Automating Adaptive Bitrate Production Workflows for Multiscreen Delivery

With the extension of video viewing from TVs to desktops, laptops, smart phones, and tablets, today's media landscape is radically different from just a few years ago. To keep up with each new platform, content owners and distributors have had to master an ever-growing array of formats and codecs. For high-volume enterprises such as broadcast and cable networks, the challenge has been not simply to expand the range of formats in which they can deliver, but to find an overall workflow that maintains quality while offering vastly greater efficiency than old-school, individually-supervised transfers from a master source to multiple destinations. Enabling such workflows and interfacing them with existing industry procedures to create a seamless process has involved significant industry-wide investment in recent years.

The fastest growth in media viewing has been on platforms utilizing adaptive bitrate streaming

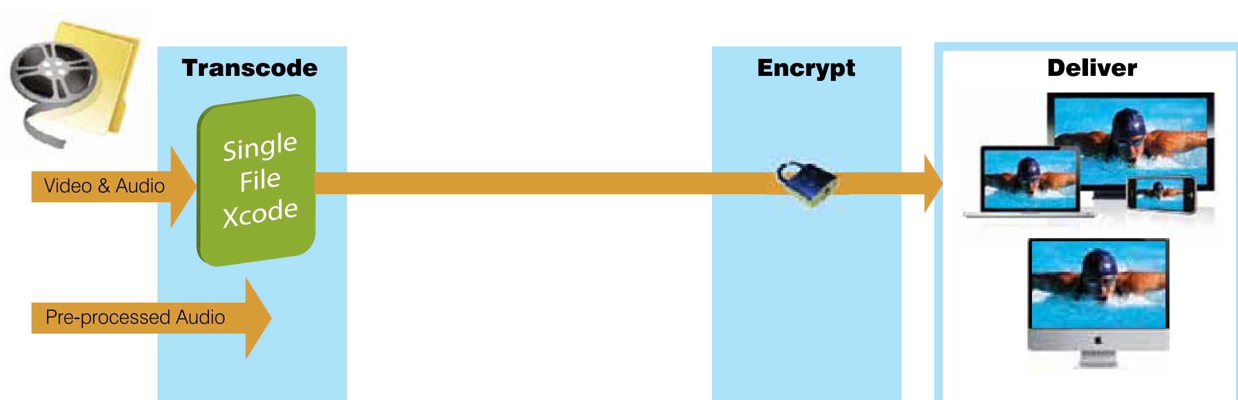
While development of media preparation and delivery systems has been focused on generating individual media files, the fastest growth in media viewing has been on platforms utilizing adaptive bitrate (ABR) streaming, which tailors streams to the resolution of the playback device and the available bandwidth of the connection. ABR streaming files are qualitatively different in significant ways from the files around which most content preparation and delivery systems were originally designed. The key question is whether modern prep and delivery systems can be adapted to handle these differences or if instead the industry should embrace entirely new and separate systems specifically designed for adaptive streaming.

An ABR file is actually a package of files rather than a single individual file

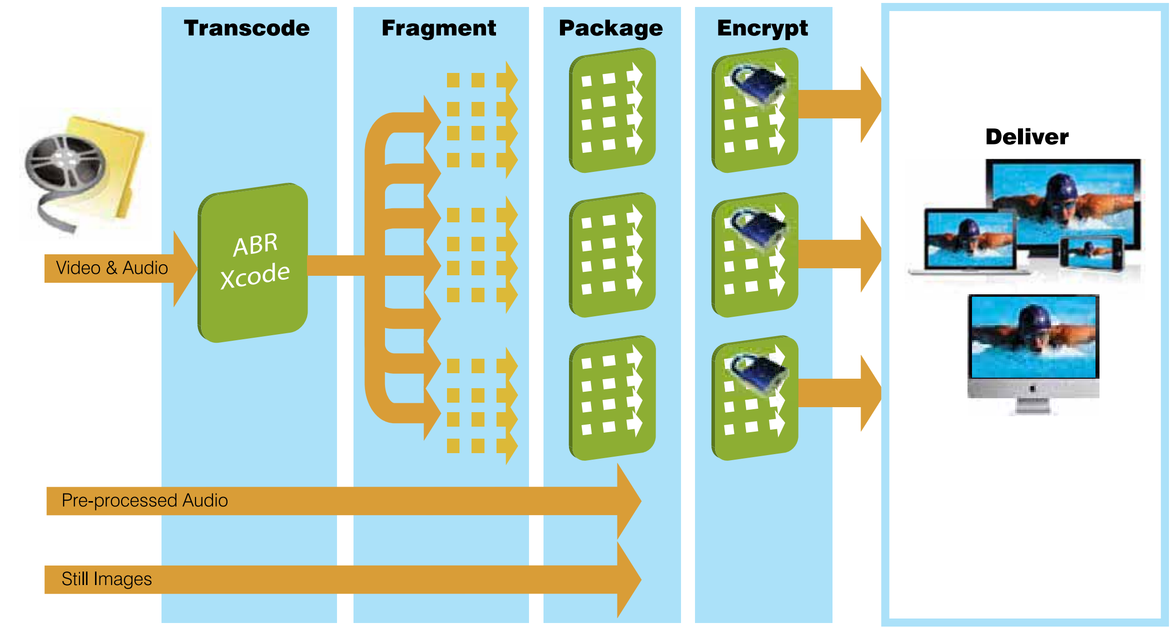
The core distinction between a standard media file and an ABR file (e.g. Adobe Dynamic Streaming, Apple HTTP Adaptive Streaming, Microsoft Smooth Streaming or MPEG-DASH) is that an ABR file is actually a package of files rather than a single individual file. An ABR package includes a manifest file, which holds the stream metadata, and set of multiple "layers," each made up of the media data for a different target bitrate. To enable switching between layers as conditions change during streaming, the content for each layer is fragmented into files of only a few seconds in duration.

The complications of dealing with this different file structure are apparent as soon as one contemplates handing off an ABR package to, for example, a content delivery network (CDN) that is designed to receive a single file for each item of media content. In a manual, step-by-step workflow, one could conceivably wrap the package's files into a .zip or .tar archive, FTP it to the CDN, and then rely on the CDN to properly extract the files and handle them such that they function as the intended ABR package. But when demand requires production and delivery of up to hundreds or thousands of files a week—the level at which effective process automation is a business necessity—the gross inefficiency of this approach becomes immediately obvious. In this context, any step that is not automated becomes a severe bottleneck.

Traditional Transcode Workflow



ABR Transcode Workflow



Components must be assembled into format-compliant packages for delivery to destinations

This same principle applies, as well, to many other aspects of media file preparation. The material for a given output clip is often drawn from multiple source clips (e.g. provider logo, main content, provider promo). The transcoded files must be QC'd. The components, both transcoded and externally provided, that make up the deliverable must be assembled into format-compliant packages for delivery to target destinations, in some cases with DRM or other encryption applied. Industry experience has long since proven that handling these tasks with discrete systems is far less efficient than handling them within a fully automated process that supports source-file playlists, tracks jobs, reports status, and handles the handoff of output materials to external systems. Given that these capabilities are already available in field-proven high-throughput systems for non-ABR content, there is no compelling reason to develop an entirely new framework for ABR processing.

Even if there were a separate ABR-specific system that addresses all of the requirements outlined above, it would be inherently less efficient to utilize a separate process. One lesson learned as file formats have proliferated over the last few years is that the most efficient use of resources is to access source materials just once, transcoding simultaneously into all of the different required variants (progressive download, TV/VOD distribution, etc.) and then assembling the results into a compliant package for each destination. It makes far greater sense to take advantage of this efficiency for ABR than to develop, test, and deploy separate systems.

Transcoding capacity must be extended with parallel processing hardware

So what does it take to fully support ABR streaming in existing video transcoding workflows? Transcoding capacity must be extended with the introduction of purpose-built parallel processing hardware, thereby maximizing throughput per unit of computational, thermal, power, and space resources. And processes that were designed around individual files—QC, job tracking, output packaging, handoff of deliverables—must be adapted so that the definition of an individual "job" encompasses a related collection of files.

Telestream Vantage Transcode HE Server streamlines ABR processing

With the introduction of Vantage Transcode HE Server, Telestream has already addressed these requirements, building on the industry's investment in proven video transcoding and workflow automation systems while adding high-volume, high-efficiency ABR processing capabilities.

Telestream

For additional information, contact Telestream at 530- 470-1300 or visit www.telestream.com.