PATCHYBFT: LLM-based Diversification of Byzantine Fault-Tolerant Systems

***Abstract*—Byzantine Fault Tolerant (BFT) and Crash Fault Tolerant (CFT) protocols are a key component in many distributed systems. In general, these protocols enable us to tolerate a fraction of faulty replicas and still provide service. However, their fault tolerance collapses as soon as replicas share a common flaw that can simultaneously affect more replicas than the tolerable threshold. Therefore, replicas need to be fault-independent, which can be achieved through diversification. While this has been considered in various ways, the same protocol implementation is commonly shared by all replicas. This is not surprising, given that the logic of these protocols is usually highly complex and concurrent, making the provision of multiple diverse implementations difficult and highly laborious. However, this also poses a major risk, as a shared protocol implementation, due to its complexity, is a prime candidate for common bugs.**

With PATCHYBFT, we demonstrate how Large Language Models (LLMs) can be utilised for the automated and scalable generation of fault-independent implementations of BFT protocols, thereby significantly reducing diversification costs. Given a reference implementation, PATCHYBFT uses LLMs to rewrite the protocol one function at a time, which can be patched into the reference implementation. We demonstrate the feasibility of generating BFT implementations using LLMs: with PATCHYBFT, we achieve up to 81% of code reimplemented, which is on par with manually implemented code with no performance overhead.

Byzantine Fault Tolerant (BFT) und Crash Fault Tolerant (CFT) Protokolle sind eine Schlüsselkomponente in vielen verteilten Systemen. Sie ermöglichen uns diese Protokolle, einen Teil fehlerhafter Replikate zu tolerieren und dennoch den Dienst bereitzustellen. Ihre Fehlertoleranz bricht jedoch zusammen, sobald Replikate einen gemeinsamen Fehler aufweisen, der gleichzeitig mehr Replikate als die tolerierbare Schwelle betreffen kann. Daher müssen Replikate fehlerunabhängig sein, was durch Diversifizierung erreicht werden kann. Obwohl dies auf verschiedene Weise in Betracht gezogen wurde, wird in der Regel von allen Replikaten dieselbe Protokollimplementierung verwendet. Dies ist nicht überraschend, da die Logik dieser Protokolle in der Regel sehr komplex und parallel ist, was die Bereitstellung mehrerer unterschiedlicher Implementierungen schwierig und sehr aufwendig macht. Dies birgt jedoch auch ein großes Risiko, da eine gemeinsame Protokollimplementierung aufgrund ihrer Komplexität besonders anfällig für häufige Fehler ist. Mit PATCHYBFT zeigen wir, wie Large Language Models (LLMs) für die automatisierte und skalierbare Generierung fehlertoleranter Implementierungen von BFT-Protokollen genutzt werden können, wodurch die Diversifizierungskosten erheblich reduziert werden. Ausgehend von einer Referenzimplementierung verwendet PATCHYBFT LLMs, um das Protokoll Funktion für Funktion neu zu schreiben, die dann in die Referenzimplementierung eingepatcht werden können. Wir demonstrieren die Machbarkeit der Generierung von BFT-Implementierungen mit LLMs: Mit PATCHYBFT erreichen wir eine Code-Reimplementierungsrate von bis zu 81%, was mit manuell implementiertem Code ohne Leistungsaufwand vergleichbar ist.