ALIBABA CLOUD LEVERAGES eBPF FOR ADAPTIVE LAYER 7 LOAD BALANCING

OVERVIEW

Alibaba Cloud is one of the largest public cloud providers, operating a globally distributed infrastructure for millions of tenants. As their Layer 7 (L7) load balancers scaled to process over 10 million requests per second across diverse workloads, traditional Linux I/O event notification mechanisms struggled to keep up. To solve this problem, Alibaba Cloud developed Hermes, an intelligent, userspace-directed I/O event notification framework powered by eBPF. By integrating real-time feedback from application workers directly into kernel scheduling decisions, Hermes enables adaptive connection management, fine-grained performance control, and greater infrastructure efficiency. This eBPF-driven approach strengthens the stability and scalability of Alibaba Cloud's global network infrastructure, reducing operational overhead, lowering cloud infrastructure costs, and enhancing the tenant experience.

CHALLENGE

Alibaba Cloud's L7 load balancers must sustain high throughput while ensuring fair traffic distribution across multiple workers. Existing Linux mechanisms presented several limitations:

- **Unbalanced Workloads**: epoll's LIFO wakeup behavior, introduced to mitigate the thundering herd problem, caused connection concentration on recently active workers, while SO_REUSEPORT's static hashing failed to detect overloaded or hung processes.
- **Limited Visibility**: Kernel-level dispatch lacked insight into userspace worker states such as pending events and connection counts, preventing adaptive scheduling.
- **Operational Risk**: Modifying the kernel was not feasible at scale, where even minor bugs could cause large-scale service disruptions.
- **Performance Sensitivity**: Maintaining low latency under multi-tenant workloads required extremely low scheduling overhead and real-time feedback.
- **Tenant Performance Isolation**: Load-balancing mechanisms failed to evenly distribute resources across tenants, causing performance imbalances in shared environments.

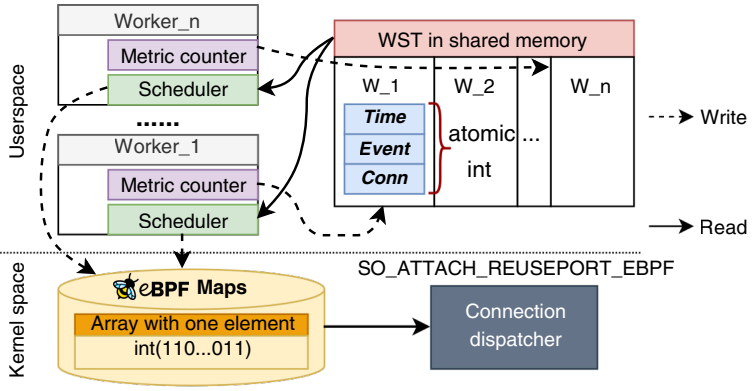
To address these issues, Alibaba Cloud sought a solution that could safely extend kernel scheduling behavior while incorporating dynamic, userspace-driven intelligence.

SOLUTION

Alibaba Cloud created Hermes, a userspace-directed I/O event notification framework that leverages eBPF to augment the Linux kernel with real-time worker metrics. This framework enables closed-loop coordination between userspace and the kernel, ensuring that connection dispatch decisions reflect actual workload conditions:

- **Closed-Loop Scheduling**: Userspace workers continuously publish metrics—such as availability, pending events, and active connections to shared memory. An eBPF program reads these metrics and dynamically selects the most suitable worker.
- **Lock-Free Synchronization**: Worker status is exchanged through atomic operations on eBPF array maps, enabling high-speed updates without locks or contention.
- **Scalable Metric Design**: Lightweight metrics allow real-time updates per I/O event with negligible overhead.
- **Two-Stage Load Distribution**: Userspace preselects eligible workers, and the kernel refines selection using a hash of the connection context.
- **Production-Safe Implementation**: The kernel dispatch logic running in eBPF prevents loops, recursion, and unbounded execution, using bitwise operations for minimal CPU overhead, and is production-ready at scale.

This architecture enables Hermes to adapt continuously to runtime conditions while preserving system stability and efficiency.



RESULTS

Hermes has delivered measurable and sustained improvements across Alibaba Cloud's L7 load balancing systems:

- **19% reduction in unit infrastructure cost** by increasing safe CPU utilization thresholds from 30% to 40% without compromising stability.
- **99.8% reduction in daily worker hangs**, as measured by health probes exceeding 200ms latency.
- **Consistent performance** across diverse traffic models, with best or near-best latency and throughput in all cases.
- **Proven scalability, deployed on 100,000 CPU cores** and handling **10 million requests per second** for more than two years.
- **Enhanced reliability** and reduced operational overhead.

WHY eBPF?

eBPF was chosen for its unique ability to safely extend kernel functionality without modifying kernel code. It provides a verified, sandboxed environment for custom logic, ensuring both safety and stability at a massive scale.

- **Non-Intrusive Customization**: The ability to safely extend kernel behavior eliminates the risk of bugs crashing the kernel that would typically lead to costly maintenance.
- **Stable API**: The CO-RE capability allows for compatibility across kernel versions without conditional compilation.
- **Efficient Coordination**: Array maps and atomic operations enable lock-free synchronization between userspace and the kernel for scheduling decisions.

By leveraging eBPF, Alibaba Cloud successfully scaled its load-balancing infrastructure to safely and efficiently handle dynamic, multi-tenant workloads.

FUTURE PLANS

- **Broader Adoption**: Generalizing the framework for other epoll-based applications such as Redis, Envoy, and Nginx through a shared SDK.
- **Cache-Aware Scheduling**: Introducing group-based worker selection models to balance load distribution and instruction/data cache reuse.
- **Expanded Use of eBPF**: Applying eBPF for future in-kernel control-plane innovations beyond load balancing.