x.264 Optimization Checklist

Practical tuning guidance for VOD and live H.264 encoding using x264 or commercial variants

Per-Title Encoding

- If you're not using per-title or content-adaptive encoding, start here
- Significant savings at no quality cost
- Works across ladders, improves all rungs

GOP Structure

Overall - H.264	.5 sec	1 sec	2 sec	3 sec	4 sec	5 sec	10 sec	20 sec
All Animation	90.25	92.75	93.90	94.33	94.48	94.59	94.81	94.90
All Entertainment	90.67	92.16	92.92	93.14	93.26	93.35	93.50	93.53
All Sports	91.11	93.94	95.23	95.56	95.78	95.88	96.05	96.11
All Office	82.61	91.21	94.21	94.85	95.07	95.26	95.43	95.53
Overall	**88.73**	**92.42**	**93.92**	**94.32**	**94.49**	**94.61**	**94.79**	**94.86**
Delta from Max	**6.13**	**2.43**	**0.94**	**0.54**	**0.37**	**0.25**	**0.07**	**0.00**

Table 1: Quality improves with longer GOPs, though it's diminishing returns after 5 seconds.

- Use longer GOPs: 5–10 seconds, depending on segment duration.
- Enable scene-change I-frame alignment (variable GOP) if supported
- Ensure compatibility with packager and players

Rate Control

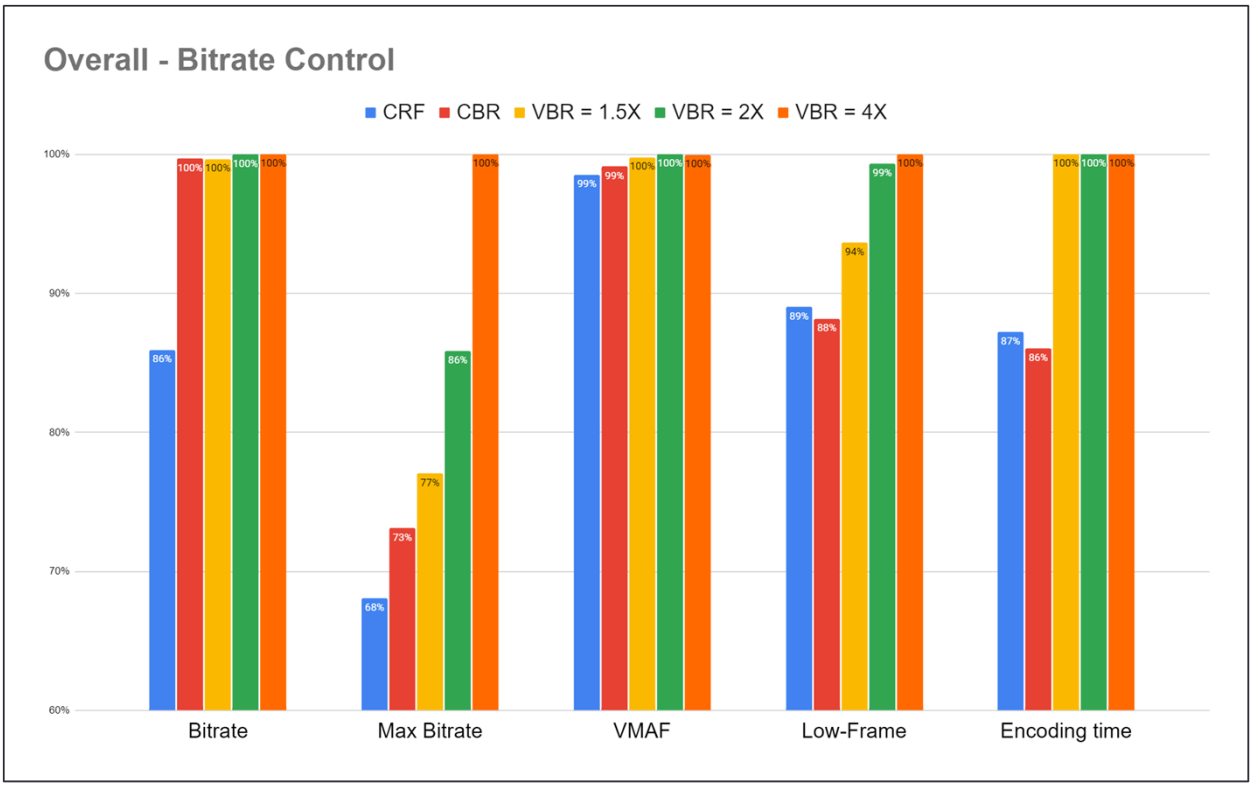

Figure 1: While 2-pass VBR yields only a ~14% increase in encoding time, the chart exaggerates that impact due to a compressed Y-axis (60–100%). Quality gains—especially in low-frame performance—are far more substantial.

- For VOD: Use 2-pass VBR with a 200% constraint;
- For constrained workflows: Use capped CRF (e.g., CRF 23–27 with VBV maxrate)
- Use CBR only where required (live or tightly managed bandwidth)

Presets

Preset	Bitrate	Encoding time
Ultrafast	196%	6%
Superfast	171%	11%
Veryfast	151%	16%
faster	123%	19%
fast	122%	26%
Medium	112%	31%
Slow	108%	43%
Slower	106%	56%
Veryslow	100%	100%
Placebo	100%	408%

Table 2: Presets trade encoding time for bitrate. Be sure to factor both into the preset selection decision.

- Low volume? Use medium or slow to reduce encoding time
- High volume? Use veryslow or placebo for bandwidth ROI
- Remember: bitrate controls quality, not preset

Threading

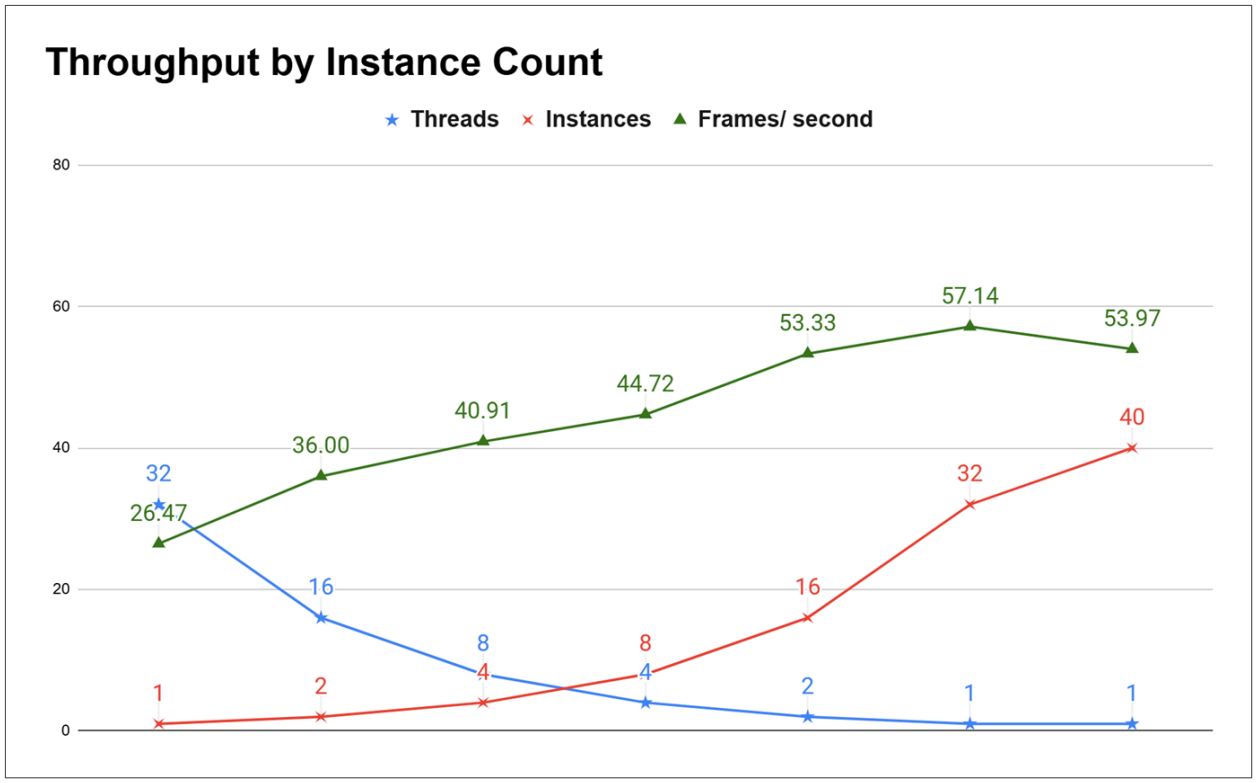

Figure 2. Single thread delivers highest quality (see handout) and highest throughput when encoding with multiple instances.

- Prefer 1 thread per job with parallel processing for quality
- For testing or prototyping: 8 threads offers good balance
- Avoid overloading threads beyond core count

Scaling and Filters

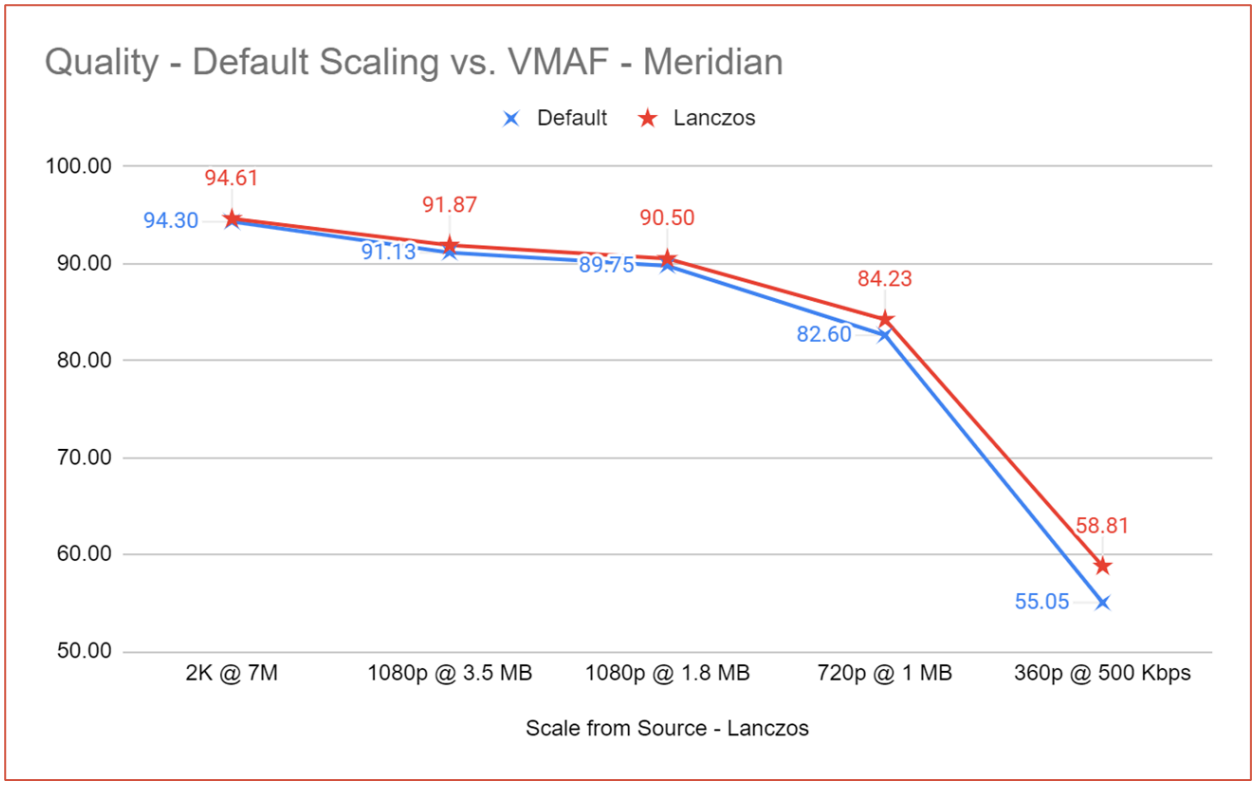

Figure 3. Lanczos delivers better quality in the lower rungs at no cost.

- Replace bilinear with Lanczos for downscaling:
  `-vf scale=640x360 -sws_flags lanczos`
- Especially useful for bottom rungs, improves VMAF up to 3–4 points

QA and Metrics

- Target VMAF: 93–95 for premium, 85–92 for general
- Track lowest frame score, harmonic mean, and standard deviation
- Use frame-level inspection to validate score spikes

Cloud/Infra Efficiency

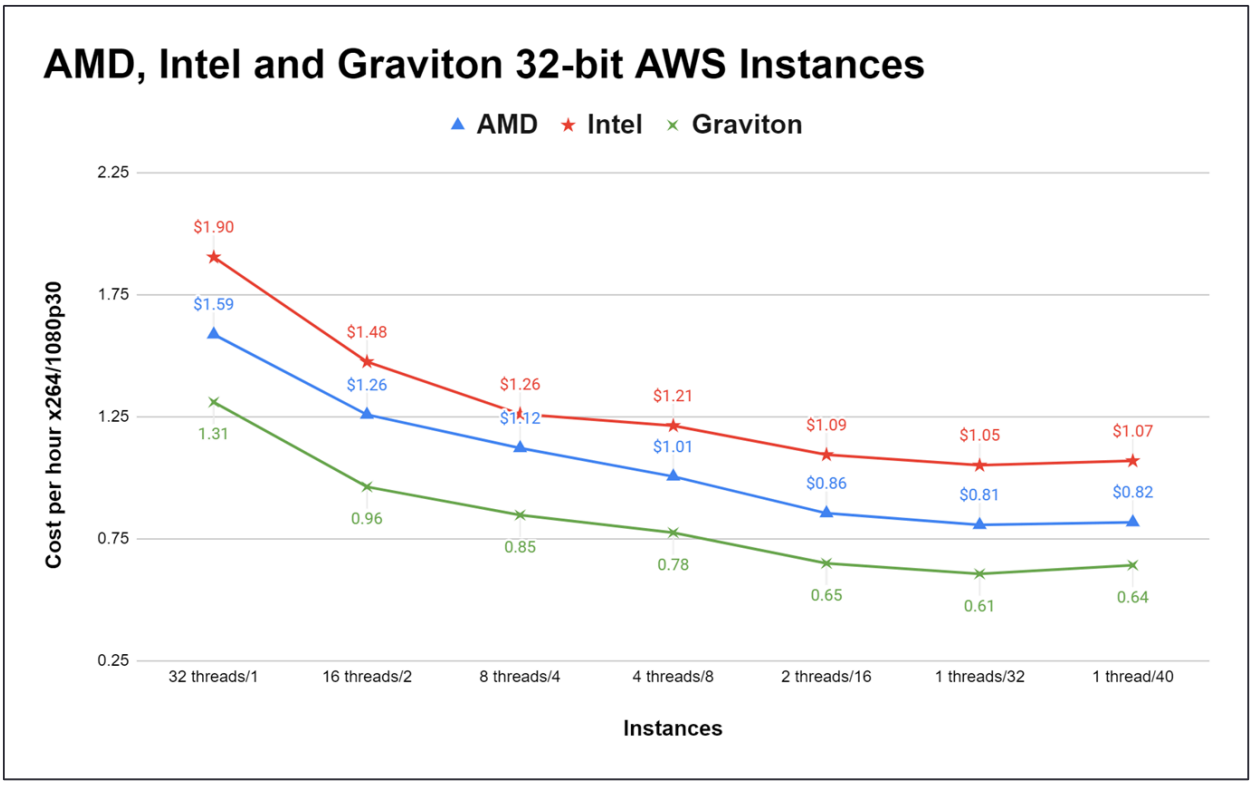

Figure 4. With x264 and FFmpeg, Graviton delivered the lowest cost per hour of encoding.

- Graviton: best $/hour for x264
- AMD: best throughput per instance
- Tune core/thread ratio to avoid quality drops or job failures

Ready to Optimize Your Encoding Workflow?

I consult to help teams improve their H.264 and HEVC encoding configurations. This can range from a quick review of existing settings to a more in-depth, testing-based optimization.

Not sure what kind of help you need? Start with a free 30-minute discovery call to review your current approach and identify the most valuable areas to tune.

Send a note to **jan.ozer@streaminglearningcenter.com** to get started.