Computer architecture @ Google

Alex Ramirez
2019/2/19
Google is a software company, right?

“To organize the world's information and make it universally accessible and useful.”

-- Google’s mission statement
Google needs a lot of hardware
Google Search capabilities

Search Timeline

+ instant search + live-translate + knowledge-graph + google now…
Video processing grows faster than compute performance

- In July 2015, 400 hours of video were uploaded to YouTube every minute
  - And video resolution is also increasing

- Upload rate grows faster than Moore’s Law
  - Even if we disregard Moore’s Law slowing down
“The datacenter is the computer”
... and now, a Google Ad ;)
The compulsory Moore’s Law slide

Single thread performance flattening

Power density flattening

Number of transistors flattening?!
Can we panic now?

Economics Is Important - The End of Moore’s Law
FORBES, JUL 26 2016

Intel Corp Officially Kills “Tick-Tock”
Bye, bye “Tick-Tock” and hello “Process-Architecture-Optimization”

Moore's Law Is Dead. Now What?
MIT TECHNOLOGY REVIEW, May 13 2016

End of Moore’s Law: It’s not just about physics
SCIENTIFIC AMERICAN

Moore’s law really is dead this time
ARS TECHNICA, 2/10/2016

Are the chips down for Moore’s Law?
26 Jul 2016
The compulsory Moore’s Law slide (II)

"It is difficult to make predictions, specially about the future"
--Mark Twain

* Moore's Law is actually about transistor density, not performance

Special purpose
more active transistors,
higher frequency

Heterogeneous
more active transistors,
higher frequency

Multicores
more active transistors, higher frequency

Single thread performance
more active transistors, higher frequency

Accelerators
Surviving Moore’s Law demise

Accelerators = special purpose hardware (fixed function?) that makes the critical workloads faster & cheaper …

… but they cost $$$, and not all applications have enough volume to afford one

Amdahl’s Law still applies …
90% speedup of 1% of the workload vs. 1% speedup of 90% of the workload
10x accelerators: Machine learning

“Google voice search queries in 2016 are up 35x over 2008” according to Google trends via Search Engine Watch

“40% of adults now use voice search once per day” according to Location World

“We estimate that 325.8 million people used voice control in the past month” according to Global Web Index (that’s almost 10% of the online population according to Internet Stats).

“65 percent of people who own an Amazon Echo or Google Home can’t imagine going back to the days before they had a smart speaker.” via Geomarketing.com

“47% expect their voice technology usage to increase” via ComScore

“72% of people who own a voice-activated speaker say their devices are often used as part of their daily routine.” via Think with Google
Further accelerator opportunities?

Google’s most expensive application uses <10% of the fleet
6 “datacenter tax” functions → 25% of fleet cycles; across applications
  ○ Bigger than the flagship applications
Video transcoding
**What is video transcoding?**

Uncompressed video is not manageable, it must **always** be compressed.

Decoding + Encoding = Transcoding  
Encoder + Decoder = CODEC

Lossy compression is much more efficient than lossless, at the same **perceptual quality**:

- **Human vision traits**
  - Luminosity vs. Chroma perception
  - Focus of attention
  - Perception of motion
  - ...

- **Spatial and temporal similarity** in the video

<table>
<thead>
<tr>
<th>CODEC</th>
<th>Settings</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncompressed</td>
<td></td>
<td>40 MB (320 Mb)</td>
</tr>
<tr>
<td>H.264</td>
<td>Lossless</td>
<td>1.9 MB (15.2 Mb)</td>
</tr>
<tr>
<td></td>
<td>CRF 18 Visually lossless</td>
<td>0.26 MB (2.1 Mb)</td>
</tr>
<tr>
<td></td>
<td>1.8 Mbps (2 bit/pixel/s)</td>
<td>0.22 MB (1.8 Mb)</td>
</tr>
<tr>
<td>VP9</td>
<td>0.9 Mbps (1 bit/pixel/s)</td>
<td>0.12 MB (0.9 Mb)</td>
</tr>
</tbody>
</table>

### Example:

- **1280x720 (HD), 30 frames, 1 second**
  - H.264: 1.9 MB (15.2 Mb), 1.8 Mbps (2 bit/pixel/s)
  - VP9: 0.26 MB (2.1 Mb), 0.9 Mbps (1 bit/pixel/s)

- **3840x2160 (UHD), 30 frames, 1 second**
  - H.264: 2.0 MB (16 Mb), 16.6 Mbps (2 bit/pixel/s)
  - VP9: 1.0 MB (8 Mb), 8.3 Mbps

---

4G mobile networks are limited to 20 Mb/s.  
Xfinity offers 15 Mb/s, 60 Mb/s, 150 Mb/s, 250 Mb/s and 400 Mb/s home internet speeds (in the SF Bay Area).  
2-4 Mbps is the quality (bitrate) used by video streaming applications for 720p contents; actual quality depends on both bitrate and encoding effort.
Chroma subsampled image
10x reduced chroma
(about ~1/3 of the original size)
Luma subsampled image
10x reduced luma
(about \(\frac{2}{3}\) of the original size)
Encoding modes

- **Low-latency (eg. interactive video)**
  - Real-time + low latency one-pass encoding, **no B frames**

- **Lagged (eg. buffered live streaming)**
  - Real-time + low-latency one-pass encoding, uses B frames

- **Offline**
  - Not real-time, can use **two-pass encoding** to distribute bits across the video
    - Use more bits for complex scenes, fewer bits for simpler scenes

![Graph showing video quality over time with one-pass and two-pass encoding](chart.png)
Transcoding metrics: bitrate (bit/s)

Bitrate (bit/s) = \frac{\text{frames}}{\text{sec}} \times \frac{\text{pixel}}{\text{frame}} \times \frac{\text{bit}}{\text{pixel}}
Transcoding metrics: video quality (eg. PSNR)

High quality (low distortion)

- PSNR 42dB
  - 4 bit/pixel/s

- PSNR 34dB
  - 0.5 bit/pixel/s

Low quality (high distortion)

- PSNR 31dB
  - 0.25 bit/pixel/s
The green transcoder offers better quality for the same number of bits, good! (or smaller video at the same perceptual quality, good!)
Bitrate vs. Quality vs. **Compute time** trade-off

But the green transcoder is 10x slower than the red transcoder ... bad!
The Zen of video transcoding

- Nuanced trade-off between
  - Speed
  - Quality
  - Size
- Choose one, hurt the other two
- Quality impacts the user’s Quality of Experience
  - QoE ~ watch time ~ revenue
- Size impacts storage and network costs
- Speed impacts the compute costs
YouTube
Not all videos are popular

A few top videos gather most of the watch time

Many videos are never watched

Not even by the person who uploaded it :)
(eg. Photos automatic uploads)
The YouTube video processing pipeline

Every video uploaded to YouTube is transcoded multiple times
Multi-output transcoding (MOT) to adapt to the client device capabilities
Popular videos amortize higher transcoding effort across higher watch time
vbench
# Previous video benchmarks and video sets

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Description</th>
<th>Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SPEC CPU 2006</strong></td>
<td>Reference H.264 decoder + 2 videos: 176x144, 300 frames 512x320, 171 frames</td>
<td>Focus on compiler + CPU improvements</td>
</tr>
<tr>
<td><strong>PARSEC</strong></td>
<td>libx264 decoder + 1 video at: 1080p, 512 frames / 30p, 3 frames / 18p, 1 frame</td>
<td>Focus on multithreading scalability</td>
</tr>
<tr>
<td><strong>Netflix</strong></td>
<td>72 videos (9 public) (1 resolution) 1080p, 120 frames</td>
<td>Focus on subjective quality measurements</td>
</tr>
<tr>
<td><strong>Xiph.org</strong></td>
<td>41 HD videos 480p, 720p, 1080p, 2160p (250-1200 frames)</td>
<td>Used as reference in AV1 development</td>
</tr>
<tr>
<td><strong>YouTube-8M</strong></td>
<td>7 million video URLs, CCby license 120-500s long, watched &gt;1000 times</td>
<td>Focus on video classification Labeled videos. Intended for training of neural networks</td>
</tr>
</tbody>
</table>

There is no **common language** to compare video transcoders.
Video set coverage (resolution vs complexity)

- **YouTube corpus** is 400+ videos, designed for high coverage, both in resolution and complexity
- **SPEC** and PARSEC (not shown) are outliers
- **Netflix** videos are clustered in one resolution, high complexity
- **Xiph.org** has better coverage, but is 40+ videos, and shows bias towards high complexity videos

But ... is coverage the right target?
   Yes, those videos exist, but ... are they relevant?

**We want representativeness, not just coverage**
Generating a representative video set

- Define the basic set of video characteristics
  - Characteristics that do not correlate with each other
  - Characteristics that correlate with transcoding results
- Gather statistics about YouTube time spent transcoding each video category
- Use K-means clustering to select a subset of categories
  - Selected using weighted multidimensional clustering
  - Centroid value is not relevant / representative
    - Use the mode value to represent the category
- Extract random CC videos that match the selected representatives
  - Enables public distribution of the video set
Selected video characteristics

- Resolution
- Framerate
- Complexity
  - Measured as bit/pixel/s at constant quality encoding

- Other video characteristics ...
  - I frame frequency
  - I frame size
  - I/P frame size
  - ...

Multidimensional clustering

- Accumulate video transcoding time for each video category
  - \{resolution, framerate, complexity\} -> weight
- Run k-means algorithm on the normalized + weighted data
  - Obtain the desired number of centroids
    - May get less than desired if clusters have irrelevant weights
  - Map data elements to centroids
  - Select most frequent resolution | framerate | complexity (mode) as cluster representative
- The initial set of cluster seeds is generated randomly
  - Two runs of the algorithm will (likely) generate two different sets of clusters
- Centroid values are not relevant / representative
  - Use the mode value to represent the category
Resulting video set
Transcoding scenarios

● **Upload**
  ○ Requires maximum quality, because the resulting video will be transcoded again
  ○ Intermediate video size is hardly relevant

● **Live**
  ○ Has to match a speed SLA, but faster than real-time is not a benefit
  ○ Quality of Experience depends on both bitrate and quality

● **VOD**
  ○ Aim at optimizing speed and video size
  ○ Can not degrade the QoE for the user

● **Popular**
  ○ Transcoding time is mostly irrelevant
  ○ Aim at optimizing quality and bitrate
Benchmark metrics

● We provide a set of reference values
  ○ Measured speed / size / quality for all scenarios on a known baseline platform (hw + sw)

● Benchmark results measure speedup/improvement vs. reference metrics
  ○ $S = \frac{S_{\text{new}}}{S_{\text{ref}}}$
  ○ $B = \frac{B_{\text{ref}}}{B_{\text{new}}}$ (smaller bitrate is better)
  ○ $Q = \frac{Q_{\text{new}}}{Q_{\text{ref}}}$

● Higher values are always better
## Benchmark scores

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Optimization target</th>
<th>Score formula</th>
<th>Constrains</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload</td>
<td>Optimize speed and quality at the expense of bitrate</td>
<td>( S \times Q )</td>
<td>( B &gt; 0.1 )</td>
</tr>
<tr>
<td>Live</td>
<td>Optimize bitrate and quality as long as it is done in real-time</td>
<td>( B \times Q )</td>
<td>( S_{\text{new}} ) is real-time</td>
</tr>
<tr>
<td>VOD</td>
<td>Optimize speed and bitrate, as long as it does not degrade quality</td>
<td>( S \times B )</td>
<td>( Q \geq 1^* )</td>
</tr>
<tr>
<td>Popular</td>
<td>Optimize bitrate and quality at the expense of speed</td>
<td>( B \times Q )</td>
<td>( B \geq 1 ) &amp; ( Q \geq 1^* ) &amp; ( S &gt; 0.1 )</td>
</tr>
<tr>
<td>Platform</td>
<td>Optimize hardware speed, no changes to software</td>
<td>( S )</td>
<td>( B = 1 ) &amp; ( Q = 1 )</td>
</tr>
</tbody>
</table>

* or \( Q_{\text{new}} \geq 45\text{dB} \) (upload quality)
Reporting results

- Report all 3 results for each scenario on every separate video
  - \{speed, bitrate, quality\} \times \{upload, live, vod, popular\} \times 15 videos

- Each video will be weighted different by different service providers
GPUs shine in Live transcoding

- GPUs are required to trade-off speed for video size in the VOD scenario
  - 5-20x faster ...
  - ... but much larger videos
- GPUs excel at live transcoding
  - Fast enough for real-time
  - Better quality
  - Smaller videos
- Adding video compression tools (quality and size) requires area and power
- There is room to trade-off speed for quality in hardware video transcoders
Want to work with us?
Internship program

- Students submit their application
  - Dates

- Phone interview(s)
  - 2 phone interviews
  - Focus on coding and problem solving

- Hosts submit their projects
  - Submission date / approval date

- Hosts select interns from the candidate pool

- Project matching interview(s)
  - Dates

- Offers are sent out

- Internship happens
Faculty research awards

- Unrestricted gifts as support for research at institutions around the world
  - Research Awards are designed to support **one year of work**, may renew for a second year

- How to apply
  - Read advice on [how to write a good proposal](https://ai.google/research/outreach/faculty-research-awards/) and learn more about our Faculty Research Awards in our [FAQ](https://ai.google/research/outreach/faculty-research-awards/)
  - Ask a Google employee to champion your proposal
    - Not strictly required … but helps make sure the proposal is relevant
  - Write your proposal using the advice mentioned in step 1. If you have a Google champion or sponsor, ask them to provide feedback
  - Applications in late summer through early October, decisions are announced in February

[https://ai.google/research/outreach/faculty-research-awards/](https://ai.google/research/outreach/faculty-research-awards/)
Jobs @ Google.com

- Select up to 3 job openings
  - google.com/careers
- Get an internal referral
- Submit your resume
- Resume screening
- Phone interview screening
  - 1 interview, focus on coding and problem solving
- On-site interview
  - 5 interviews: coding skills, design skills, domain specific knowledge & expertise
- Hiring committee
Backup / Extra
Frame types, Group Of Pictures (GOP)

A video is structured as a sequence of frames:

- **I-frame**: intra-coded frame, self-referenced
  - Actually a still image (like a jpeg)
  - Also called **key-frames**
- **P-frame**: predicted frame, encodes changes relative to a previous frame
  - Also called **delta-frames**
- **B-frame**: bi-predicted frame, encodes changes relative to previous or future frames
  - Requires additional buffering for decoding
  - Introduces lag when encoding

I-frames define the boundaries of a Group Of Pictures (GOP)
Macroblocks

Video frames are decomposed into **16x16 pixel groups**, called macroblocks.

Advanced codecs define a 64x64 “superblock”
- Recursively split into 32x32 superblocks, and then into 16x16 macroblocks
- Macroblocks also split into 8x8 and 4x4 subblocks

Macroblocks can also be of type I, P, B.
Motion search

A **motion vector** describes the movement of a macroblock from a reference frame to the target frame.

**Motion search** looks for the most similar macroblock in the reference frame(s)
- **Range**: how far from the original position
- **Method**: how to search the reference frame

The compressed video encodes the motion vectors + the **residual frame** (difference vs. reference frame)

Frame from the Big Buck Bunny movie, (c) copyright 2008, Blender Foundation / www.bigbuckbunny.org
Quantization + Entropy coding

Quantization: the lossy part of video encoding

- Convert blocks from spatial domain to frequency domain using DCT
- The matrix of coefficients is quantized
- The quantized matrix is divided by another matrix
  - More zeros -> lower entropy -> better compression

Entropy coding

- Multiple algorithms used
  - Run length encoding (RLE)
  - Huffman encoding
  - Arithmetic coding
- Context-adaptive binary arithmetic coding (CABAC)