

March 2026

Kernel Observability

The Missing Layer in Cloud Native Engineering



AI Agents are the new workload



Dangerously skip permissions in prod



Let's create some CVE slop

```
kubectl exec -it claude-code-observed -c claude -- bash -c '
  chown -R node:node /home/node /usr/local/go/src/ /tmp;
  su - node -c "claude --dangerously-skip-permissions -p \"Audit
the Go standard library source code at /usr/local/go/src/ for
security vulnerabilities. Focus on crypto/, net/, and os/. Read
as many source files as possible. For each finding, read the
relevant file again to confirm. Write a detailed security audit
report to /tmp/go-stdlib-audit.md with code snippets.\""'
```

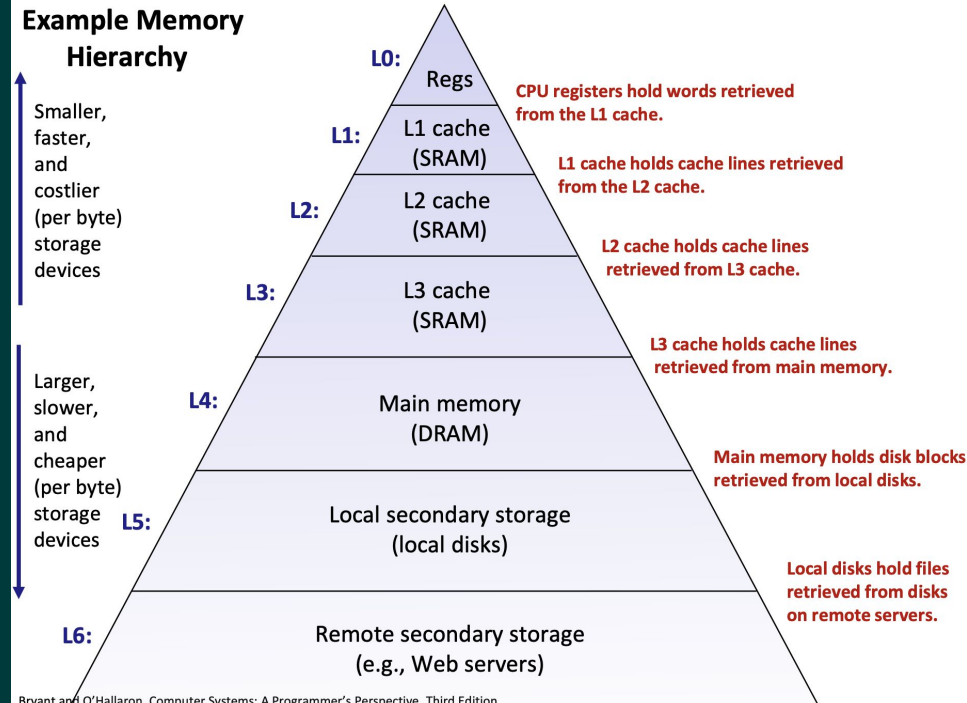
Agents Are Not Microservices

Standard Kubernetes workloads	AI Agents
Memory is <i>proportional to load</i>	Memory <i>grows with context window</i>
Workloads are short-lived	Non-deterministic: <i>minutes to hours</i> . State is irreplaceable
OOMkills are recoverable (stateless replicas)	35 minutes of work, gone

**Let's check in on
the dashboard**

What Every Programmer Needs to Know About Memory

Example Memory Hierarchy



Latency Comparison Numbers (~2012)

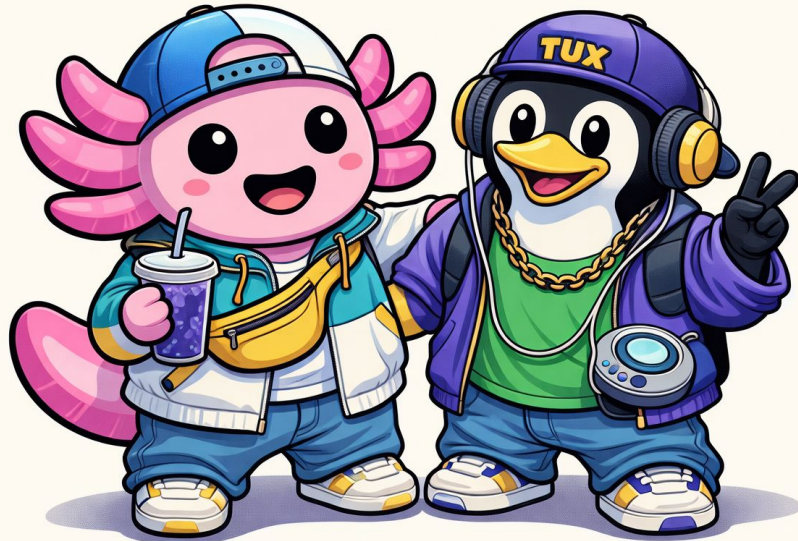
L1 cache reference	0.5	ns		
L2 cache reference	7	ns		
Main memory reference	100	ns		
Read 4K randomly from SSD*	150,000	ns	150	us
Read 1 MB sequentially from memory	250,000	ns	250	us
Read 1 MB sequentially from SSD*	1,000,000	ns	1,000	us
Disk seek	10,000,000	ns	10,000	us
10 ms				
Send packet CA->Netherlands->CA	150,000,000	ns	150,000	us
150 ms				

1 ms = 1,000 us = 1,000,000 ns

Cgroups: The Shadow on the Wall

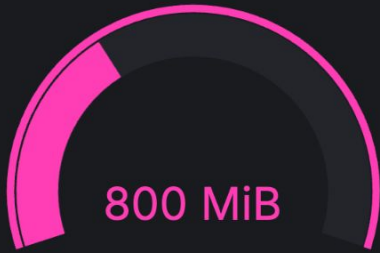


What If Every Workload Had Its Own Kernel?

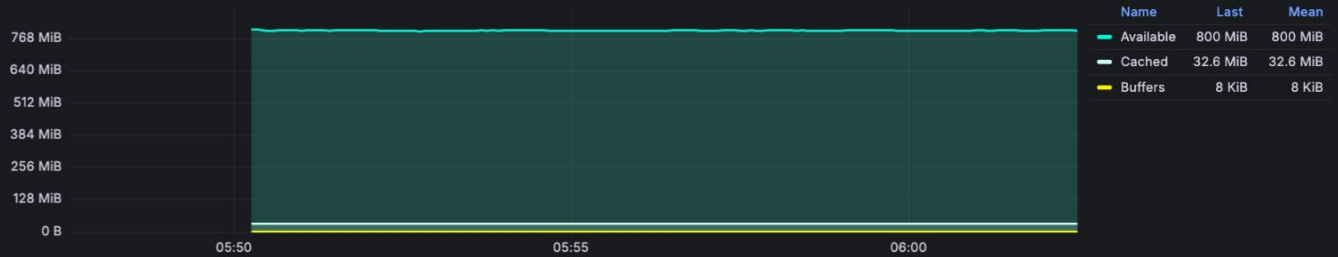


Memory: The kernel sees reality

Memory Available (Edera Zone)



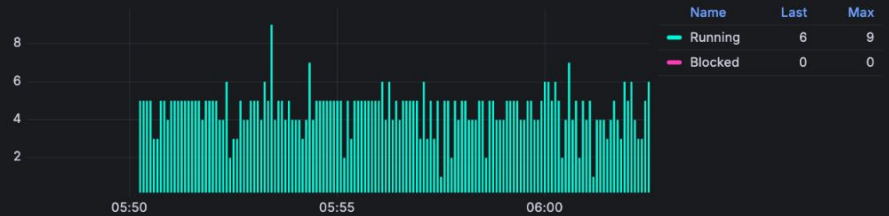
Zone Kernel Memory - Isolated View



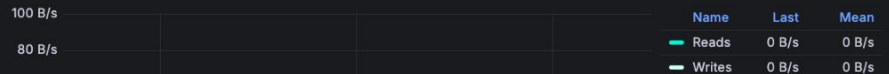
Page Fault Rate (YOUR Zone Only)



Process States (Zone Processes)



Disk I/O Throughput (Zone Storage)



Monitoring Claude Code's Performance



Life of an OOMkill

- Process calls `malloc()`
- The kernel can't find any `reclaimable memory`
- All page cache is `dirty` (not yet flushed to disk)
- `MemAvailable` is effectively `zero`
- Kernel scores processes by memory and `kills the highest score`
- For AI agents, OOMkill is `catastrophic`, everything is lost



Availability Matrix

Metric	Shared kernel	Dedicated kernel observability
MemAvail	No cgroups v2 support	Per-workload
Reclaimable memory	No cgroups v2 support	Per-workload
Processes running	No cgroups v2 support	Per-workload
I/O scheduler	No cgroups v2 support	Per-workload
Context switches	Noisy (node-wide)	Per-workload
Page faults	Per-workload	Per-workload

Thank You

Jed Salazar
Field CTO
jed@edera.dev