ADAPTIVE CONSISTENCY FOR LARGE-SCALE DATA REPLICATION

Résumé du projet de recherche (Langue 1)

In large scale systems, such as clouds, content distribution network, entreprise networks or peer-to-peer systems, data is distributed and replicated over the network. By placing data near its users, and by providing redundant copies of data, this improves response time, and availability for read-mostly workloads. For instance, a CDN places multiple copies of a popular web page across the Internet, so that distributed users can access the page quickly despite WAN latencies. As another example, a replicated database makes multiple copies of each data item to ensure that it survives despite crashes or network disconnection. When shared data is mutable, however, such large-scale data distribution and replication raise the issue of consistency. In order to maintain application invariants, updating an object must be coordinated across its replicas and with related objects (transactional updates). The general solution is to force a serial order of operations (using a consensus protocol) that all replicas will follow [1]. Although obviously correct, this approach does not scale. Consensus constitutes a serialisation bottleneck, and as the number of replicas increases, the load on all processors increases. However, many applications do not require strong consistency. In fact, there is a whole spectrum of trade-offs between performance, correctness, and consistency. Some techniques that are known to work are: - Partitioning the data. Operations that affect independent data items need not be mutually ordered. - Applying commutativity. Operations that commute do not need to be mutually ordered. Commutative replicated data types (CRDTs) require no foreground synchronisation at all. - Optimistic approaches move consensus moved into the background, off the critical path. Replicas may progress in parallel; they may diverge temporarily, as long as they converge eventually. This assumes the application can tolerate observing inconsistent, tentative state. - Reduced isolation levels. These guarantee a weaker class of invariants than serialisability, but support more parallelism. - Partial replication, i.e., a replica executes only the operations of interest to the local user. This reduces the load, but requires more complex protocols.

Résumé du projet de recherche (Langue 2)

Consistency requirements are application specific. We propose to adapt the consistency system to specific application requirements. Ideally, our approach will be to reify (in a "consistency SLA") the invariants that define the application's correctness, and let the system use the best-performing consistency protocol that guarantees the specified invariants. This approach allows developers to focus on application functionality and semantics, leaving global arbitration and distributed system issues to the system. Telex, our existing proof-of-concept platform, shows that this is possible. Telex is very general but quite complex. The goal of this thesis is to explore the practical engineering space based on the same concepts. The idea is to identify "sweet spots" in the design spectrum. Social networking and collaboration constitutes one promising class of applications. Cloud architectures, composed of subsets of highly-coupled machines interconnected by slow, unreliable WANs is a promising topology.

Informations complémentaires (Langue 1)

Stage "PhD internship" obligatoire