

Spec-Guide Scaler Node Capacity Analysis:  
Charter Internal Note.  
Draft Version 1.0.0

Brad Schoenrock  
Video Operations Engineering  
Charter Communications  
Greenwood Village, CO

July. 2019

## Contents

# 1 Introduction

The scaler cluster is used to deliver lower resolution live video feeds from local market hubs that go to the small and medium tiles in Spec-Guide. These streams are sent out in a multicast, so each piece of content only has to be scaled once and then provided for every market which can use that content. Due to the nature of scaling the content and providing it as a multicast the load on the scaler cluster does not fluctuate based on concurrency but is constant throughout the day.

There are two scaler clusters, one in Saint Louis (SLDCMO), and one in Peakview (PVDCCO). The SLDCMO cluster has 84 blades which serve [PLD-COR, RENONV, EDPRMN], while the PVDCCO cluster has 72 blades which serve [MDDCWI, KNWDML, SLDCMO, SLDCLA, SPDCSC]. The scaled tiles feature is currently turned on in the 8 QAM markets listed.

Scalers have a definition of degraded similar to stitchers. They go degraded when memory reaches 85% use, CPU reaches 80% use, or disk space reaches 95% use. When a scaler goes degraded it continues to process scaled tiles that have already been assigned to it, but will not be assigned new jobs by the SCM. If resources used by the current scaling jobs continue to grow then corruption of the scaled streams is possible. In Q1 2019 scalers were going degraded frequently due to log rotation management which has since been addressed by checking for log rotations more quickly and forcing rotations on a tighter schedule. Also in Q1 2019 a memory leak was discovered, this is being managed and is expected to be fixed in AV2.16. With those pieces of technical debt addressed earlier this year, scalers have been running smoothly day to day without issues.

When a Scaler Node does go down, the Scaler Manager makes sure tiles keep running by immediately assigning the scaling jobs of the failed Scaler Node to the other Scaler Nodes in the group. The affected tiles will not show updated output or may be black for a few seconds while the scaling jobs get reassigned. This behaviour creates a robust delivery system since any missing streams will automatically be reassigned in a matter of seconds and do not significantly affect the Spec-Guide experience in the majority of cases.

One piece of work will be needed in 2020, and that is that 42 of our scalers in SLDCMO are Cisco M3 blades which are going end of life which will need to be replaced.

## 2 Scaler Node Resource Analysis

The scalers (as well as any server we operate) should at a minimum be assessed on memory consumption, CPU use, disk space utilization, networking usage, and application performance.

### 2.0.1 Networking Usage

Since the scalers use a multicast to send scaled tiles the stitchers subscribe to the established stream, and this poses no concerning constraint on networking or the

number of connections to the scaler node. We will not be bound by networking capacity in the foreseeable future. Scalers have a 10Gb NIC per blade, but in future hardware acquisitions a 1Gb NIC would be sufficient and could present significant savings.

Stitchers don't create additional connections to the scaler nodes, but do, however, have to directly subscribe to the Scaler Cluster Manager (SCM), which will be addressed in an analysis of SCM Capacity.

### **2.0.2 CPU Utilization**

Scaler nodes have either 56 or 72 cores (depending on blade model) and are using 10% of that allocation. There is no constraint on the scaler nodes based on CPU usage unless we expand our content provided by a factor of 5-10.

### **2.0.3 Application Performance**

Scalers use one process which gets used to transcode the stream known as "transcoder\_core". This process is generally quite stable, with processes running on scalers for 3+ months without issue. The one known bug in the application is a memory leak causing one transcoder\_core process to grow over the course of a few hours until it uses all of the memory on the blade (250+GB!). This bug is expected to be addressed in AV2.16. Operationally until the bugfix is delivered we are automatically killing transcoder\_core processes that grow over 8GB in order to manage memory consumption.

### **2.0.4 Memory Consumption**

The average transcoder\_core process uses approx. 500MB of memory. With approx. 30 transcoder\_core process per scaler node we are using 16GB of memory of 256GB available. There are no foreseeable memory constraints on the scaler nodes.

## **3 Conclusion**

Technical debt has been addressed in Q1/Q2 2019, and with those addressed there are no foreseeable concerns over capacity through EOY 2020. End of life M3 blades will need to be replaced so we have continued hardware support from Cisco.

Stitchers support having a primary and secondary scaler cluster for geographical failover. The Stitchers implement failure detection and stream aliveness sensing to fall back to the secondary cluster in case the primary fails. Once the primary comes back up, Stitchers switch back to the primary cluster. This geographic redundancy is not currently configured, but might be recommended as dictated by future projects such as turning on scaled tiles in DOCSIS markets, the build out of a Charlotte SRDC, or QAM to DOCSIS market conversions.