Overload Control for Scaling WeChat Microservices

WeChat

The new way to connect

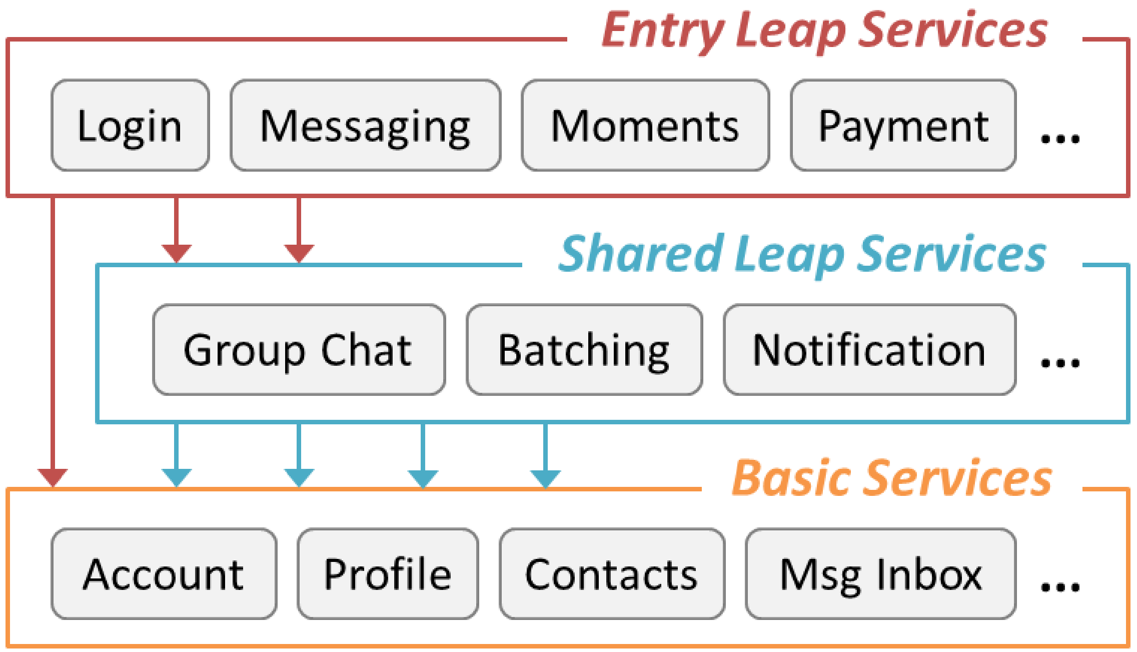
- Chat
- Moments
- Contacts
- Search
- Pay

1 Billion

monthly active users

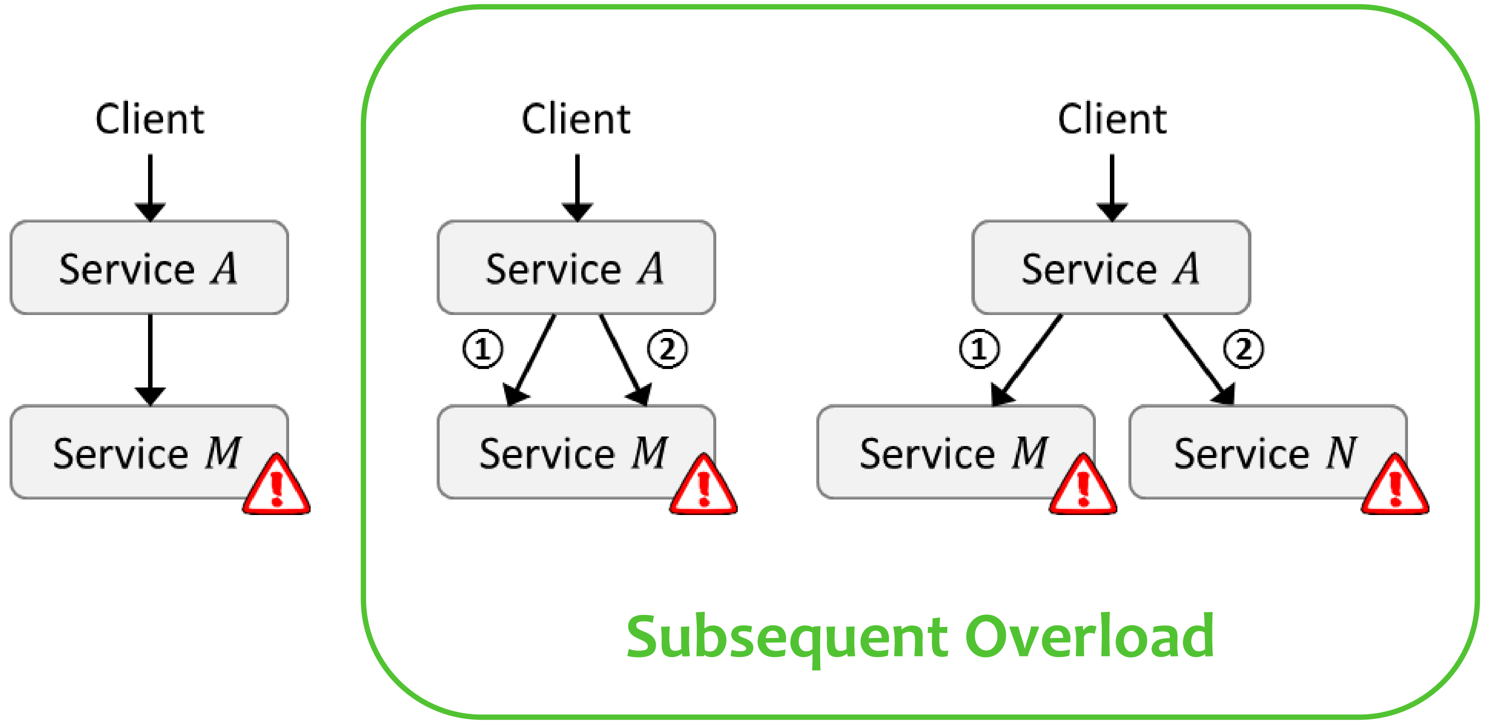
WeChat's Microservice Architecture

- **Service DAG**
  - Vertex: a distinct service; Edge: call path
  - **Basic service**: out-degree = 0
  - **Leap service**: out-degree ≠ 0
    - **Entry service**: in-degree = 0



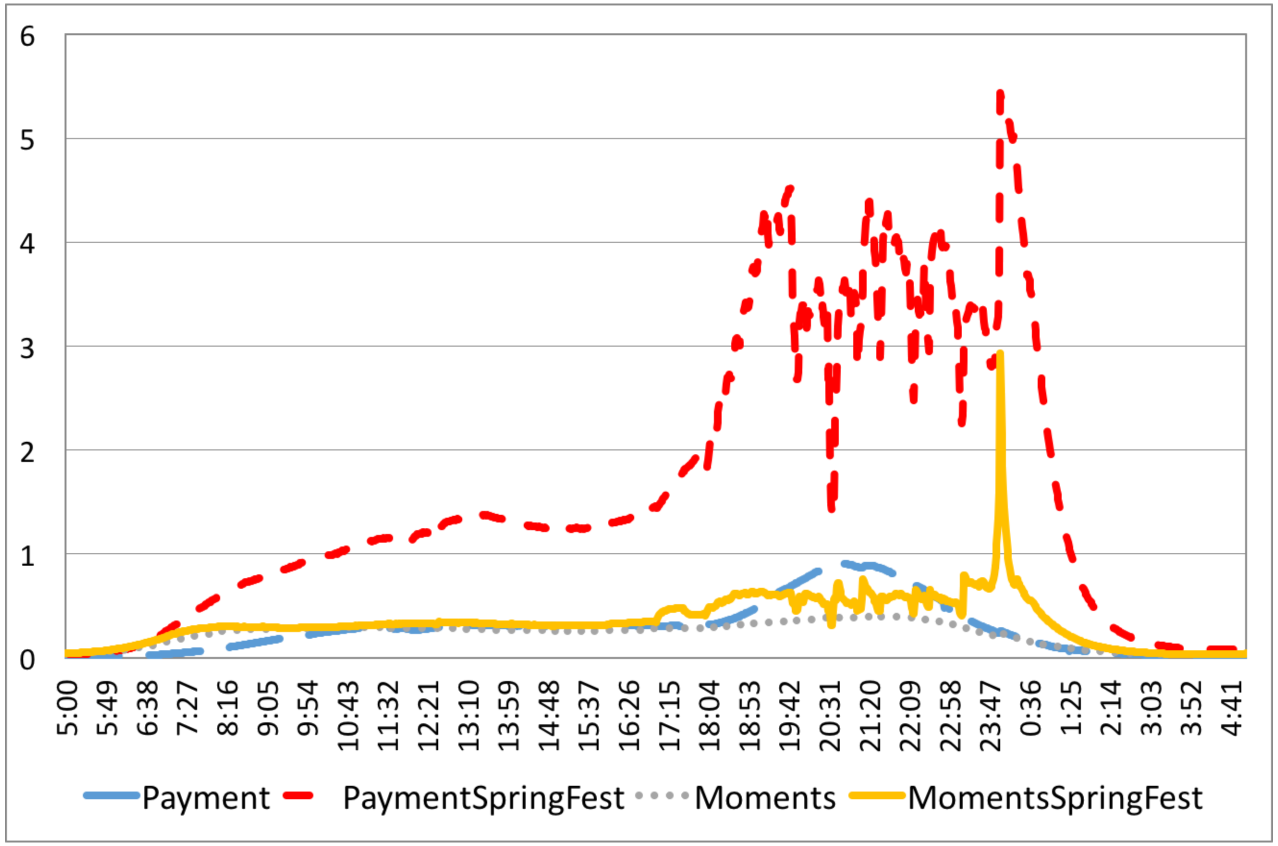
Dealing with Overload

- It's usually hard to estimate the dynamics of workload during the development of microservices.



How about random load shedding?

Dynamic Workload



Relative Statistics of WeChat Service Requests

DAGOR

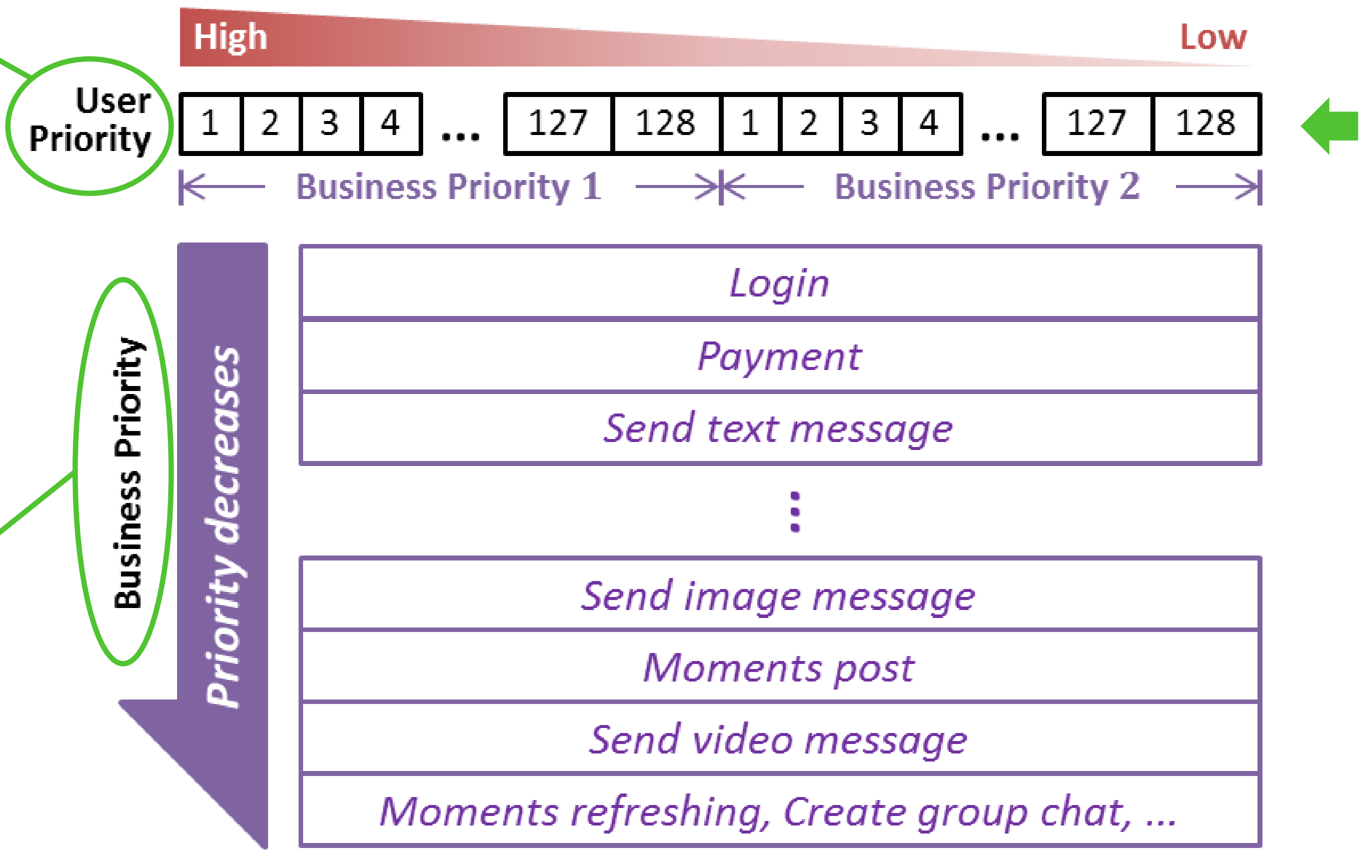
- **Overload detection**
- **Service admission control**

- **Requirements**
  - Service agnostic
    - Benefit the ever evolving microservice system
    - Decouple overload control from the business logic of services
  - Independent but collaborative
    - Decentralized overload control
    - Service-oriented collaboration among nodes
  - Efficient and fair
    - Sustain best-effort success rate of service when load shedding becomes inevitable
    - Bias-free overload control

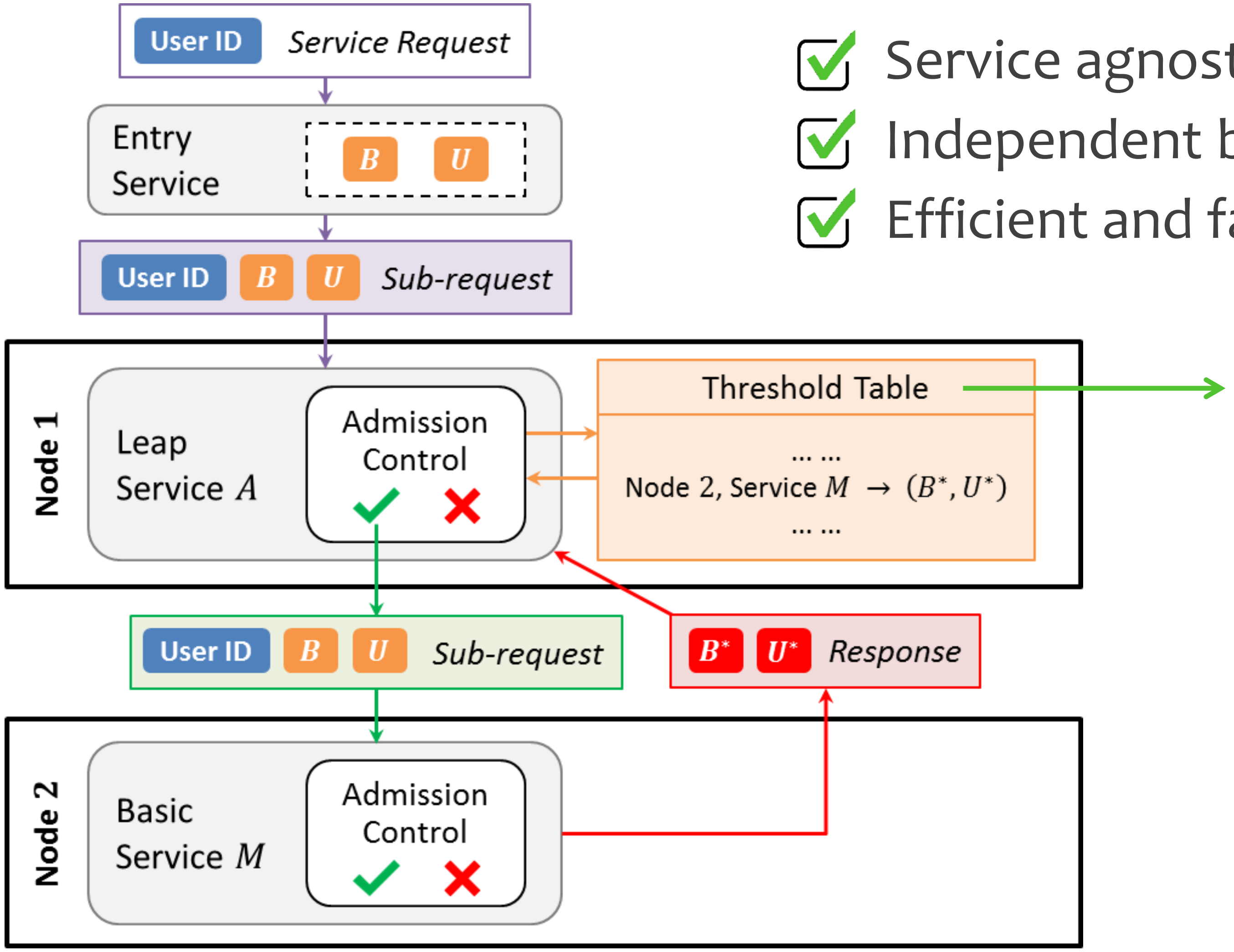
Overload Detection

- **Load indicator of a node: Queuing time**
  – Rationale: to manage queue length for SLA
- **Why not response time?**
- **Why not CPU utilization?**

Service Admission Control



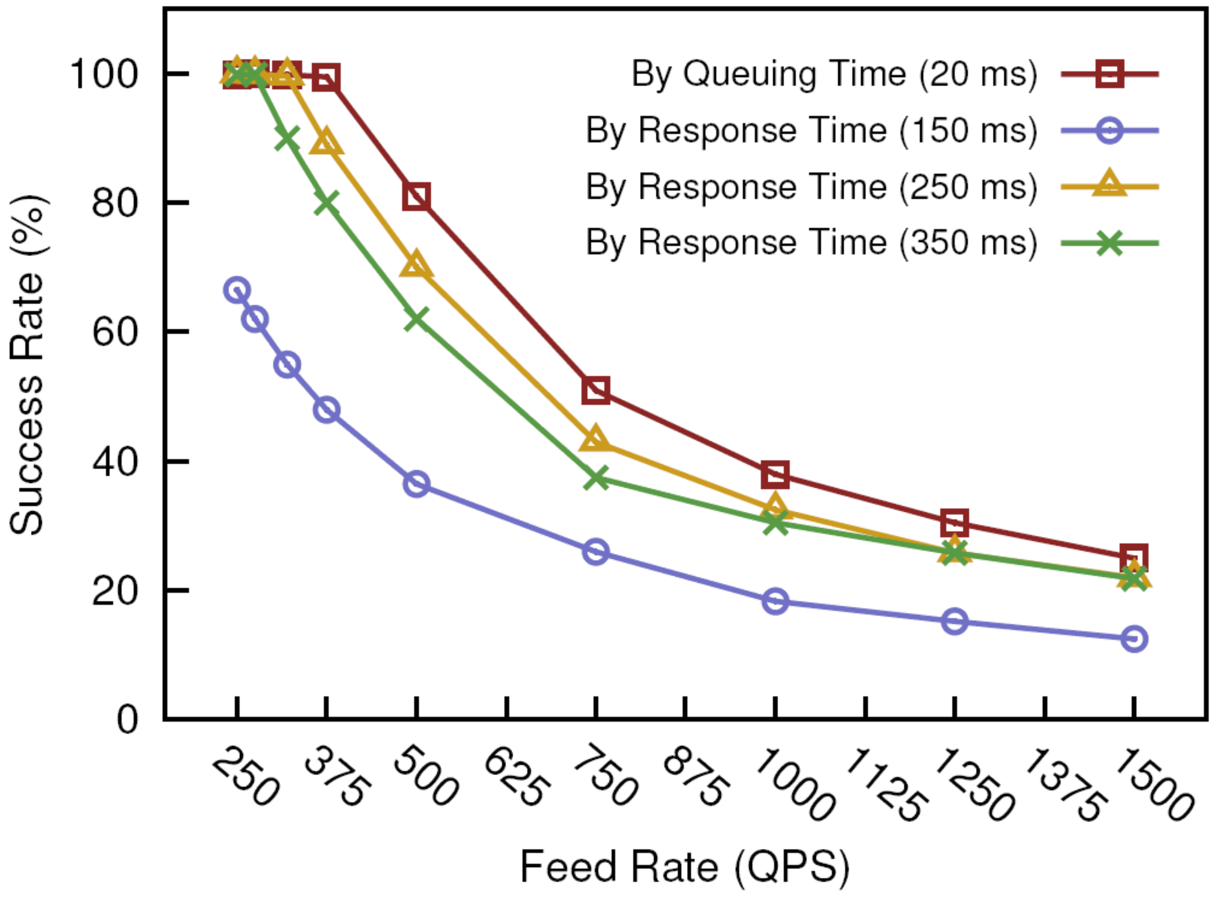
DAGOR Workflow



Collaborative Admission Control

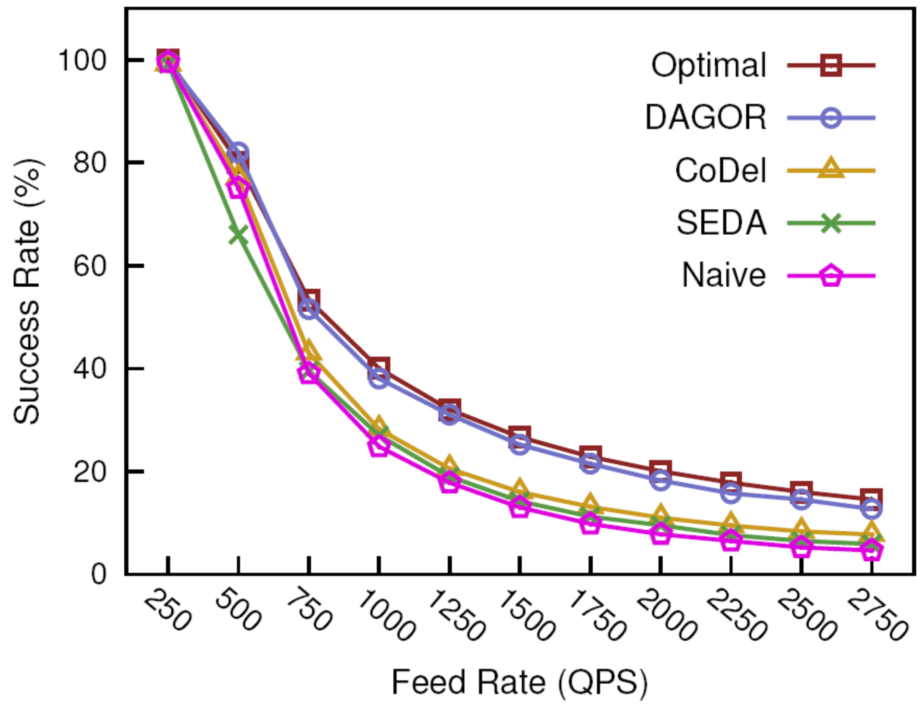
- ✓ Service agnostic
- ✓ Independent but collaborative
- ✓ Efficient and fair

Overload Detection

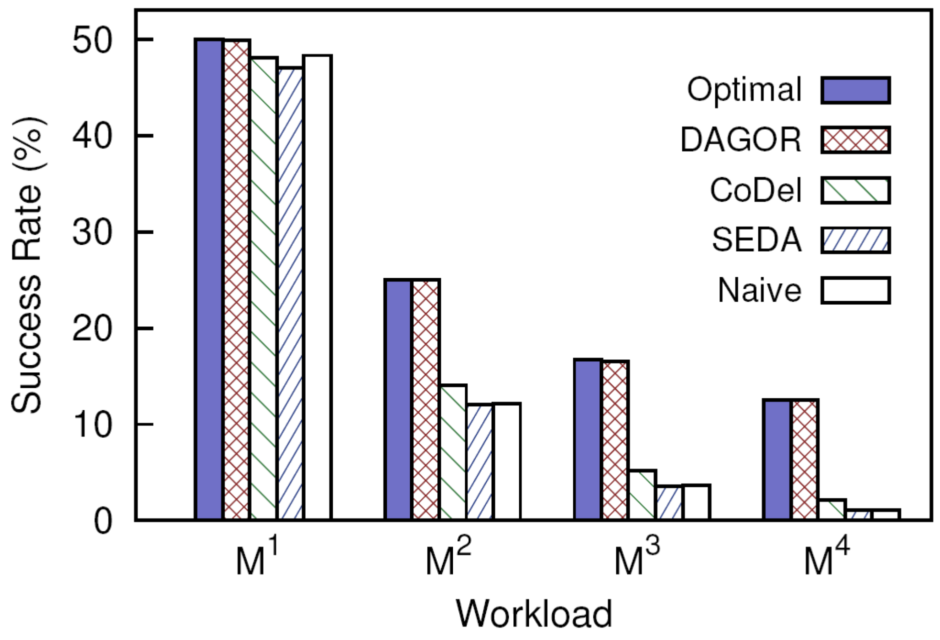


Queuing Time vs. Response Time

Scalability



Overload Control with Increasing Workload (M^2)

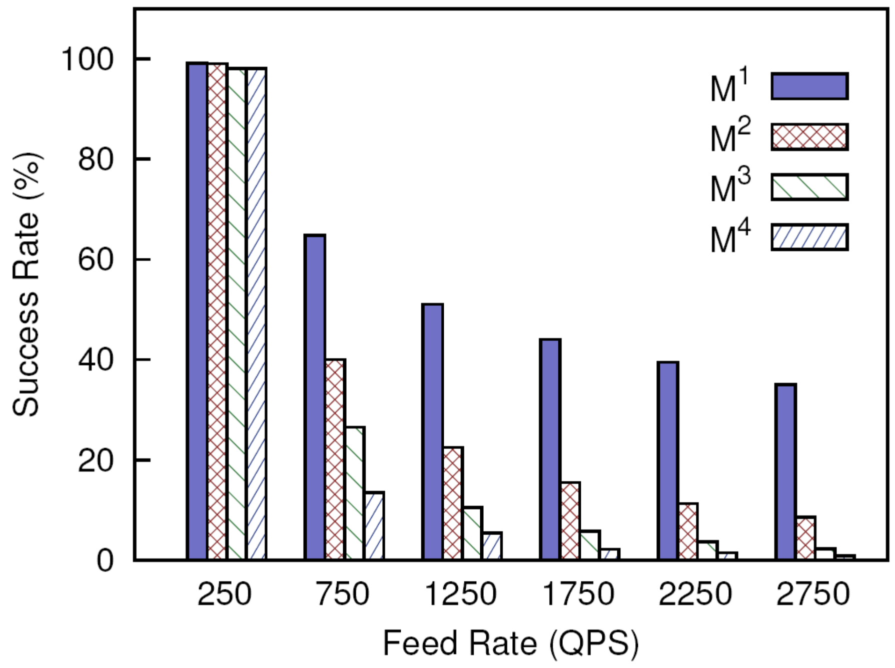


Overload Control with Different Types of Workload

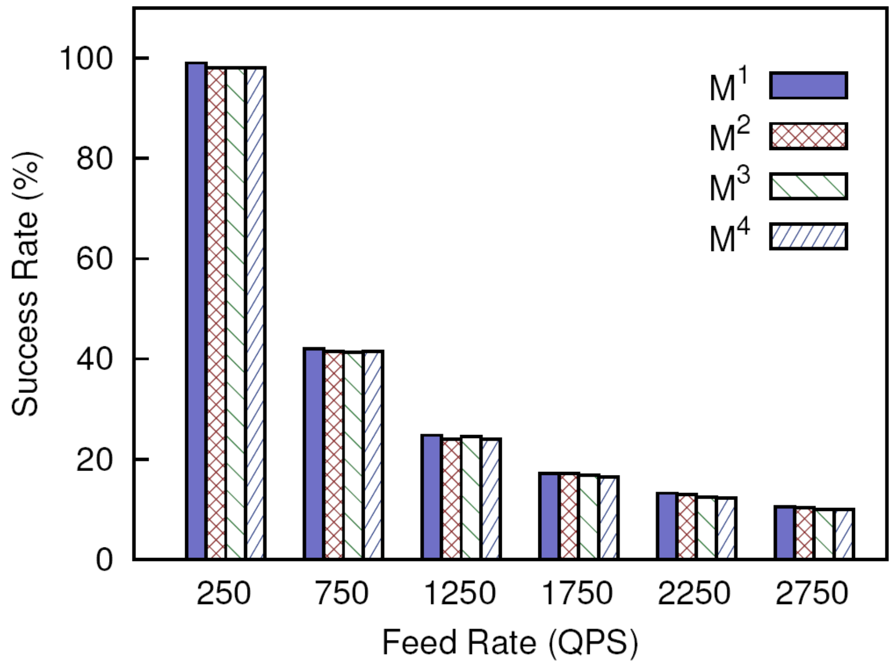
Optimal Success Rate = f_{sat}/f

Fairness

CoDel



DAGOR



Takeaways: DAGOR Design Principles

1. **Must be decentralized and autonomous in each service/node**
   - Essential for the overload control framework to scale with the ever evolving microservice system
2. **Employ feedback mechanism for adaptive load shedding**
   - Essential for adjusting thresholds automatically
3. **Prioritize user experience**

Thank You ALL!