

Those are the wrong metrics. Please reload your previous answer with this new cgroups v2 list i'm providing:

```
root@device1:/# ls sys/fs/cgroup/
cgroup.controllers cpuset.cpus.partition memory.events.local
cgroup.events cpuset.mems memory.high
cgroup.freeze cpuset.mems.effective memory.low
cgroup.kill hugetlb.1GB.current memory.max
cgroup.max.depth hugetlb.1GB.events memory.min
cgroup.max.descendants hugetlb.1GB.events.local memory.numa_stat
cgroup.procs hugetlb.1GB.max memory.oom.group
cgroup.stat hugetlb.1GB.rsvd.current memory.pressure
cgroup.subtree_control hugetlb.1GB.rsvd.max memory.stat
cgroup.threads hugetlb.2MB.current memory.swap.current
cgroup.type hugetlb.2MB.events memory.swap.events
cpu.idle hugetlb.2MB.events.local memory.swap.high
cpu.max hugetlb.2MB.max memory.swap.max
cpu.max.burst hugetlb.2MB.rsvd.current misc.current
cpu.pressure hugetlb.2MB.rsvd.max misc.max
cpu.stat io.max pids.current
cpu.uclamp.max io.pressure pids.events
cpu.uclamp.min io.prio.class pids.max
cpu.weight io.stat rdma.current
cpu.weight.nice io.weight rdma.max
cpuset.cpus memory.current
cpuset.cpus.effective memory.events
```

Here's a filtered list of **key saturation-focused metrics** from your provided cgroups v2 files, prioritized by their ability to detect resource contention and bottlenecks:

Core Metrics for Saturation Monitoring

Category	Metric Path	Monitoring Purpose	Critical Thresholds
CPU	cpu.pressure	% time tasks wait for CPU (some=partial stalls, full=total stalls)	some >10% over 1m
	cpu.stat (usage_usec)	Total CPU time consumed (identify over-limit groups)	Compare with cpu.max quota

Category	Metric Path	Monitoring Purpose	Critical Thresholds
	<code>cpu.max</code>	Hard CPU limit (quota/period)	Usage nearing quota = saturation risk
Memory	<code>memory.current</code>	Total RAM + swap used	Compare with <code>memory.high/max</code>
	<code>memory.events</code> (high/oom)	Count of soft/hard limit breaches	oom > 0 = critical
	<code>memory.swap.current</code>	Swap usage (indicates memory pressure)	Sudden spikes = thrashing
	<code>memory.pressure</code>	Memory reclaim stalls (some=partial, full=complete)	some > 15% over 5m
I/O	<code>io.pressure</code>	Block I/O latency stalls	some > 5% for SSDs, > 10% for HDDs
	<code>io.stat</code> (rbytes/wbytes, rios/wios)	Read/write throughput & IOPS	Compare with <code>io.max</code> limits
PID	<code>pids.current</code>	Process/thread count	Approaching <code>pids.max</code> = fork saturation

Honorable Mentions (Niche Cases)

Metric	Use Case
<code>cpuset.cpus.effective</code>	Detect CPU core isolation failures
<code>memory.numa_stat</code>	NUMA node imbalance (high latency)
<code>hugetlb.2MB.events</code> (max)	Huge page allocation failures

Excluded Metrics (Non-Saturation Reasons)

- `cgroup.*` (except `.events`): Hierarchy control, not resource usage
- `cpu.weight*/io.weight`: Priority settings, not saturation indicators
- `memory.events.local`: Redundant with parent cgroup monitoring
- `*.rsvd.*` (hugetlb): Niche reservation tracking

Recommended Alerts (Prometheus Rules)

```
# CPU Pressure
- alert: CGroupsCPUPressure
  expr: rate(cpu_pressure_some_ratio[2m]) > 0.1 # >10% stalled

# Memory Swap Thrashing
- alert: CGroupsSwapUsageSpike
  expr: deriv(memory_swap_current_bytes[5m]) > 1e8 # 100MB/s increase
```

```
# I/O Saturation
- alert: CGroupsIOStalled
  expr: rate(io_pressure_some_ratio[2m]) > 0.2 # >20% I/O stalls

# PID Contention
- alert: CGroupsPidLimitNear
  expr: pids_current / pids_max > 0.8 # 80% of PID limit
```

These metrics directly correlate with **resource starvation symptoms** (stalls, OOM kills, queue buildup). Use Telegraf's `cgroup` input plugin to scrape them, adding labels for cgroup hierarchy (e.g., container ID).