WHITE PAPER

Service Providers
Telecommunications

Virtualizing Voice Transcoding Applications

Intel® Packet Voice SDK, as used in Nokia's Session Border Controller*, enables voice transcoding applications to be virtualized with densities comparable to digital signal processors

Executive Summary

Intel® Packet Voice SDK (Intel® PVSDK) enables voice transcoding applications to be virtualized, in line with network functions virtualization (NFV) strategies that are sweeping through the communications industry.

In recent years, voice codecs have increased in complexity as the industry drives to offer higher quality. Dedicated digital signal processors (DSPs) have previously been used to carry out media transcoding because of the high level of performance required.

General-purpose CPUs from Intel can be used to host virtualized media transcoding applications, and Intel PVSDK helps to achieve densities comparable to special-purpose processors. Intel PVSDK provides a suite of voice codecs that are optimized to take advantage of the parallel processing capabilities of Intel® processors, including Intel® Advanced Vector Extensions (Intel® AVX).

The Nokia Session Border Controller* (SBC) offers a virtualized platform for controlling and securing media and signaling traffic that crosses the edge of the communications network. The solution incorporates Intel PVSDK for its voice transcoding and takes advantage of the virtualization capabilities of Intel processors. Organizations using the Nokia SBC can right-size the resources allocated to the media transcoding function and benefit from lower-cost hardware and the optimizations built into Intel PVSDK.

Business Challenge

In recent years, many communications service providers have been replacing dedicated network appliances with virtualized applications running on industry-standard, commercial off-the-shelf (COTS) servers. This NFV strategy provides greater flexibility and scalability to network build-out. For example, demand typically spikes on New Year's Eve when people send messages to their friends or phone their families. This demand can be met by temporarily reallocating resources from lower priority applications, such as billing. Previously, the network would need to be provisioned for peak demand at every point, an approach that was becoming unsustainable because of rising demand. Using NFV, the network can draw on a pool of shared resources to provision the right applications at the right time.

Table of Contents

	GSM/GERAN EFR	AMR-NB	AMR-WB	AMR-WB+	Opus	EVS
Year standardized	1996	1999	2001	2004	2012	2014
Audio bandwidth	Narrowband	Narrowband	Wideband	Fullband	Fullband	Fullband Super wideband Wideband Narrowband
Usage	Used in the GSM system.	Default codec for voice in 3G and beyond (WCDMA, LTE)	Codec for HD voice. Default codec for wideband voice in 3G and beyond (WCDMA, LTE).	Recommended codec for generic audio in 3G and beyond (WCDMA, LTE)	WebRTC and OTT	Evolved HD voice codec for LTE
Bitrate(s)	12.2 kb/s	4.75 - 12.2 kb/s	6.6 - 23.85 kb/s	6 - 48 kb/s	6 - 510 kb/s	5.9 - 128 kb/s

Table 1: Evolution of codecs, showing how audio bandwidth and complexity has increased with successive protocols

Some applications have previously been difficult to virtualize because of the high performance they require. When voice calls cross network domains or require different codecs to be used in the same domain, voice transcoding is required. Traditionally, embedded digital signal processors (DSPs) have been used to carry out this intense workload.

However, dedicated DSPs cannot easily be shared between different devices that require transcoding, such as different SBCs. Additionally, the DSP resource may be overprovisioned most of the time. In a large corporate network, for example, the majority of the calls might take place in the network, and not require transcoding, but the SBCs would all need to include DSPs.

Voice transcoding involves several tasks, including call control, packet processing, jitter buffer management, voice decode, sample rate conversion, tone processing and voice encode/decode. The most demanding signal processing tasks are voice encode and decode functions. Optimizing codecs for real-time performance can be a time-consuming task when using DSPs because it requires low-level programming of time-critical functions in assembly language or using intrinsic functions. DSPs also require specialized knowledge and software tools to program.

Call control and operations and management functions are traditionally implemented on Intel processors. Packet processing is widely implemented using the Data Plane Development Kit (DPDK), which is optimized for Intel processors.

For organizations that are creating and managing NFV applications based on industry-standard hardware, there is an added cost and complexity in maintaining separate skill sets for the DSPs and the rest of the applications. Intel processors enable organizations to implement all the transcoding functions on general-purpose CPUs using a single environment and set of tools.

There is also a trend toward using higher bandwidth and higher complexity codecs to improve call quality, error resilience and compression. Table 1 shows the evolution of mobile voice and audio codecs.

High-definition voice (HD voice) is being deployed worldwide and has an audio bandwidth that is more than twice that of traditional telephony, making it easier to communicate clearly, even in noisy locations. The EVS and AMR-WB+ codecs cover the full range of human hearing, with an audio bandwidth up to 20,000 Hz. However, this quality comes at a price—the increased complexity of transcoding the audio, with greater amounts of audio data and higher complexity algorithms to process it in real time.

The EVS codec serves as one example of the increased computational complexity of audio transcoding used in more recent codecs. EVS has been designed to support not only speech, but also music, and can provide intelligible speech with frame loss of more than 10 percent by generating signals to replace the lost frames.

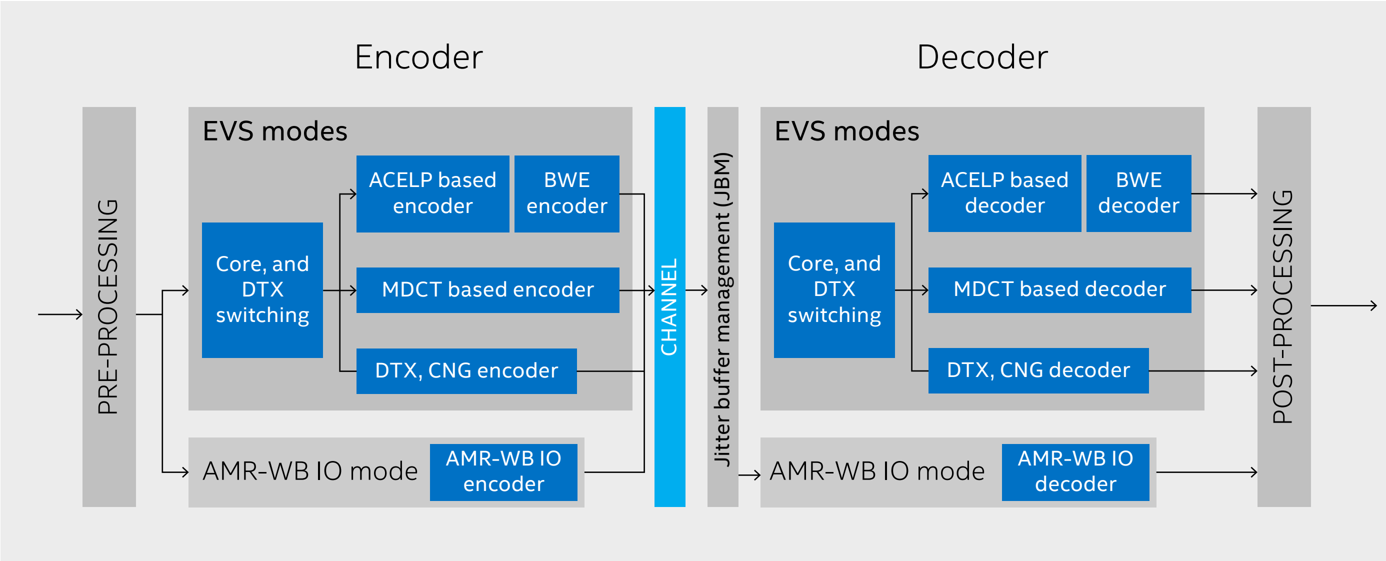

Figure 1: EVS codec uses the most suitable algorithm depending on content

Figure 1 shows the content-dependent approach that EVS uses to provide good quality for speech and music signals. The pre-processing unit classifies the signal type, which is then used to apply the most appropriate coding method.

The Algebraic Code-Excited Linear Prediction (ACELP) method, carried over from the AMR-WB standard, is used for speech. The section of bandwidth that ACELP does not cover is encoded using a time-domain bandwidth extension (BWE) approach. Music is encoded using modified discrete cosine transform (MDCT). Battery life can be optimized in discontinuous transmission (DTX) by using comfort noise generation (CNG) to create synthetic background noise instead of encoding and transmitting real background noise. The EVS codec also has backwards compatibility with the AMR-WB codec using an interoperable (IO) mode.

Solution Overview

With improvements to Intel AVX in Intel® Xeon® processors, supporting signal processing vector instructions, COTS servers are now providing an alternative to DSPs for voice transcoding, including for some of the most demanding codecs such as EVS.

Intel PVSDK helps to meet the growing interest in virtualizing audio transcoding functions. Building on sample voice codecs that were previously included in Intel® Integrated Performance Primitives (Intel® IPP) libraries, Intel PVSDK provides optimized codecs to enable voice workloads running on Intel processors.

Enhancements in the performance of Intel® C++ Compilers and vectorization tools developed by Intel help to speed up the optimization process, compared to working with DSPs.

By integrating Intel PVSDK with open-source components optimized for Intel® architecture, such as the DPDK and OpenStack, it is possible to create a high-performance NFV voice transcoding solution quickly.

Solution Architecture

Figure 2 shows the Intel PVSDK software stack. The Intel PVSDK codecs have been hardened for high-density production environments and are supported by a customized IPPSC (Intel® Performance Primitive speech codec) library, which includes computationally intensive functions and is provided as a binary. The high-level state machines for codecs are provided as reference source code, so they can be studied and modified as required. The solution uses the existing Unified Speech Component (USC) APIs from Intel.

DPDK is a set of open-source libraries and drivers for fast packet processing. Other ipp libraries are provided along with IPPSC for the complete codec implementation (see Figure 3). The IPPSC library is used for signal processing, and ippcore is used for core functions.

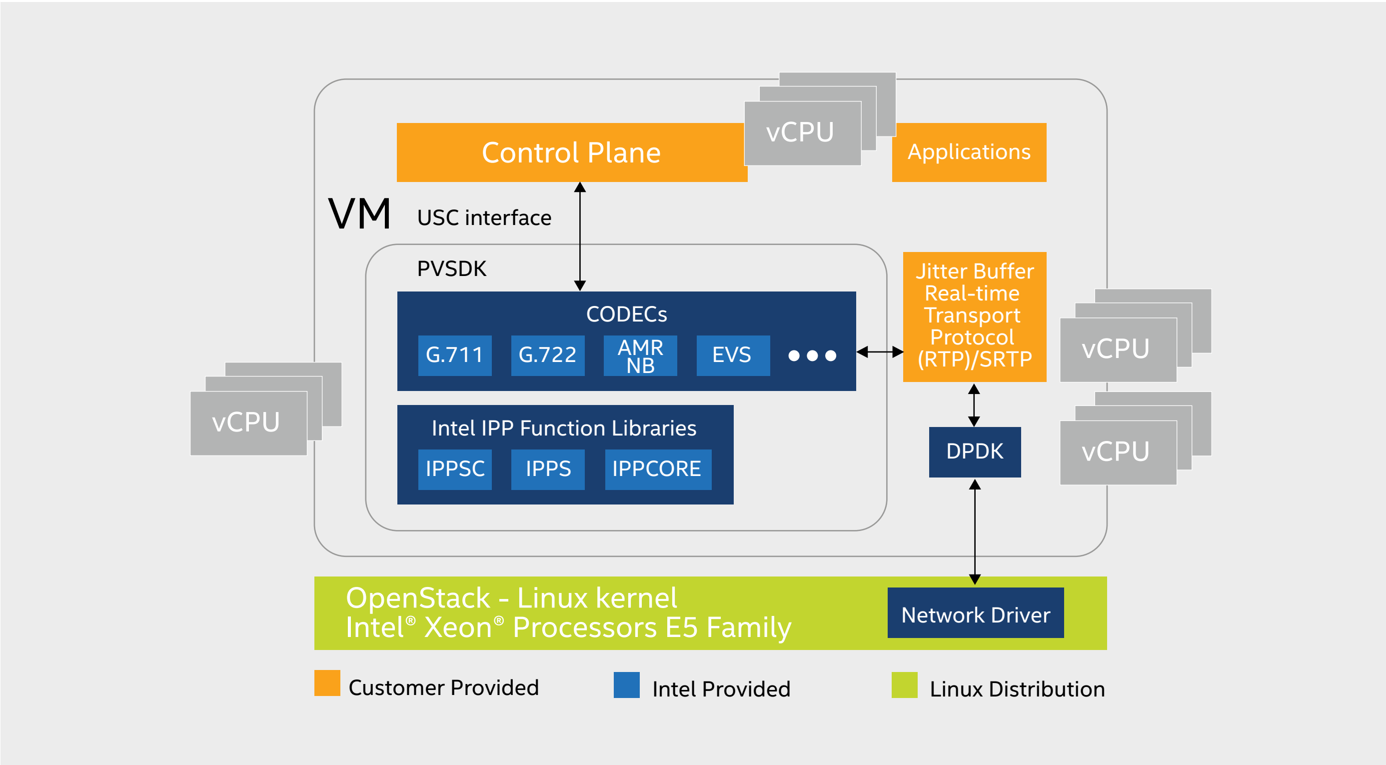

Figure 2: The Intel® Packet Voice SDK software stack

Intel AVX can accelerate complex algorithms, including voice transcoding, using SIMD (single instruction, multiple data) instructions. These instructions enable several data items to be processed at the same time with a single instruction, for highly parallel voice transcoding.

The Intel® Xeon® E5-2600 v4 processor offers Intel® Advanced Vector Extensions 2 (Intel® AVX2), which has 256-bit integer instructions. It adds support for floating-point, fused, multiply-add instructions as well as gather operations. The new Intel® Xeon® Scalable platform introduced Intel® Advanced Vector Extensions 512 (Intel® AVX-512), enabling ultra-wide 512-bit vector operations. Applications can pack 32 double-precision or 64 single-precision floating point operations per clock cycle in the 512-bit vectors, or eight 64-bit and sixteen 32-bit integers.

The voice codecs in Intel PVSDK have been optimized to take advantage of SIMD instructions, with restructured code and data buffers. Some crucial functions and loops have been implemented in assembly code and intrinsic functions using Intel AVX instructions.

Intel® VTune™ Amplifier and the profile option in Intel C++ Compilers were used to identify hot spots in the speech codecs, to focus optimization efforts. Using the profile-guided-optimization procedure over thousands of test vectors and multiple Intel architectures, the Intel team has created a library that offers stable and consistent performance.

Virtualizing the SBC

An SBC secures and controls the signaling and data streams as they move into and out of an internet protocol (IP) network. One of its functions is media transcoding: converting the audio signal to ensure that it can be used by both endpoints. The specific codecs used are negotiated as the call is set up, depending on the codecs that the endpoints can accept.

The Nokia SBC (see Figure 3) can be deployed as:

- A physical appliance, in the form of virtualized software running on HPE ProLiant* Rack Mount Servers, powered by Intel Xeon processors. This option is often chosen by customers that want to replace a purpose-built hardware SBC with an appliance;
- A fully virtualized cloud-based software-only SBC, for deployment on VMware* or OpenStack. This option gives the customer flexibility in the hardware resources used for deployment; or
- A hybrid solution comprising both approaches.

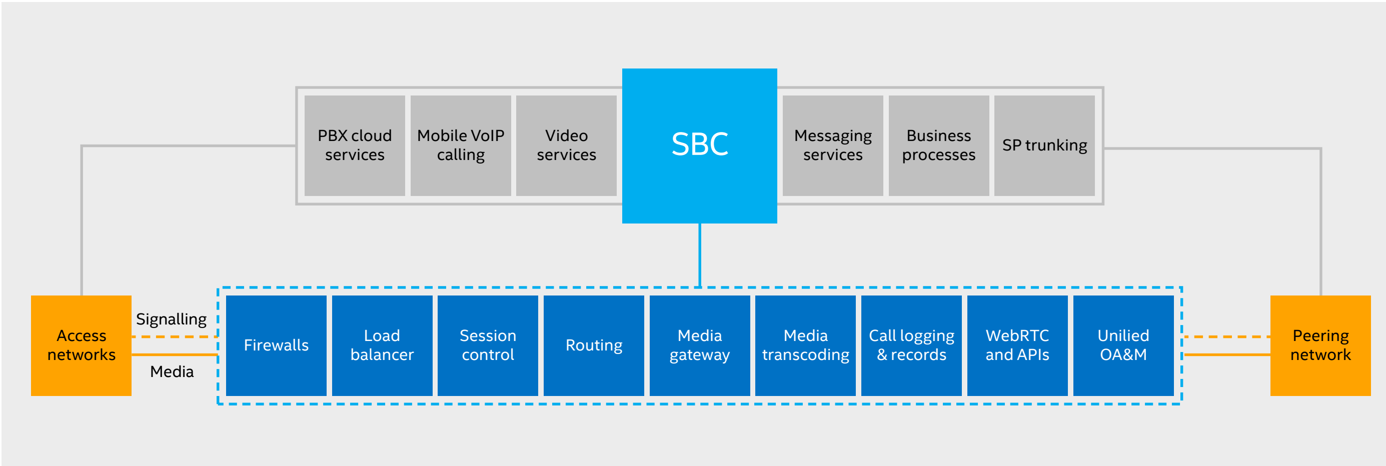

Figure 3: Nokia Session Border Controller* (SBC)

The media transcoding component of the virtualized SBC software makes use of the Intel PVSDK, and Nokia runs it using Intel Xeon processors. The optimizations in Intel PVSDK enable the Nokia SBC to take advantage of the 256-bit register for vectorization. Nokia has plans to upgrade to the Intel Xeon Scalable platform, which will enable the virtualized SBC to achieve higher densities required for their products.

Whereas a dedicated physical appliance would previously pass transcoding off to a DSP farm based on specialized hardware, the Nokia SBC solutions use a software implementation of those resources that runs on general-purpose CPUs. The transcoding can be performed by software based on Intel PVSDK, and the processed audio is then returned to the mainline SBC media processing software.

Because the functions are virtualized, there is greater flexibility in how they are resourced and managed. For example, it's possible to right-size the hardware allocated to the media transcoding function. The transcoding function no longer needs to be embedded in all the SBCs: some SBCs with transcoding functions can be positioned in the middle of the network to enable the transcoding functions to be shared by multiple SBCs.

Potentially, the number of media transcoding instances can be reduced from the number of dedicated DSPs that would be required in an all-hardware implementation, depending on the type of media traffic the network conveys. A company-internal network, for example, might need fewer transcoding instances if most communications happen within the network using similar devices.

Using Intel AVX architecture in the Intel Xeon processor and the Intel PVSDK enables Nokia to achieve the voice transcoding density required for its products, including for more complex codecs such as EVS.

Using a virtualized media transcoder has the potential to reduce costs through:

- The use of industry-standard COTS servers;
- A lower consumption of data center resources due to the avoidance of the need to provision specialized hardware dedicated to a specific function; and
- A more streamlined management and optimization process.

Nokia also benefits from the broader Intel ecosystem, with easier access to expertise and lower-cost availability of tools. Intel PVSDK enables Nokia to benefit from Intel's work on optimizing voice codecs without Nokia needing to invest resources to optimize a generic, open-source codec.

As the codecs, including EVS, continue to evolve, the use of industry-standard COTS servers provides a smoother upgrade path for Nokia.

Conclusion

Intel PVSDK provides voice codecs for virtualized transcoding that are designed to take advantage of the processor resources in Intel architecture, including Intel AVX SIMD instructions.

Using Intel PVSDK, it is possible to virtualize voice transcoding applications, for example in virtualized SBCs. Virtualized solutions, such as the Nokia SBC, can benefit from right-sizing the hardware allocated to media transcoding and from the ability to share media transcoding instances between SBCs.

Find the solution that is right for your organization. Contact your Intel representative or visit **intel.com/communications**

Learn More

- Nokia Session Border Controller*
- Intel® Xeon® Scalable platform
- Intel® Advanced Vector Extensions (Intel® AVX)
- Data Plane Development Kit (DPDK)
- HPE ProLiant* rack servers

Solution Provided By:

0718//SB/CAT/PDF 337891-001EN