

**Legenda das Colunas QP:**

- **QP01 (Impacto no Desempenho):** Resumo do impacto do Service Mesh (latência, throughput, CPU, RAM).
- **QP02 (Ferramentas de Monitoramento/Tracing):** Lista das ferramentas utilizadas para detecção e análise.
- **QP03 (Práticas Recomendadas c/ VNFs):** Práticas recomendadas para integração do Service Mesh com VNFs.

N. do Artigo	Nome do Artigo	Linha de Pesquisa Principal	QP01 - Qual é o impacto de um Service Mesh no desempenho de cluster Kubernetes?	QP02 - Quais ferramentas utilizadas para detecção, análise de gargalos e falhas em tempo real em cadeias de Service Mesh?	QP03 - Quais são as práticas recomendadas para a integração do Service Mesh com VNFs em cluster Kubernetes?
1	A Mirco-Service Tracing System Based on Istio and Kubernetes	Observabilidade e Tracing Distribuído	Pouco impacto no desempenho do sistema original.	Jaeger, Zipkin	N/A (Foco em tracing genérico de microsserviços)
2	A Runtime Trust Evaluation Mechanism in the Service Mesh Architecture	Segurança e Confiabilidade do Service Mesh	N/A (Foco em avaliação de confiança)	Prometheus	Implementar um Avaliador de Confiança em Tempo de Execução (RTE) no plano de controle do Service Mesh.
3	A service mesh approach to integrate processing patterns into microservices applications	Padrões de Arquitetura e Implementação	Melhora o desempenho e flexibilidade.	N/A	Utilizar estruturas descentralizadas (Eblocks) para integrar padrões de processamento, permitindo réplicas

					distribuídas.
4	A Web-based Orchestrator for Dynamic Service Function Chaining Development with Kubernetes	Service Function Chaining (SFC)	N/A (Foco em orquestração SFC)	Prometheus	Automação via orquestrador web para criar SFCs baseadas em CNFs e integrar monitoramento.
5	Adaptive and Application-agnostic Caching in Service Meshes for Resilient Cloud Applications	Desempenho e Latência do Service Mesh; Resiliência e Tolerância a Falhas	Redução de tráfego (40%) e 80% das requisições via cache.	N/A	Implementar caching agnóstico de aplicação para resiliência.
6	An Empirical Study of Service Mesh Traffic Management Policies for Microservices	Resiliência e Tolerância a Falhas	Impacto no tempo de resposta (circuit breakers, retries).	N/A	Ajustar finamente políticas de resiliência (circuit breakers em camadas baixas, retries com parcimônia).
7	An Empirical Study on Kubernetes Operator Bugs	Análise de Bugs/Problemas em Operadores	N/A (Foco em bugs de operadores)	N/A	N/A (Foco em bugs de operadores)
8	An Evaluation of Service Mesh Frameworks for Edge Systems	Desempenho e Latência do Service Mesh; Comparação de Service Mesh Frameworks	Linkerd: menor latência e menor consumo (20-43% menos memória, 22-25% menos CPU que Istio).	Prometheus	N/A (Foco em avaliação geral de SM para Edge)
9	Analyzing and Monitoring	Observabilidade e Tracing	N/A (Foco em monitoramento)	Zipkin, Envoy (plugin	N/A (Foco em monitoramento)

	Kubernetes Microservices based on Distributed	Distribuído	)	KMamiz), Kiali, Grafana	não intrusivo de microsserviços )
10	Automated Testing and Resilience of Microservice's Network-link using Istio Service Mesh	Resiliência e Tolerância a Falhas	Impacto no desempenho em testes de falha.	Jaeger, Prometheus, Grafana	Utilizar injeção de falhas para testar resiliência; implementar escalonamento , failover e circuit breakers.
11	Balancing Load An Adaptive Traffic Management Scheme for Microservices	Gerenciamento e Otimização de Configurações	Minimiza tempo de resposta, estabiliza carga.	Kiali, Prometheus, Grafana, Jaeger	Configuração dinâmica e adaptativa de tráfego com base na carga em tempo real.
12	Boosting Microservice Resilience	Resiliência e Tolerância a Falhas	Melhora estabilidade, respostas rápidas, resiliência a falhas de Pods.	N/A	Usar testes caóticos para avaliar o impacto do Istio na resiliência.
13	Dependency-A ware Traffic Management for Configuring On-demand in Service Meshes	Gerenciamento e Otimização de Configurações	Economiza armazenament o (40-60%), reduz atualizações.	Prometheus	Adotar mecanismo de gerenciamento de tráfego "configure on-demand" para evitar configurações desnecessárias .
14	Dissecting Overheads of Service Mesh Sidecars	Desempenho e Latência do Service Mesh	Aumento de latência (27-269%), CPU (42-163%); degradação de throughput	MeshInsight	Otimização do consumo de recursos (ex: Sockets de Domínio Unix para IPC).

			(15-70%). Parsing (63-77% sobrecarga), IPC/sockets (principal fonte TCP).		
15	Intelligent Service Mesh Framework for API Security and Management	Segurança e Confiabilidade do Service Mesh	N/A (Foco em segurança de API)	N/A	Implementar autenticação e autorização baseadas em identidade (mTLS, JWT); gerenciar credenciais de forma robusta.
16	Istio API Gateway Impact to Reduce Microservice Latency and Resource Usage on Kubernetes	Desempenho e Latência do Service Mesh; Comparação de Service Mesh Frameworks	Istio Ingress melhor latência > 175 RPS; Nginx Ingress melhor latência < 175 RPS. Istio Ingress > CPU/RAM.	Prometheus, Grafana	N/A (Foco em comparação de Gateways)
17	Latency and RAM Usage Comparison of Advanced and Lightweight Service Mesh	Desempenho e Latência do Service Mesh	Linkerd melhor RAM/frontend. Istio melhor latência backend.	Grafana	N/A (Foco em comparação geral de SM)
18	Microarchitect ural Analysis and Characterizatio n of Performance Overheads in Service Meshes with Kubernetes	Desempenho e Latência do Service Mesh	Latência (1-7ms/salto, 471-891% aumento); Istio: 57% sobrecarga em instruções, 16% em ciclos CPU.	Linux-perf	N/A (Foco em análise microarquitetur al de sobrecargas)
19	Service	Service	N/A (Foco em	N/A	Desacoplamen

	Function Chaining Design & Implementation Using Network Service Mesh in Kubernetes	Function Chaining (SFC)	design/implementação SFC)		to de serviços; Cluster Kubernetes bem configurado; Injeção automática de sidecar; Gerenciamento declarativo. <b>Contexto principal: CNFs/SFCs.</b>
20	Service Mesh Based Distributed Tracing System	Observabilidade e Tracing Distribuído	N/A (Foco em sistema de tracing)	Jaeger, Zipkin	N/A (Foco em sistema de tracing não intrusivo)
21	The Utilisation and Implementation of the Istio Framework in the Architectonic Progression of Online Learning Platform	Desempenho e Latência do Service Mesh	Latência reduzida (após adaptação); RPS estável (97-120 RPS); supera monolíticos em alta concorrência.	Apache JMeter	Desacoplamento de serviços; Cluster Kubernetes bem configurado; Injeção automática de sidecar; Gerenciamento declarativo; Centralização da lógica de rede. <b>Contexto principal: plataforma de aprendizado online (microsserviços, não VNF explícita).</b>
22	Threat Intelligence Sharing Component in	Segurança e Confiabilidade do Service Mesh	N/A (Foco em compartilhamento de inteligência de	Prometheus, RTE	Implementar compartilhamento de inteligência de

	the Service Mesh Architecture		ameaças)		ameaças (TIS) no plano de controle.
--	-------------------------------------	--	----------	--	---