

11 - Arquitetura de Pipeline

2025-12-20 19:51 pm

- Pipeline ou *pipe and filters*
- Princípio usado em bash ou zsh
- Parece programação funcional - análogo ao modelo MapReduce

Topologia

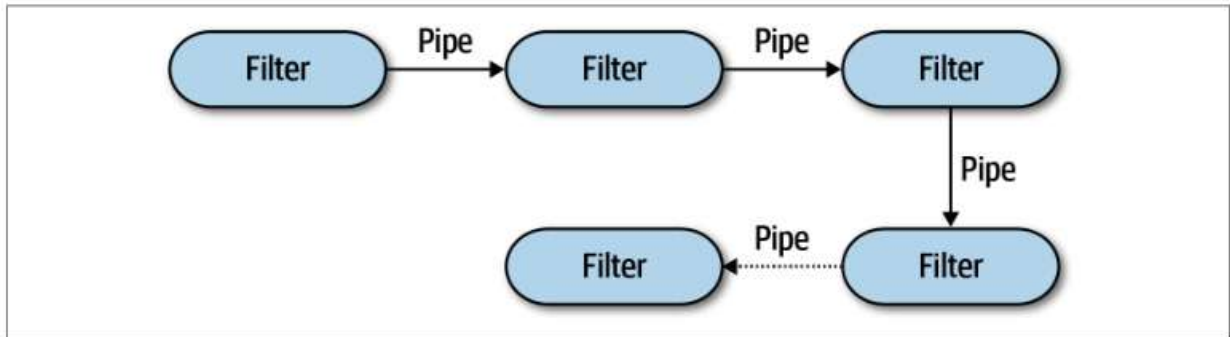


Figure 11-1. Basic topology for pipeline architecture

- **Pipe**
 - Canal de comunicação entre os filtros
 - Unidirecional e de ponta a ponta.
 - Preferível que o input seja pequeno
- **Filters**
 - São autocontidos e independentes dos outros filtros
 - Stateless
 - Um filtro deve executar uma única tarefa
 - Quatro tipos de filtros:
 - **Producer:**
 - Source, ponto de partida
 - **Transformer:**
 - Transforma um input e o passa ao destino. Um **map**
 - **Tester:**
 - Aceita input, testa algum(ns) critérios e gera ou não um output. Um **reduce**
 - **Consumer:**
 - Ponto de destino. Pode persistir resultado em um database e/ou exibir resultado.
 - Simplicidade dos componentes favorece reuso.

Exemplo

- ETL costumam usar esta arquitetura

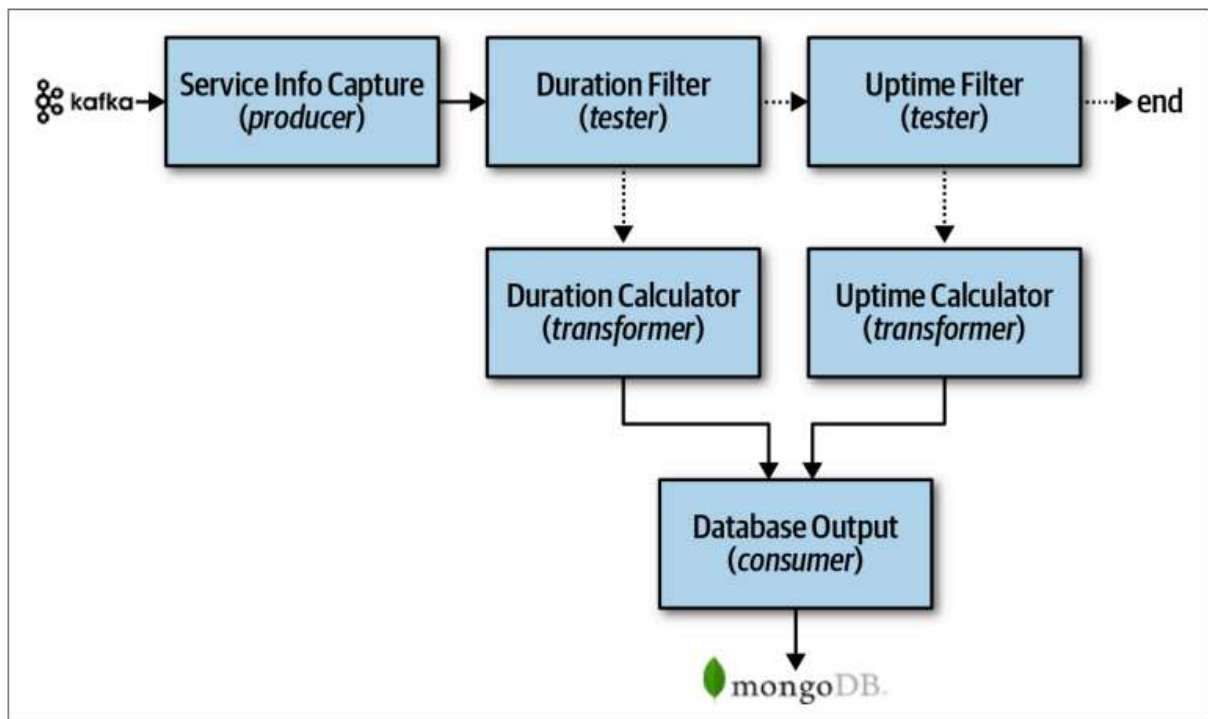


Figure 11-2. Pipeline architecture example

- - Neste caso, um novo tester poderia ser facilmente inserido após o uptime filter.

Classificação

Architecture characteristic	Star rating
Partitioning type	Technical
Number of quanta	1
Deployability	★ ★
Elasticity	★
Evolutionary	★ ★ ★
Fault tolerance	★
Modularity	★ ★ ★
Overall cost	★ ★ ★ ★ ★
Performance	★ ★
Reliability	★ ★ ★
Scalability	★
Simplicity	★ ★ ★ ★ ★
Testability	★ ★ ★

- Estilo tecnicamente particionado.
- Costuma ser um deploy monolítico, um só quantum de arquitetura.
- Pontos fortes são simplicidade, custo e modularidade (mas ainda é um monolito)
- Testes requerem teste de todo o monolito.
- Pouco escalável, desacoplar desempenho das peças é muito complexo.
- Erro de um pedaço causa erro em tudo.