsibility in a form that cannot be disputed, and hence, will be recognized as trustable in a court trial. The law has recourse to the scientific policy (for ADN comparison,...) to resolve murders ; we seek for the same level of confidence in the domain of information system.

**Subject** Prior to any analysis (debugging, testing, verification, ...) of a distributed system is the ability to establish the order (called *causality*) between the occurrence of two events \(e, e'\) emitted by two different components. In an ideal world the *causality* between two events refers to the global time: \(e \rightarrow e'\) if \(\text{date}(e) < \text{date}(e')\). It is well known however that in distributed systems we
can’t assume the existence of a global clock – a clock that would be common to all components of the system. Each component has its own clock with its own frequency. Actually, establishing a global clock requires complex protocol and many exchange of messages between the components. This solution doesn’t suit for systems with limited capacities (in bandwidth, computation, energy,...). This is the case in the Lise project where we consider small systems with non-permanent connection (like cellular phones). Each component records critical actions and communication (sending/receiving of a critical message) in a log file.

In order to establish the causality between two events we can only rely on local ordering of the events in the log file of each component and recorded communications between components. Then we are faced to a set of traces of events from which we must discover causality relation. For any given events $e, e'$ we look for a way to prove one of the statements:

- $e \rightarrow e'$ the order between the events can be established and we can prove that $e$ happened before $e'$
- $e' \rightarrow e$ we can prove that $e$ happened before $e'$
- $e \sim e'$ the two events belongs to the same frame and it can be proved that the events cannot be ordered. In that case the size of the frame can be estimated by providing the last event before the frame $l$ and the fist event after the frame $f$ such that $l \rightarrow \{e, e'\} \rightarrow f$

As an appetizer we provide two examples of deduction rules that can be used in proof.

\[
\begin{align*}
\frac{e_1 \rightarrow e_2 \quad e_2 \rightarrow e_3}{e_1 \rightarrow e_3} \quad & \quad \frac{C!m \quad C'?m}{!m \rightarrow ?m} \quad & \quad \frac{\text{com} \quad e; e' \in C.log}{e \rightarrow e'} \\
& \quad \text{trans} & \quad \text{seq}
\end{align*}
\]

Rule ($\rightarrow_{\text{trans}}$) expresses the transitivity of causality. Rule ($\text{com}$) establishes a causality between the events in a communication: the reception of the message (denoted by the event $?m$) and its emission $!m$. If a reception of the message $m$ has been logged by component $C'$ and a emission of the message $m$ has been logged by component $C$, then the emission should have occurred before the reception. Rule ($\text{seq}$) derives the order relation $e \rightarrow e'$ from the fact that the event are recorded in this order in the log of a component.

**Work** The student will have to study previous works on causality in distributed systems, define a deduction system for causality, and implement an analysis of distributed system traces that establishes some causalities together with a proof of its results.

The tool could be improved to take as inputs a set of component logs and a property to prove or disprove.

An interesting experiment would be to model the network as a component (with a log) that can introduced delay, lose messages, or replay them. The analysis can
then be extended to discover (and prove) irregularities in the causality, missing events in a log, detecting cheating log or system failure.

In this work we assume that the log of each component is trustable. It could be interesting to avoid this assumption: it then requires to consider the specification of component in a simple language that mainly focuses on communication in order to check that a log is conform to the specification.

Contacts: michael.perin@imag.fr, marie-laure.potet@imag.fr

Laboratoire Verimag (CNRS, INPG, Université de Grenoble)
CTL building
2 av. de vignates
38610 Gières, France

http://www-verimag.imag.fr/