Concurrency Design Architecture

Concurrency Model Patterns

Here are some design patterns that can help make multi-threading deterministic:

- Producer-Consumer Pattern - Foundation for many designs
- Actor Pattern - Erlang/Elixir style message passing
- Reactor Pattern/Proactor Pattern - Event-driven architectures
- **Pipeline Pattern** - For processing workflows
- **Worker Pool Pattern** - Load distribution patterns
- **Event Sourcing Pattern**/ Command Query Responsibility Segregation Pattern - For complex state management
- Write-Heavy Design Patterns - For write heavy applications

Data Sharing Style

Shared State - two way behavior, requires locks and atomic operations.
Message Passing - one way directional behavior, avoids locks by isolating state.
Immutable Data - safe to share freely.