

# PhD Thesis Topic: Optimising Coordination in Concurrent and Geo-Distributed Systems

Many modern applications are concurrent or even distributed. Dividing the application into separate processes enables to parallelize processing, to place data and computation close to their point of use, or to replicate them. This helps improve performance metrics such as response time, bandwidth, availability, fault tolerance, and/or energy consumption.

However, parallel computations have highly non-deterministic behavior. This makes it difficult to program the application correctly. Similarly, debugging and verification are notoriously hard, because of the state space explosion.

## Background

The correctness of an application can be formalised as a set of *safety invariants*, i.e., assertions about its state that must always be true. In previous work [1, 2] we have identified generic conditions for maintaining a given invariant. We have also developed automated analysis tools that identify precisely what is the minimal coordination (i.e., concurrency control) that a given program must contain to satisfy these conditions. If the application has the required coordination, it is guaranteed that every execution remains safe, despite concurrency.

However, there remains a big gap between this specification and efficient implementation. There are many open design decisions: at what granularity to coordinate (e.g., fine- vs. coarse-grain locking), what concurrency control primitives to use (e.g., locks vs. transactions vs. consensus), what variant thereof (e.g., mutual exclusion vs. shared-exclusive), and where to place the coordination master (e.g., centralized vs. decentralized vs. migrating). There are many trade-offs; the best choice depends on the specific application, on the performance metric to be optimised, and on the workload.

In preliminary work [3], we have started to develop a theory of coordination performance. Focusing on locks, we formalize the possible decisions as a Coordination Lattice, which enables to systematically explore the decision space, based on a model of the system topology and of the workload. By simulation (based

on simple applications and synthetic data), we have obtained some interesting and non-intuitive results.

## Research plan

The objective of this PhD is develop a practical approach to optimizing concurrency control in actual, complex applications. The PhD student will aim to develop a tool that assists the application developer with implementing coordination. It shall analyze the application and its requirements, identify what coordination is necessary for correctness, analyze the performance space based on models and on real traces, and identify the trade-offs relevant to the developer's performance objectives, under the constraint of maintaining correctness.

The research will build upon the Coordination Lattice approach, applying it to actual distributed applications with non-trivial invariants and interesting workloads. It will extend the current theory to a more comprehensive set of coordination primitives (e.g., leases or consensus), consider the additional dimensions of liveness and fault tolerance. The modelling of environmental constraints, such as workload or network latency, will use either real traces, or a statistical model, or even machine learning. The aim is to be able to predict robust real-world performance metrics.

## References

- [1] *'Cause I'm Strong Enough: Reasoning about Consistency Choices in Distributed Systems*. A. Gotsman, H. Yang, C. Ferreira, M. Najafzadeh, M. Shapiro. **Symp. on Principles of Programming Languages**, Jan. 2016. St. Petersburg, Florida, USA
- [2] *Proving the safety of highly-available distributed objects*. S. Nair, G. Petri, M. Shapiro. **29th European Symposium on Programming (ESOP)**; April 2020; Dublin, Ireland
- [3] *Designing safe and highly available distributed applications*. S. Nair. PhD Thesis, defended 1 July 2021. Sorbonne-Université, Paris, France.