

How do we handle then [scaling](#) problems described in [Scalability Problems](#) ?

- [Use Resources Proportional To Fixed Limit](#)
- [Merge Aggregation](#)
- [Delete Aggregation](#)
- [Batch Aggregation](#)
- [Change Aggregation](#)
- [Integration Aggregation](#)
- [Load Shedding](#)
- [Tie Work to Resource Availability](#)
- [Reference Counting](#)
- [Data Grid](#)
- [End to End Flow Control](#)
- [Remove Round Trips](#)
- [Design Idempotent Protocols](#)
- [Design Stateless Servers](#)
- [Design Crash Only Software](#)
- [Congestion Control](#)
- [Use Static Mappings](#)
- [Load Balancing](#)
- [Prioritize Work](#)
- [Remove Retries](#)
- [Caching](#)
- [Share Nothing Architecture](#)
- [Use Data Parallel Algorithms](#)
- [Use Multi Processor Machines](#)
- [Remove Single Points of Serialization](#)
- [Hash Based Node Selection](#)
- [Make it Big Enough](#)
- [Sampling](#)
- [Grid Space Architectures](#) ↗

References

- [Handling Infinite Work Loads](#)
- [Architecture Discussion](#)
-