Retroview Manual

Retroview

Flussonic

Table of contents

1. Products

1.1 Retroview - Monitoring System

Retroview is a comprehensive monitoring and troubleshooting system designed for video streaming service operators and system administrators. It provides real-time monitoring of all video streams, automatic issue detection, and advanced troubleshooting capabilities to maintain service quality.

1.1.1 Overview

Retroview addresses critical challenges faced by video streaming operators:

- **Rapid Issue Detection**: Quickly identify the source of poor video quality when complaints arise
- **Comprehensive Monitoring**: Real-time monitoring of all video stream health and performance
- **Proactive Alerting**: Configure alerts for server and video stream problems before users notice
- **Root Cause Analysis**: Pinpoint exact source of quality degradation in complex streaming infrastructure

Target Audience

- **Video Streaming Service Operators**: Monitor and maintain streaming service quality
- **System Administrators**: Track server health and infrastructure performance
- **NOC Teams**: 24/7 monitoring and incident response
- **Quality Assurance**: Verify streaming quality and compliance

1.1.2 Key Capabilities

Issue Detection and Troubleshooting

Finding Source of Poor Video Quality:

When users complain about video quality issues, Retroview helps you:

- **Trace Stream Path**: Follow video stream through entire infrastructure
- **Identify Bottlenecks**: Pinpoint exact point where quality degrades
- **Analyze Metrics**: Review bitrate, framerate, resolution, and codec issues
- **Historical Analysis**: Compare current state with historical performance data

Diagnostic Tools:

- Stream topology visualization
- Real-time quality metrics
- Frame-by-frame analysis capabilities
- Network path tracing
- Server performance correlation

Comprehensive Stream Monitoring

Real-time Monitoring:

Retroview continuously monitors all video streams across your infrastructure:

- **Video Quality Metrics**:
- Bitrate stability and variations
- Frame rate consistency
- Resolution accuracy
- Codec performance
- Audio/video synchronization
- **Stream Health Indicators**:
- Connection status
- Packet loss and errors
- Buffer health
- Latency measurements
- Jitter analysis
- **Infrastructure Monitoring**:
- Server resource utilization
- Network bandwidth usage
- Storage performance
- Processing pipeline status

Alert Configuration

Proactive Problem Detection:

Configure intelligent alerts for various failure scenarios:

Server Alerts:

- CPU overload warnings
- Memory exhaustion alerts
- Storage capacity thresholds
- Network connectivity issues
- Service availability monitoring

Video Stream Alerts:

- Video quality degradation
- Stream connection failures
- Bitrate drop below threshold
- Frame rate instability
- Audio/video desynchronization
- Black screen or frozen frame detection
- Stream startup failures

Alert Delivery Methods:

- Email notifications
- SMS/mobile alerts
- Webhook integrations
- Dashboard notifications
- Integration with incident management systems

1.1.3 Core Features

Real-time Dashboards

- **Overview Dashboard**: High-level view of entire streaming infrastructure
- **Stream Details**: Detailed metrics for individual streams
- **Server Health**: Comprehensive server performance monitoring
- **Alert Management**: Centralized alert viewing and management
- **Custom Dashboards**: Create custom views for specific needs

Historical Data Analysis

- **Performance Trends**: Track quality metrics over time
- **Capacity Planning**: Analyze growth trends for infrastructure planning
- **Incident Reports**: Generate reports on past incidents
- **SLA Compliance**: Track service level agreement metrics
- **Comparative Analysis**: Compare performance across different time periods

Integration Capabilities

- **Flussonic Integration**: Native integration with Flussonic Media Server
- **Mcaster Integration**: Full support for Mcaster infrastructure
- **Third-party Systems**: REST API for external integrations
- **Monitoring Tools**: Integration with Prometheus, Grafana, and other tools
- **Incident Management**: Integration with PagerDuty, Opsgenie, and similar platforms

1.1.4 Use Cases

Complaint Investigation

Scenario: User reports poor video quality on specific channel

Retroview Solution:

1. **Locate Stream**: Quickly find affected stream in monitoring dashboard
2. **Review Metrics**: Check current and historical quality metrics
3. **Trace Path**: Follow stream through infrastructure to identify issue point
4. **Identify Cause**: Determine if issue is at source, transcoding, or delivery
5. **Resolve**: Take corrective action based on identified root cause
6. **Verify**: Confirm resolution through continued monitoring

Proactive Monitoring

Scenario: Prevent issues before users notice

Retroview Solution:

- **Continuous Monitoring**: All streams monitored 24/7
- **Early Warning**: Alerts triggered before critical thresholds
- **Automatic Detection**: AI-powered anomaly detection
- **Trend Analysis**: Identify degradation patterns early
- **Preventive Action**: Fix issues before they impact users

Infrastructure Management

Scenario: Manage large-scale streaming infrastructure

Retroview Solution:

- **Centralized View**: Monitor hundreds or thousands of streams from single interface
- **Server Fleet Management**: Track all server performance metrics
- **Capacity Planning**: Use historical data for scaling decisions
- **Load Balancing**: Identify overloaded servers and redistribute load
- **Maintenance Planning**: Schedule maintenance based on usage patterns

1.1.5 Technical Architecture

Data Collection

- **Agent-based Monitoring**: Lightweight agents on each server
- **API Integration**: Direct integration with streaming servers
- **Network Monitoring**: Passive network traffic analysis
- **Log Aggregation**: Centralized log collection and analysis

Metrics Processing

- **Real-time Processing**: Sub-second metric updates
- **Time-series Storage**: Efficient storage of historical data
- **Aggregation**: Statistical aggregation for trend analysis
- **Correlation**: Automatic correlation of related metrics

Alert Engine

- **Rule-based Alerts**: Configure custom alert rules
- **Threshold Monitoring**: Trigger alerts on threshold violations
- **Anomaly Detection**: Machine learning-based anomaly detection
- **Alert Aggregation**: Group related alerts to reduce noise
- **Escalation Policies**: Configurable alert escalation workflows

1.1.6 Getting Started

Initial Setup

1. **Deploy Retroview**: Install Retroview monitoring service

2. **Configure Sources**: Add streaming servers to monitoring
3. **Set Thresholds**: Configure alert thresholds for your environment
4. **Test Alerts**: Verify alert delivery mechanisms
5. **Train Team**: Familiarize operators with dashboard and tools

Best Practices

- **Start Simple**: Begin with critical streams, expand coverage gradually
- **Tune Thresholds**: Adjust alert thresholds to reduce false positives
- **Regular Reviews**: Periodically review and update monitoring rules
- **Document Procedures**: Create runbooks for common issues
- **Team Training**: Ensure all operators understand monitoring tools

Performance Optimization

- **Agent Configuration**: Optimize monitoring agent resource usage
- **Metric Selection**: Monitor essential metrics, avoid over-monitoring
- **Storage Management**: Implement retention policies for historical data
- **Network Impact**: Minimize monitoring overhead on production network

1.1.7 Troubleshooting with Retroview

Common Scenarios

POOR VIDEO QUALITY INVESTIGATION

1. Check stream quality metrics in Retroview dashboard
2. Review recent alerts and warnings for affected stream
3. Analyze bitrate graphs for drops or instability
4. Check server CPU/memory at time of issue
5. Trace stream path to identify failing component
6. Verify network connectivity and bandwidth
7. Review source stream quality if transcoding is involved

SERVICE AVAILABILITY ISSUES

1. Check server availability in Retroview
2. Review infrastructure-wide alerts
3. Analyze network connectivity metrics
4. Check for cascading failures
5. Verify load balancer health
6. Review recent configuration changes
7. Analyze resource exhaustion patterns

PERFORMANCE DEGRADATION

1. Monitor resource utilization trends
2. Identify increasing load patterns
3. Check for capacity saturation
4. Analyze network congestion
5. Review storage I/O performance

6. Check for memory leaks or resource leaks
7. Plan capacity upgrades based on trends

2. Dashboards

2.1 Server Stats Dashboard

The Server Stats dashboard provides high-level monitoring of server load metrics across your entire streaming infrastructure.

Its overview section is similar to standard server monitoring tools, but the important thing is that it's already included in the service and you don't need to monitor additional agents.

2.1.1 Most Overloaded Servers

This panel displays servers with the highest resource utilization across key measured parameters.

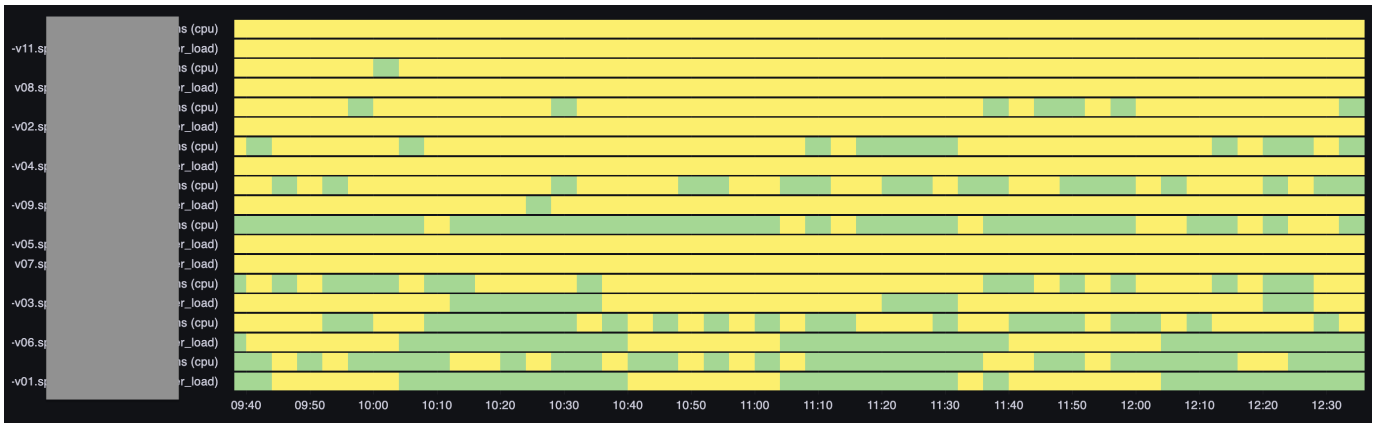
Displayed Information:

- Servers with maximum load on at least one parameter are raised to the top
- Load distribution over time. It's recommended to change the time range - narrowing it down to hours and minutes may bring servers with recent issues to the top

Use Cases:

- Select the last 2-3 hours in the dashboard, see servers with red indicators, proceed to resolve the issue
- Select the last 7 days in the dashboard, see periodic load spikes up to red levels. Identify the source of periodic growth and if it shouldn't exist, resolve the issue
- Select the last 30 days in the dashboard, see steady load growth, start planning infrastructure expansion

Overloaded Service Example



On this service, the CPU and load situation is all in the yellow zone. This can be considered efficient server utilization because there are no signs of plateauing on the CPU graph, however these servers cannot handle any additional load.

2.1.2 CPU and Virtual Machine Load

The following two panels provide an overview of central processor and virtual machine load. These are different metrics - you cannot rely on just one.

A common mistake by system administrators is making decisions without knowledge of virtual machine operation and complaining about CPU growth without considering the scheduler.

80% CPU load is not critical, although elevated. The streaming server's operability must be assessed by scheduler load, which is a more reliable metric. Virtual machine operation may involve high CPU load, and in special cases, 100% full load of several CPU cores is normal.

Important: you cannot extrapolate CPU load by adding several streams and expect linear growth. The internal mechanisms that allow scaling a streaming server to thousands of simultaneous streams have certain costs, so growth will be non-linear.

On this graph and further graphs are divided by versions. Very convenient to track: whether there was actual performance degradation or just an illusion.

Critical Load Example

On these two graphs, you can see that CPU is at the limit (this service was mentioned above) and possibly hitting a plateau, i.e., not coping, but the scheduler graph shows there's no plateau - the server is just at the limit. However, you'll see later that its disks are not coping.

2.1.3 Memory Utilization

RAM usage monitoring on servers.

Shows total memory usage - this is the graph to focus on.

Normal state is stable.

The graph doesn't show swap usage for one simple reason: on a streaming server, swap should be disabled. It's not needed in any scenario, and it can lead to the system going into failure instead of emergency shutdown, resulting in dozens of minutes of downtime instead of a few minutes.

2.1.4 Disk Write Errors

Tracking disk write errors.

Displayed Metrics:

- Collapsed writes to disk
- Failed write attempts

When storage begins to lag in writing (this is normal behavior for network storage, which very often cannot deliver constantly stable and predictable write speeds for weeks), the media server first starts grouping adjacent writes. Each such grouping is reflected in the `Collapsed writes` graph and represents an alarming situation. This shouldn't happen, but it's not a problem yet.

Typically, after prolonged ignoring of collapsed writes, you can observe write failures: `failed writes`. The media server cannot keep segments in the write queue forever - it stores them only as long as they are available for live viewing, so when the playback window passes and the write hasn't occurred, the video is lost irretrievably.

This is already a serious service failure.

Use Cases:

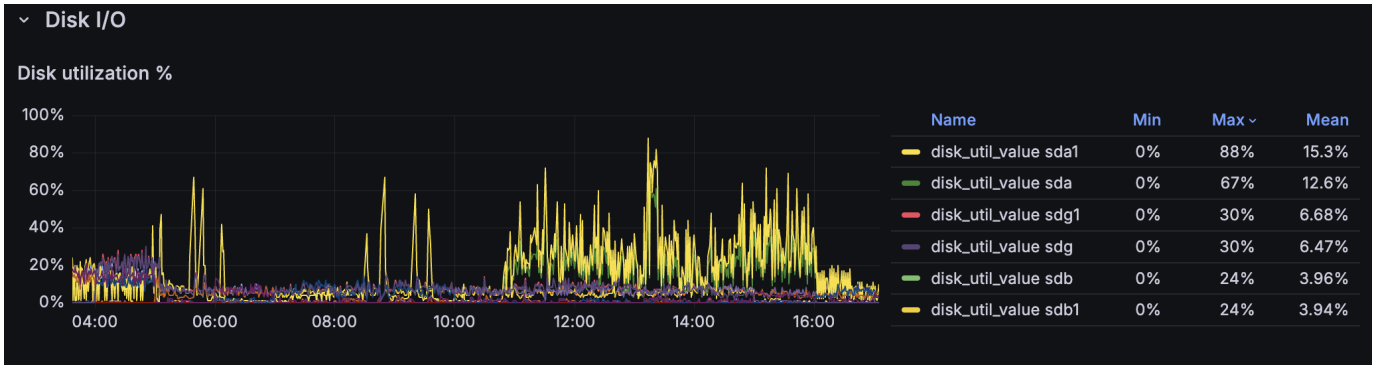
- Detect failing storage hardware
- Identify filesystem issues
- Plan disk replacement
- Monitor storage reliability

2.1.5 Disk Utilization

Monitoring disk operation speed allows you to see potential problems on individual disks.

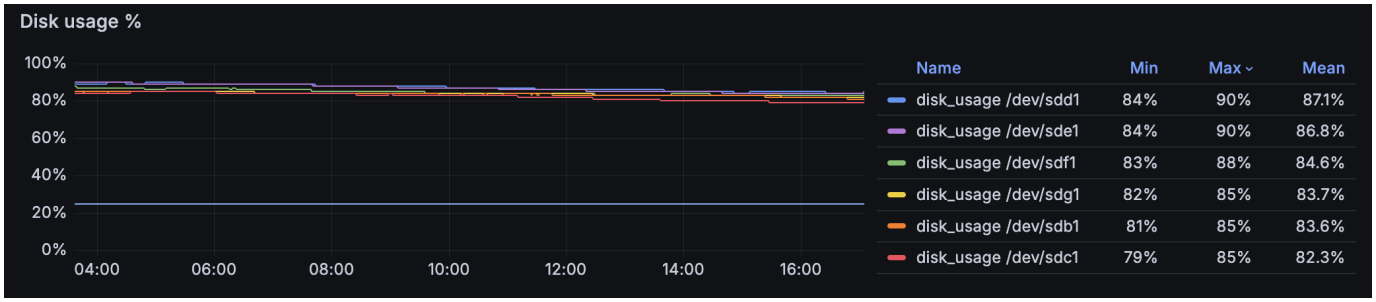
This section makes sense to visit if there are indications of disk write or read errors. Otherwise, this section is informational in nature - check once a month that there are no anomalies.

Disk I/O Percentage



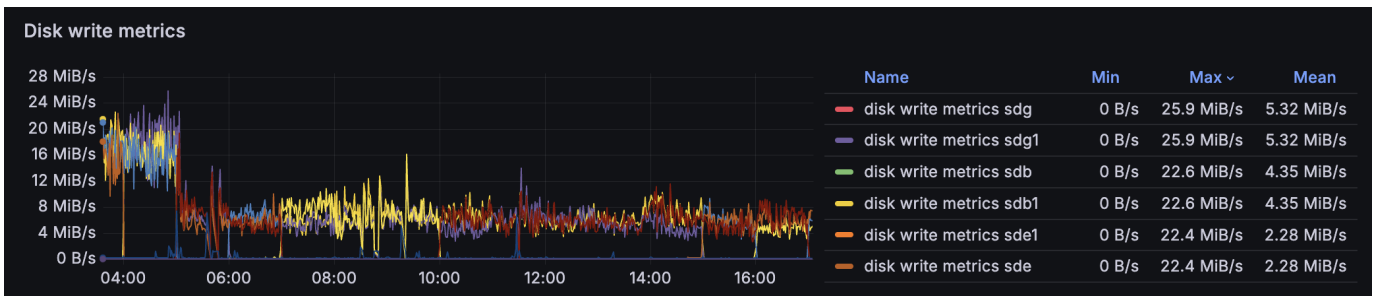
Normally should be no more than 80%. Individual spikes are acceptable, nothing critical if they don't lead to service operation errors.

Disk Fill Level



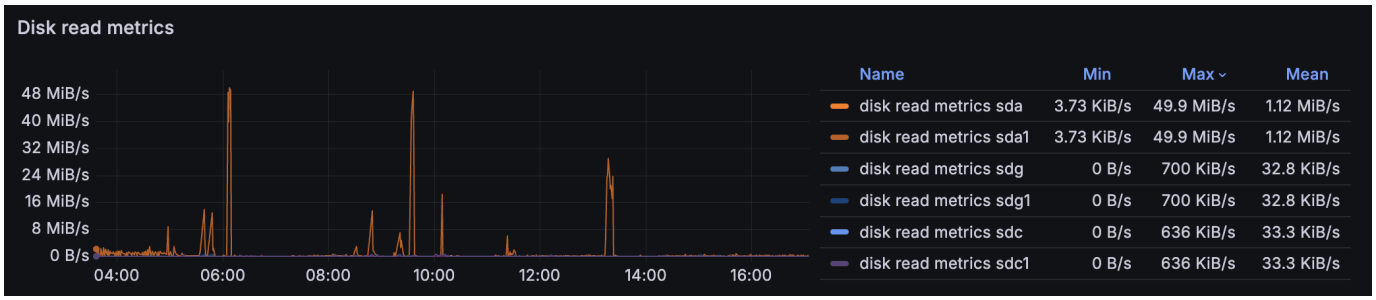
Should be stable around the limit. 98% is normal if your storage uses ext4. If you decided to use btrfs, service failure is possible at 55%, but we won't be able to help you.

Disk Write Speed



There's no single norm - it differs by orders of magnitude for spinning disks and NVMe. Sharp jumps or plateaus are of interest.

Disk Read Speed



Similar to the previous: there's no single norm. Pay attention to sharp changes, as well as trends over months.

Example of Poor Disk Choice

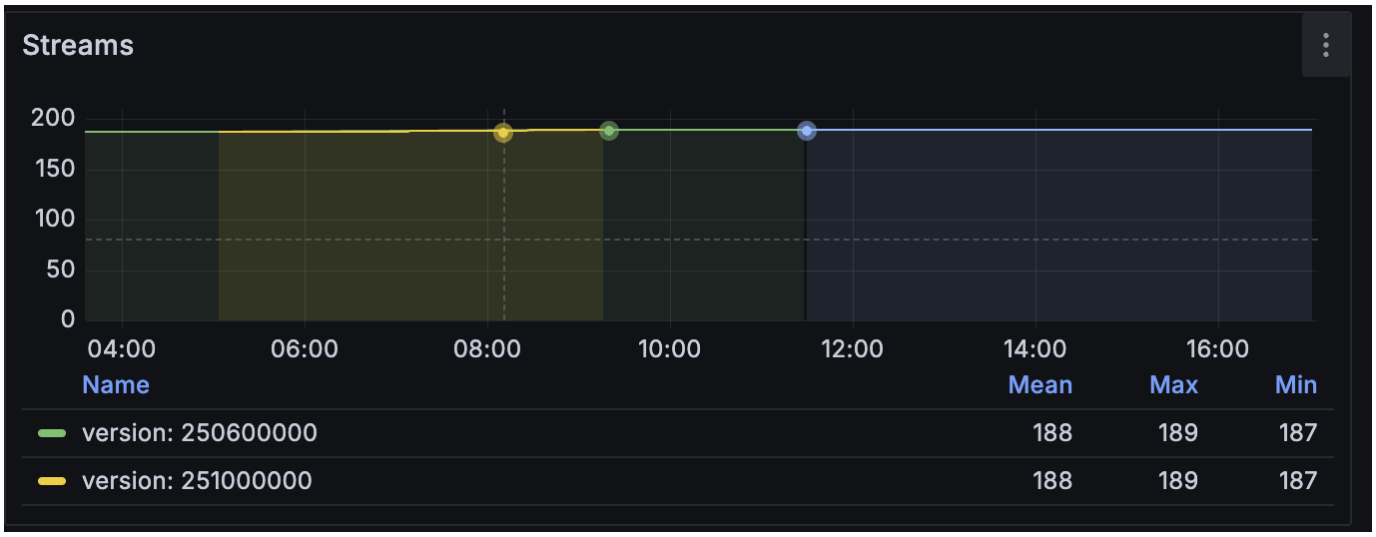
From the graphs above, you can draw the following conclusions:

- Disks were purchased very large and wasted - they cannot be filled with data
- Some disks are overloaded with writes. It might make sense to contact support for help finding a strategy for more even disk load distribution
- Flussonic RAID copes with protecting adjacent disks from overload. Despite one failing disk, the rest handle the workload

2.1.6 Streams and Clients

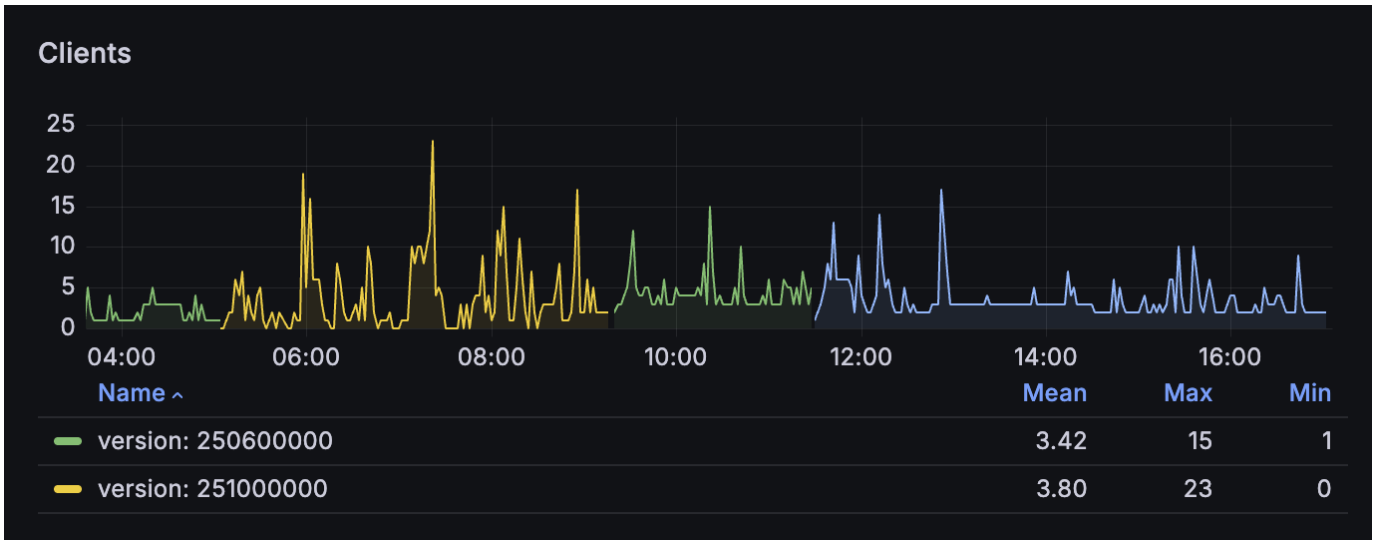
Monitoring of active streams and connected clients. This is an overview graph, more detailed picture is on adjacent dashboards.

Number of Streams



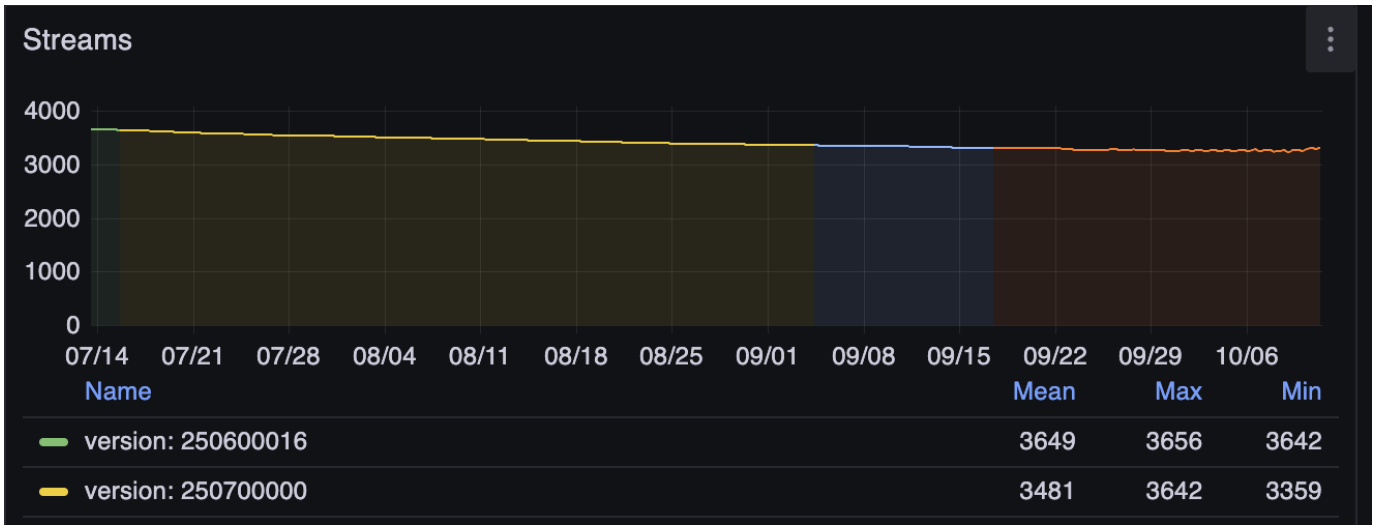
Recommended to monitor during upgrades and ensure the picture doesn't change after upgrade.

Number of Clients



Recommended to monitor during upgrades and combine with load balancer operation, ensure the number reaches the pre-upgrade level.

3-Month Retrospective



You can see that on this server over 3 months, the number of streams didn't change sharply - they are smoothly migrated to new servers.

2.1.7 Network Traffic

Monitoring of incoming and outgoing network traffic.

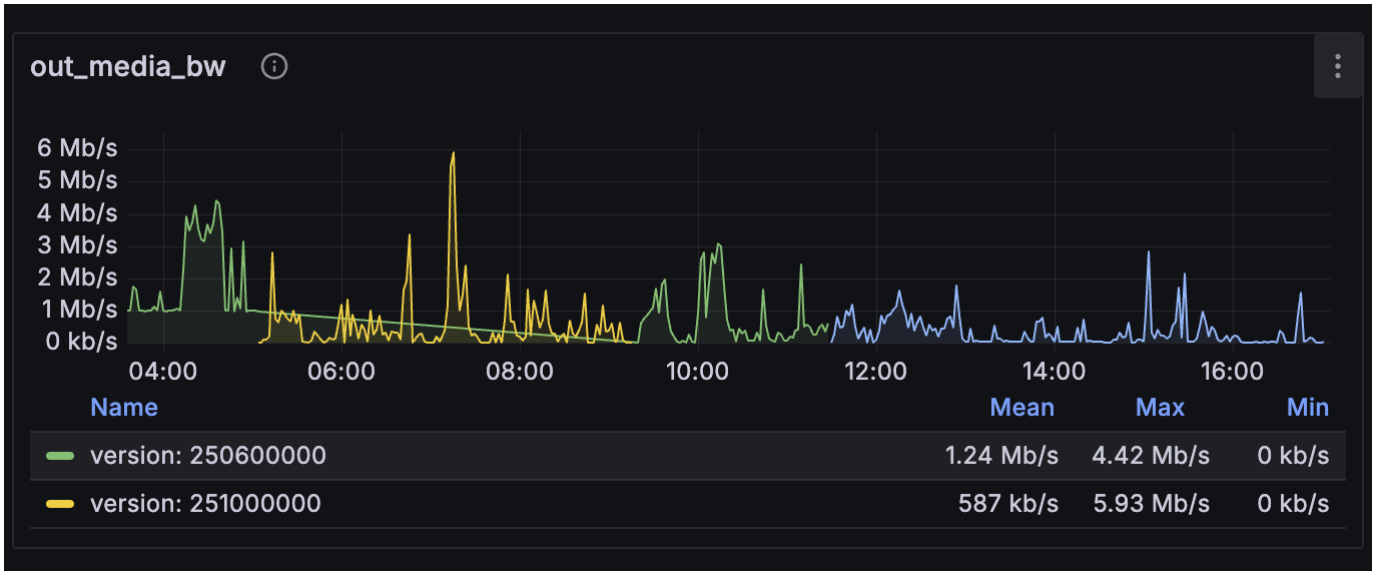
Incoming Traffic to Media Server

Gives an idea of how much incoming traffic the media server itself sees. Sometimes can differ radically from the system graph if there's capture from ASI, SDI, or loopback.

Incoming System Traffic

How much traffic enters the operating system. Can differ noticeably and be the cause of network failures if there are other traffic consumers besides the media server.

Outgoing Media Server Traffic



Outgoing System Traffic