

HIGH PERFORMANCE CLUSTER COMPUTING AS A TOOL FOR 4G WIRELESS SYSTEM DEVELOPMENT

Contributors

Lars Thiele

Fraunhofer Heinrich Hertz Institute

Thomas Wirth

Fraunhofer Heinrich Hertz Institute

Michael Olbrich

Fraunhofer Heinrich Hertz Institute

Thomas Schierl

Fraunhofer Heinrich Hertz Institute

Thomas Haustein

Fraunhofer Heinrich Hertz Institute

Valerio Frascolla

Intel Mobile Communications

“Future digital processing platforms need to comply with scalability, expandability,…”

This article discusses the importance, benefits, and potential of electronic design automation (EDA) tools for the pre-development of features in the context of current and future wireless communication and other digital signal processing systems. The strengths, weaknesses, and overall added values will be assessed, looking forward to a more structured potential adoption into standardization bodies. Critical systems features of EDA systems such as scalability, expandability, flexibility, and ubiquitous application space are addressed in detail and illustrated with specific application examples.

Introduction

The full adoption of the Internet in every aspect of modern society creates huge opportunities for companies, institutions, and the public in general. The digital age we live in is characterized by a variety of digital representations of data, which can be sensor data from temperature sensors to web cams, documents, accounting data, and everything else that can be digitized. Furthermore, availability of such an enormous amount of data requires fast and reliable communication between places of origin and places where the data is processed in order to convert it into information valuable for enterprises, consumers, or processes. Such reliable communication links are the backbone of our industries and everyday lives and are facing a steady increase of data rates to access and transfer locally and/or over long distances.

With all systems for communication between systems and subsystems and systems for data acquisitions, processing, and analysis, we observe a steady rise in complexity that requires digital processing platforms with the following characteristics as an answer: scalability, expandability, flexibility, and ubiquitous application space.

The growing complexity of such systems arises from specific complex digital data or signal processing mechanisms inside. This can be the complexity of a signal processing algorithm in itself, meaning the number of multiplications and additions and memory accesses might be very high, or, as with many numerical approximation algorithms of iterative nature, the procedure is repeated again and again until a predefined stop criterion is reached. Another common increase in complexity comes from the amount of data to be processed either in parallel or in serial, one data block after another. Here, it is necessary to have the ability to partition the data analysis and processing such that as many pieces of data or algorithmic code can be accessed and processed in parallel. In many cases the data sets and processing chains have interrelated

inputs and outputs, meaning input and output data are a function of each other—another source of increased complexity.

In order to keep the overall processing time within a reasonable range, for instance for system evaluations of large and complex communication systems or in case of real-time requirements within affordable time constraints of the application, it is of utmost importance to keep the signal processing capacity and memory management scalable, flexible, and expendable in order to increase parallelization of computation tasks appropriately to the application at hand.

Another important aspect comes from the fact that in most cases next-generation communication devices require a well-balanced tradeoff between capital expenditure (CAPEX) and operational expenses (OPEX) platforms tailored to specific digital signal processing (DSP) tasks. This again requires a significant scalability in units sold within a certain timeframe such as user equipment. OPEX here is not to be understood just in terms of cost to operate the devices; often aspects of energy consumption will make a decisive difference in determining whether a device will be considered suitable for daily use, such as, for example, sufficient battery life for the application. Furthermore, most of the modules or subsystems should be the basis for further use in future systems or derivatives in the evolution of existing standards for cellular communication systems like 3GPP LTE and LTE-Advanced.

On the infrastructure side, the lifecycle of equipment is very much different due to high CAPEX during initial rollout of a new infrastructure and the constant need of functional updates for communication infrastructure optimization. Therefore, especially on the infrastructure side, software-defined-radio platforms are state of the art today. Nevertheless, the use of commercial off-the-shelf (COTS) hardware usually used for PCs or for data centers would make it possible to take advantage of the market scale and therefore reduce CAPEX and allow potential sharing of hardware improvement cycles and the resulting upgrade path for more signal processing speed and capabilities. On the other side, the real-time requirements for involved signal processing demand the design of new modular signal processing architectures to balance adaptively between universal computation capabilities (flexibility aspect) and specialized signal processing functional support by hardware accelerators (task-dependent high-performance aspect).

This article is structured in the following way: the introductory section sets the scene with a general introduction of requirements for increasing signal processing capabilities. The section “An HPC Approach for SMEs and Research Labs” introduces the use of high-performance computing needs of small- and medium-sized enterprises (SMEs) and research labs that are not sufficiently supplied with PCs and simply can’t afford to use big data centers’ computing capabilities. We introduce the architecture and the modular design that allows the tailoring of high-performance computing (HPC) to customer needs. We show performance benchmarks with common applications used in the scientific community.

“... the use of commercial off-the-shelf (COTS) hardware usually used for PCs or for data centers would make it possible to take advantage of the market scale and therefore reduce CAPEX...”

The section “High-Performance Computing Application Examples” illustrates the performance potential with some meaningful use cases from various application fields, including cell planning for large wireless deployments and wireless system performance analysis, and real-time wireless signal processing in the cloud RAN architecture context, which became very popular with mobile network operators just recently in order to handle the rising complexity in their networks in an adaptive and scalable manner.

The concluding section provides an outlook on relevant research issues to be addressed in future work.

An HPC Approach for SMEs and Research Labs

This section deals with the motivation and a prototypical setup for an expandable and scalable HPC architecture. Finally this section summarizes performance benchmarks using extensive mathematic computations.

Motivation

In the past the top reason not to deploy HPC at the enterprise level turned out to be both the hardware and application development costs.^[1] On the other hand, Joseph et al.^[1] highlight the fact that HPC solutions will play an important role that can dramatically gain importance at R&D labs.

Therefore, it is of high interest to develop a scalable and expandable HPC architecture that can easily be extended once the computation complexity increases. In essence, buying an HPC system that is just big enough to handle today’s processing tasks saves a lot of CAPEX now and OPEX over the years.

Other constraints such as privacy and know-how protection issues will endorse R&D labs to operate self-owned and self-maintained clusters on premises.

High-Performance Computing Architecture

The suggested HPC architecture is a collection of processing hardware such as standard x86 central processing units (CPU) or other highly specialized DSPs (see Figure 1). For some applications it may be worth spending the effort to integrate advanced graphical processing units (GPUs) or Intel’s MIC^[12] (many integrated cores) boards into such a computing architecture. The decision highly depends on the processing tasks case by case, and the overhead in transferring the data from the host CPU to the acceleration board has to be considered in detail.

As a second important building block, the proposed architecture integrates a centralized storage system. The simplest solution is a network-attached storage device (NAS). However, in many HPC applications with a massive amount of data exchange, simple NAS architectures do not provide sufficient data transfer bandwidth for high volume data and a large amount of data file access. At this stage, we propose to add a cluster file system that can be distributed among several storage nodes.

“...Joseph et al. highlight the fact that HPC solutions will play an important role that can dramatically gain importance at R&D labs.”

“...constraints such as privacy and know-how protection issues will endorse R&D labs to operate self-owned and self-maintained clusters on premises.”

“...simple NAS architectures do not provide sufficient data transfer bandwidth for high volume data and a large amount of data file access.”

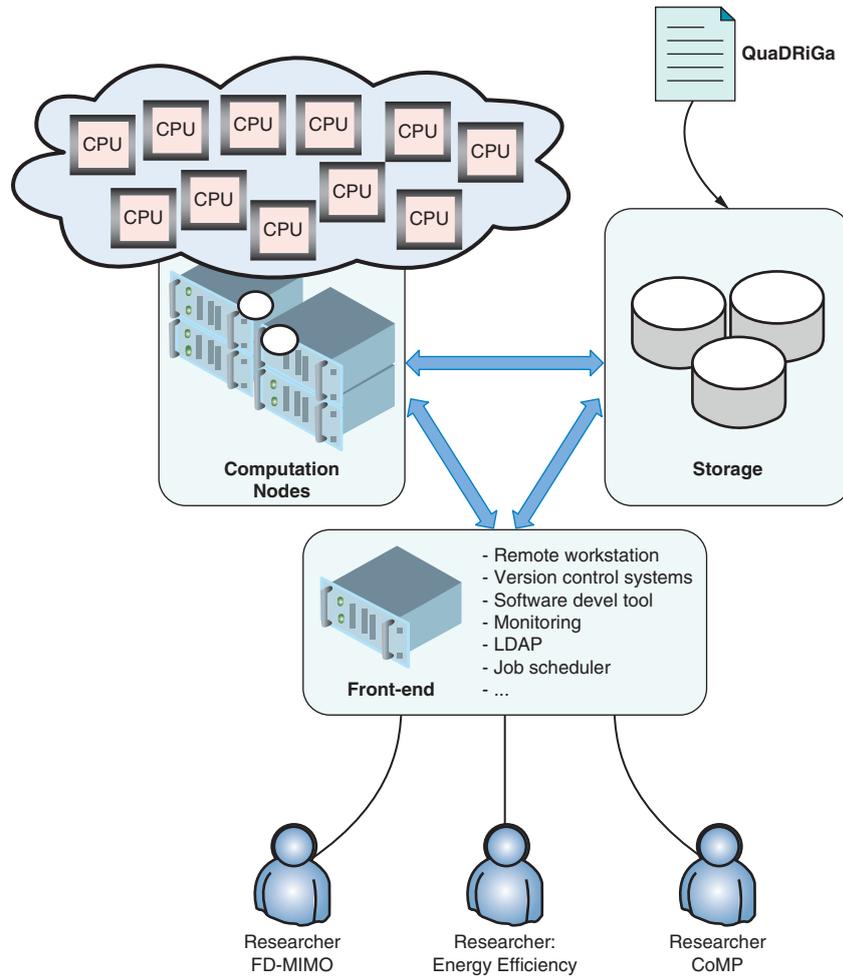


Figure 1: An expandable HPC architecture for R&D labs and SMEs.
(Source: Intel Corporation, 2014)

Performance Benchmarking

For initial testing purposes, our HPC environment consists of 12 computing nodes, 2 nodes for centralized, shared storage of user data and a set of 15 virtualized servers. The parameters are summarized in Table 1. For high level performance benchmarking we use the standard Linpack package^[3] for solving various systems of linear equations. In order to optimize the basic mathematic computations we utilize the OpenBLAS (Basic Linear Algebra Subprograms)^[4], which can be tuned to the available system architecture. The chosen approach results in the following observations and benchmarks:

- At moderate efficiencies (around 60 percent): For the *Intel® Xeon® (X5690)*^[13] processor, based on an enhanced *Nehalem* architecture called *Westmere-EP*^[14], $NB=216$, and *hyperthreading enabled* we observe a computation performance increase that is almost linear to the number of involved nodes (loss in efficiency is less than 3 percent).

“...we observe a computation performance increase that is almost linear to the number of involved nodes...”

“Almost 90 percent efficiency is reached without hyperthreading and with thread pinning to the cores.”

“...in applications that are not highly optimized and that also involve massive file I/Os and high volume data exchange, it may be beneficial to use...”

Computing nodes	12 nodes each with
CPUs	2 Intel® Xeon® X5690 3.47 GHz processors
cores	2 6
HT cores	2 12
Memory	96 GB
Interconnect	QDR InfiniBand*
OS	RHEL 6.4
Compiler	gcc 4.4.6
Math library	OpenBLAS 0.2.8
MPI	OpenMPI 1.6.4
Linpack	HPL 2.1
Rpeak	166.56 Gflops
Storage nodes	Distributed file system over 2 nodes each with
CPUs	2 Intel Xeon X5620 2.4 GHz processors
cores	2 6
HT cores	2 6
Memory	16 GB
Interconnect	QDR InfiniBand
OS	RHEL 6.4
HDDs	SATA Raid 6, 26 TB,

Table 1: High performance cluster computing configuration (Source: Intel Corporation, 2014)

- At high efficiencies: The loss in weighted peak computing performance is less than 13 percent.
- Almost 90 percent efficiency is reached without hyperthreading and with thread pinning to the cores.
- The *Sandy Bridge (E5-2680)*^[15] architecture doubles the overall computing performance of a *Westmere-EP (X5690)* architecture.
- In practice, we have to carefully select the right configuration for block size NB and the use of hyperthreading. For example, in applications that are not highly optimized and that also involve massive file I/Os and high volume data exchange, it may be beneficial to use these options, since time durations of CPU idles can be used by other processes.

These observations allow us to clearly argue that the suggested architecture is scalable in performance.

High-Performance Computing Application Examples

Within the following section we highlight the different applications for using HPC architecture in 4G wireless system development. It ranges from radio propagation modeling, large-scale radio network system-level analysis to real-time signal processing such as cloud-RAN or video re-encoding.

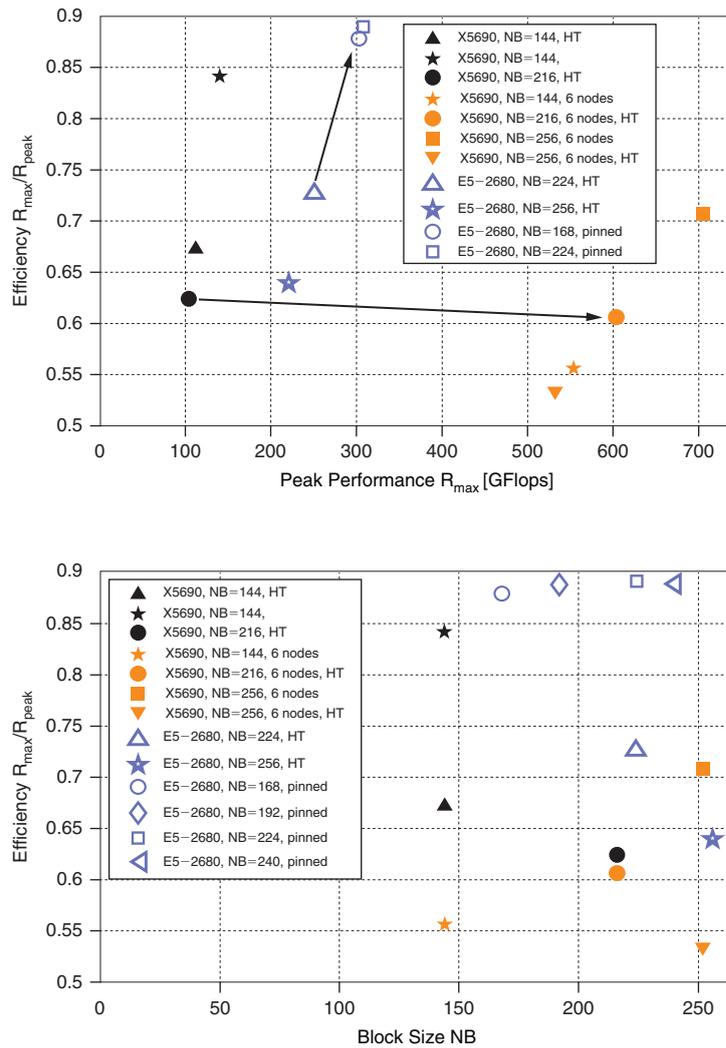


Figure 2: Measured Linpack performance using Open MPI 1.6.4 and Open BLAS 0.2.8 on both a Westmere-EP and a Sandy Bridge CPU architecture. Interconnection is done using QDR InfiniBand. (Source: Intel Corporation, 2014)

Radio Propagation Modeling

Current and next-generation wireless communication systems need to be well planned prior to rollout. In fact it is key to choose for an optimized number of access nodes, so to provide the required coverage and initial capacity in the beginning. This also allows to evaluate migration paths towards cell densification or further capacity enhancing hardware and feature updates at the macro-cellular sites. In order to evaluate such complex communication systems, we need accurate channel models to capture the performance-relevant effects of the wireless transmission channel. Therefore, the geographic topology, positions of base stations, cell layout, frequency bands to be used, and the antenna configurations at the base station and terminal side will greatly influence the wireless system performance. In order to keep the

“Figure 2 shows highest efficiency without hyperthreading as well as a linear increase in peak performance with amount of nodes in the cluster.”

“...we need accurate channel models to capture the performance-relevant effects of the wireless transmission channel.”

“...wireless channel models were derived and applied as performance benchmarking and feature performance comparison in standardization and for radio cell planning.”

“Quadriga was recently published under GPL license and extends the family of the Winner channel models with several features.”

system complexity manageable and the dominant physical effects sufficiently reflected in the evaluation, specific wireless channel models were derived and applied as performance benchmarking and feature performance comparison in standardization and for radio cell planning.

The guidelines of the 3GPP^[16] spatial channel model (SCM)^{[9][10]} introduce a ray-based bidirectional multilink model. The model was improved by the European project called Wireless World Initiative New Radio (WINNER)^[8], to cover emerging requirements for 3GPP standardization of future cellular air interfaces such as Long Term Evolution (LTE) or LTE-Advanced (LTE-A). The fundamental idea of those channel models is to emulate the wireless channel with a set of rays, having a direct connection or being scattered at obstacles in the surrounding environment denoted as line-of-sight (LOS) and non-line-of-sight (NLOS), respectively. Each ray arrives at the receiver with a certain delay and power under a deterministic angle for the LOS connection. For the NLOS or multipath components (MPC), this angle is following certain geometry, yielding a multi-tap channel profile. The Fraunhofer Heinrich Hertz Institute developed a geometric-stochastic channel model denoted as Quasi-Deterministic Radio Channel Generator (Quadriga). *Quadriga*^{[5][6][7]} was recently published under GPL license and extends the family of the Winner channel models with several features. The most prominent features are the 3D quasi-deterministic propagation assumptions, time evolution, and antenna representation with geometric polarization. Furthermore, *Quadriga* enables coherent generation of propagation conditions for different cell hierarchies such as macro cells, indoor and outdoor small cells, and satellite to ground communications, as shown in Figure 3.

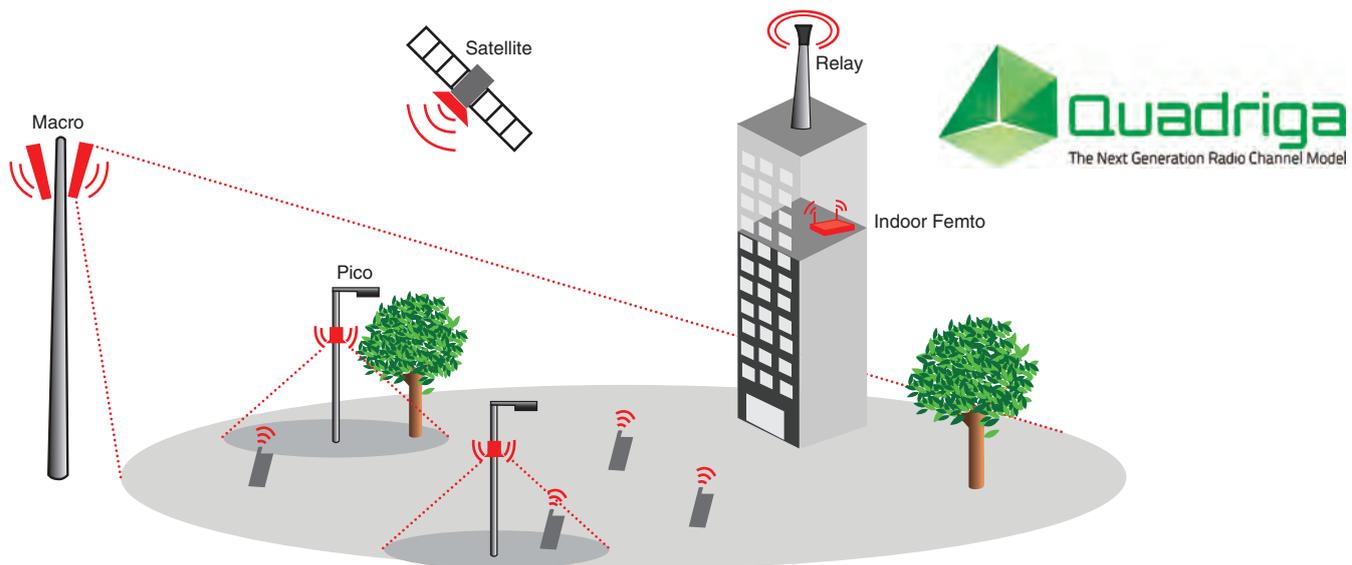


Figure 3: Wireless HetNet deployment scenario to be represented by Quadriga channel model (Source: Fraunhofer HHI, 2013^[2])

Such wireless channel models contain statistical and deterministic processes that generate a very high volume of data to accurately model wireless system parameters such as channel time evolution and antenna polarization for hundreds of thousands of wireless links. An overview of the modeling steps is shown in Figure 4. The user provides the network layout, that is, the positions of the base stations, antenna configurations, antenna downtilts, the positions and trajectories of the mobile terminals (MTs), and the propagation scenarios. The channel coefficients are then calculated in seven steps that are described by Jaeckel et al.^[5] in sections 3.2 and 3.3.

“...very high volume of data to accurately model wireless system parameters such as channel time evolution and antenna polarization...”

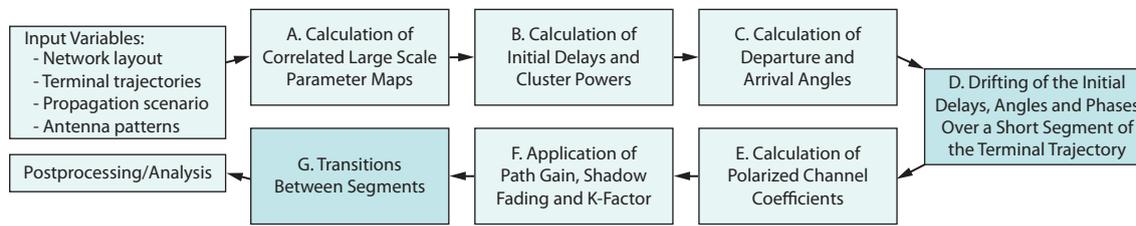


Figure 4: Block diagram from Quadriga Documentation^[2]
(Source: Fraunhofer HHI, 2013^[5])

The computational complexity for reasonable cell layouts increases quickly and easily exceeds the processing and storage capabilities of a standalone desktop or laptop computer. Table 2 lists the required storage space depending on the desired deployment topology and amount of antennas per link and time samples (snapshots), as well as independent Monte-Carlo drops.

“The computational complexity for reasonable cell layouts increases quickly and easily exceeds the processing and storage capabilities of a standalone desktop or laptop computer.”

Type	eNBs	ues	ant_eNB	ant_ue	taps	drops	Channel coefficients (1e9)	Needed Storage (GB)
Macros	21	630	4	2	20	500	2	47
Macros	57	630	4	2	20	500	6	128
Picos	42	630	2	2	16	500	2	38
Picos	114	630	2	2	16	500	5	103
LSAS	57	630	64	2	20	500	92	2,055
LSAS	57	630	128	2	20	500	184	4,110
LSAS	57	630	256	2	20	500	368	8,219
LSAS	57	630	512	2	20	500	735	16,438
LSAS	57	630	1024	2	20	500	1471	32,877

Table 2: Complexity of propagation modeling. For future wireless networks, such as large-scale antenna systems (LSAS), the number of channel coefficients is growing beyond billions.
(Source: Fraunhofer HHI, 2013^[2])

Table 3 summarizes the computation time for a 57 macro base station setup on a standard laptop using two hyperthreaded cores. The computation time can be reduced from 300 hours down to 9 hours when submitting the processing job to HPC with 140 hyper-threaded cores. The time saving scales linearly with

Quadriga simulation with 450 GB disk space	Laptop with Intel® Core™ i7 processor	HPCC
Involved Cores	4 @ 3.4 GHz	140 @ 3.4 GHz
Processing time	300 hours / 12.2 days	9 hours

Table 3: Cluster performance at the application level: it involves propagation modeling through Quadriga^[2] and uses both extensive processing and storage capabilities.

(Source: Intel Corporation, 2014)

the amount of CPU cores. Note: the dimension for parallel processing is sufficiently large since we use an amount of 500 independent Monte-Carlo drops. Figure 5 depicts a typical coverage plot showing the signal-to-interference-and-noise ratio in a map layout. This metric is influenced by the path loss, the shadow fading, antenna pattern and transmit power per base station.

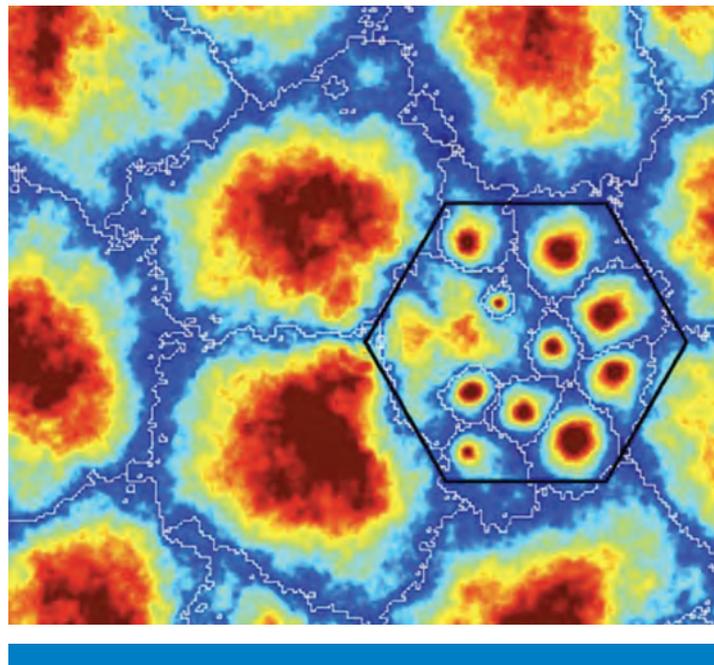


Figure 5: Wideband SINR coverage plot for a heterogeneous cellular deployment. This is usually considered as a high-level result from propagation modeling.

(Source: IEEE Asilomar Conference, 2013)

Other Applications for HPC Use

Other use cases for HPC usage are tasks requesting a heavy processing power, such as large-scale radio network system-level analysis for LTE/LTE-Advanced, data format conversion, or data analytics. A prominent example of data format conversion is video format re-encoding and/or rescaling in order to provide appropriate video formats to be transmitted over the Internet and to be displayed on a variety of screens. Since nowadays about half the Internet

“...large-scale radio network system-level analysis for LTE/LTE-Advanced...”

traffic is video, the offering of an appropriate video size and compression/encoding format is of utmost importance. In fact it not only provides the basis for a good quality of experience for the end user, but also is needed to achieve a fast processing of the packets through the IP network, in order to allow fast download or streaming, and significant overall system energy savings. Different access bandwidths on the server and client side have to be supported, as well as a wide range of user equipment capabilities. The latest trends advocate cloud edge servers in order to reduce the data transport distance and therefore application-layer response time at least for predictable content to be consumed in the near future or by many people in the same area. As a consequence, jobs like data format conversion have to be done locally and distributed just where and when the need occurs. In our chosen example this means video re-encoding at the cloud edge if a video streaming or download on demand is requested from a nearby user. In an extreme case this might mean virtual real-time re-encoding of YouTube* videos for users employing poor bandwidth wireless connections to the Internet or overloaded cells. This provides a good motivation for localized signal processing using scalable HPC architectures that can be extended depending on needed and observed data traffic and signal processing requirements.

Another prominent application is big data analysis. In this context we consider a big data analysis job to be performed where the huge amount of data to be analyzed can be distributed over many locations or is aggregated at a single location. In the former case, usually data analysis clients or agents (software modules) are executed at each location and analysis results or condensed information and data is forwarded and aggregated at a centralized point or forwarded to another location. These distributed agents do part of the analysis separately and forward the condensed part of the information to another more centralized data analysis instance. By doing so, data and preprocessed data from many locations can be analyzed in depth for correlations and other results of interest at the central location. For such applications HPC is the recommended architecture to do the job.

Advanced Real-Time Signal Processing

Besides large data processing for system behavior and system performance evaluation at the system level, many applications require true real-time processing for signal or data format conversion, signal or data transmission, or information extraction/collection for such things as pattern recognition. Real-time in this context means that the signal or data processing capability of the computing machine can keep pace with the speed and amount of incoming data or signals. Computation-induced delays must be limited to tolerable delays dictated by the application itself or the process that follows. For many applications these real-time requirements are stringent and in state-of-the-art solutions are often addressed with signal-processing task-matched hardware and hardware-based signal processing accelerators.

A prominent example is real-time video compression for live streaming of camera pictures, such as for live events like open air concerts or content

“...jobs like data format conversion have to be done locally and distributed just where and when the need occurs.”

“...localized signal processing using scalable HPC architectures that can be extended depending on needed and observed data traffic and signal processing requirements.”

delivery from many cameras in a stadium to users or distribution centers. Since the application like a football game has to be streamed in real-time to the consumers at minimized latency, real-time processing is a must. The latest video standards such as HEVC aka ITU-T Rec. H.265 allow the specific compression rate to be a function of the input/output bit rate versus time. Such parameter space makes it possible to provide the appropriate video format for the transmission bandwidth and the video display size and decoder capability requesting the streaming service. Also the number of videos and video formats encoded simultaneously in real-time demand scalable HPC architectures.

HEVC/H.265

In January 2013, and ten years after the widely-used ITU-T Rec. H.264/MPEG-4 AVC video coding standard was published, the High Efficiency Video Coding (HEVC) standard or ITU-T Rec. H.265 was finalized, published in June 2013 by ITU-T and in November 2013 by ISO/IEC. The coding efficiency of HEVC achieves a bit rate reduction of 50 percent for the same subjective quality compared to the best profile of H.264/MPEG-4 AVC, the High Profile. Thus it allows for doubling the number of services using HEVC compared to H.264 if using, for example, the same network resources.

The coding efficiency gain of HEVC comes with an increased complexity on the encoder side, the coding complexity has increased by roughly a factor of 10 to 15 compared to H.264/MPEG-4 AVC. Therefore the HEVC coding standard was prepared to be executed on multiprocessor, high performance platforms.

Unlike H.264/MPEG-4 AVC, where parallelism was an afterthought, the HEVC design contains several techniques making the codec better “parallelizable.” H.264/MPEG-4 AVC supports slices, which were introduced mainly to prevent loss of quality in the case of transmission errors, but can also be used to parallelize the encoder. Employing slices for parallelism, however, introduces several problems such as a reduced video quality. The two main parallelization approaches included in the HEVC design are Tiles and Wavefront Parallel Processing (WPP). Both allow for creating picture partitions, but only the latter furthermore allows parallel processing without incurring any significant coding losses. Tiles not only target an effective parallelization of the codec, but also are optimized for conversational services.

The idea of WPP is to partition the picture into rows of video processing blocks, which are called in HEVC Coding Tree Units (CTUs). Each row is to be processed by a different core or processor in what is called a wave-front parallel fashion. That is, while coding such blocks in a so-called raster scan order from the top to the bottom row plus from the very left to the very right block per row, coding such blocks is dependent on prior coded blocks, that is, the upper left one and the right one block. This dependency gives the name to the parallel processing procedure, which is not all processors start their rows to process at the same time, but by a shift, so that the upper left block is available from the predecessor row’s coding process. The picture below shows the “shifted” and

“...coding efficiency of HEVC achieves a bit rate reduction of 50 percent...”

“...the coding complexity has increased by roughly a factor of 10 to 15 compared to H.264/MPEG-4 AVC.”

“Each row is to be processed by a different core or processor in what is called a wave-front parallel fashion.”

row-wise processing fashion of WPP. The very new thing of WPP in HEVC is that the wave-front processing is not only applicable to the transform coding but also to the entropy coding parts of the hybrid video coding process. This means that in HEVC the full coding process can be executed in wave-front fashion, where for example in H.264/MPEG-4 AVC only the transform coding part was processable in a wave-front fashion. Figure 6 also shows the entropy coding dependencies as arrows for the very first left block of each row. This arrow indicates the initialization value for the entropy coding process, which originally was taken from the last block of the predecessor CTU row.

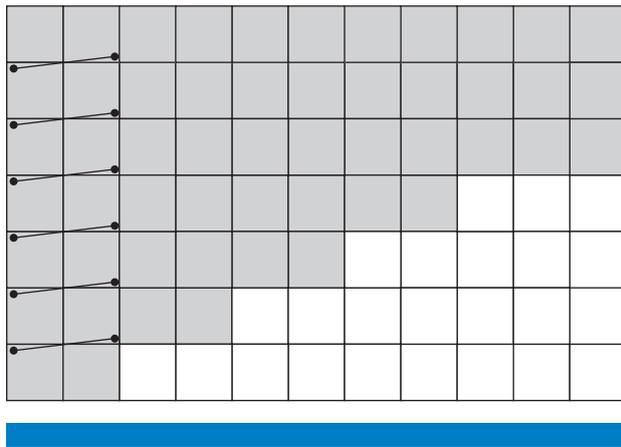


Figure 6: HEVC Wavefront Parallel Processing scheme
(Source: IEEE Transactions on Circuits and Systems for Video Technology, 2012)

“In HEVC, both transform and entropy coding process can be executed in wave-front fashion.”

Since WPP’s parallelization is limited by the number of CTU rows, WPP may even further scale on multi-core systems if a picture-overlapping approach were used, that is, the next picture is already processed, while the process of the current picture is still ongoing. We called that process Overlapping Wave-front Processing (OWP), as presented in [TCSVT-HEVC]. Figure 7 illustrates the OWP approach, where Thread T4 already processes the next picture, while Threads T1–T3 are still processing the current picture.

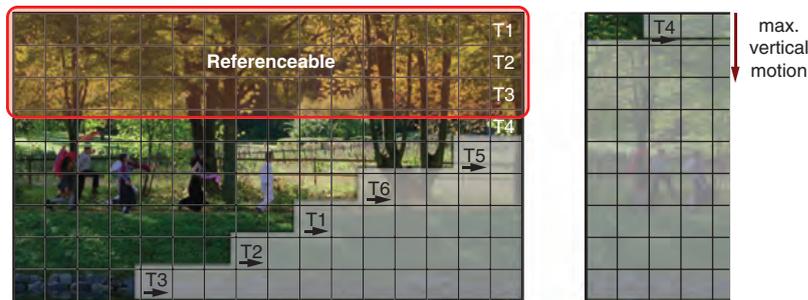


Figure 7: Overlapping Wavefront scheme
(Source: IEEE Transactions on Circuits and Systems for Video Technology, 2012)

“The increasing capabilities of COTS computing hardware allows for virtualization of many signal processing functions, also known as network function virtualization (NFV),...”

The parallelization techniques of HEVC may allow the extensive use of the processing performance available on multi-core/multiprocessor platforms as present in high performance clusters.

Cloud-RAN Baseband Signal Processing

Another true real-time signal processing task is motivated by centralized signal processing or wireless transmission signals. The current trend of software-defined radio architectures in the radio access network (RAN) allow more and more software-based implementations of signal processing functions. The increasing capabilities of COTS computing hardware allows for virtualization of many signal processing functions, also known as network function virtualization (NFV), where the actual virtualization started originally from core network functions and is going step by step closer to the near antenna signal processing (see Figure 8).

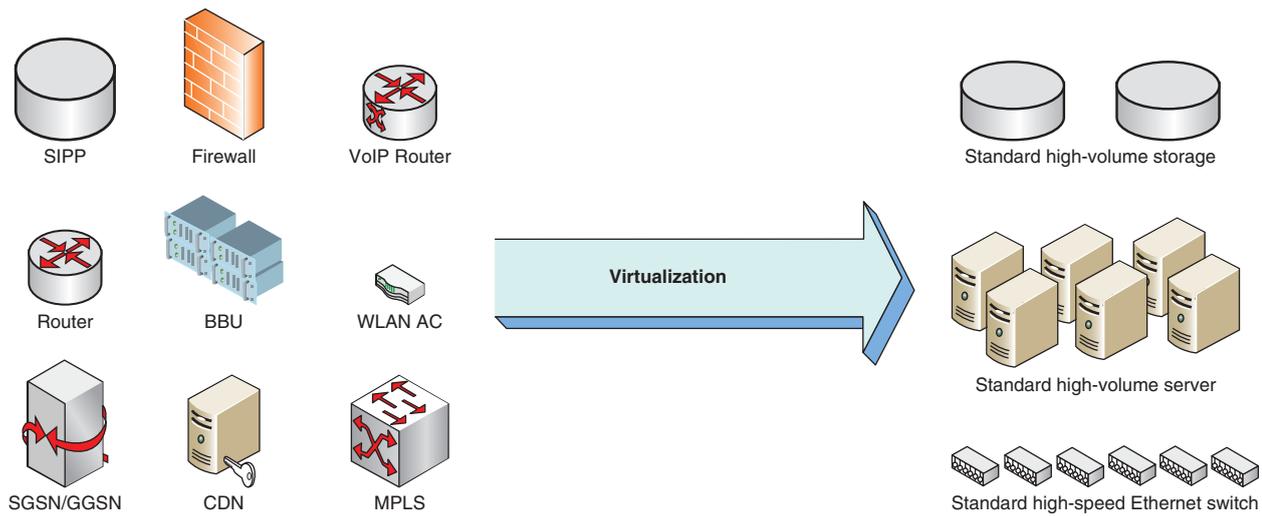


Figure 8: Cloud RAN architecture exploiting network function virtualization
(Source: Intel Corporation, 2014)

The benefits expected from such RAN virtualization are:

- Multi-RAT support: each RAT can be implemented as one virtual machine on a single piece of computing hardware.
- Cost reduction and resource utilization improvement: multiple independent BBU (base band unit) entities fit into the same physical server (see Figure 9).
- Live migration to consolidate resources, so to save power.
- Resource sharing and consolidation according to traffic variance.

“...platforms require a real-time OS and hardware accelerators for radio-standard-specific high complexity algorithms...”

Two stringent requirements should be mentioned when considering standard servers. Most open platforms require a real-time OS and hardware accelerators for radio-standard-specific high complexity algorithms like matrix-vector multiplications, matrix inversion, turbo-decoding, or fast Fourier transforms (FFTs).

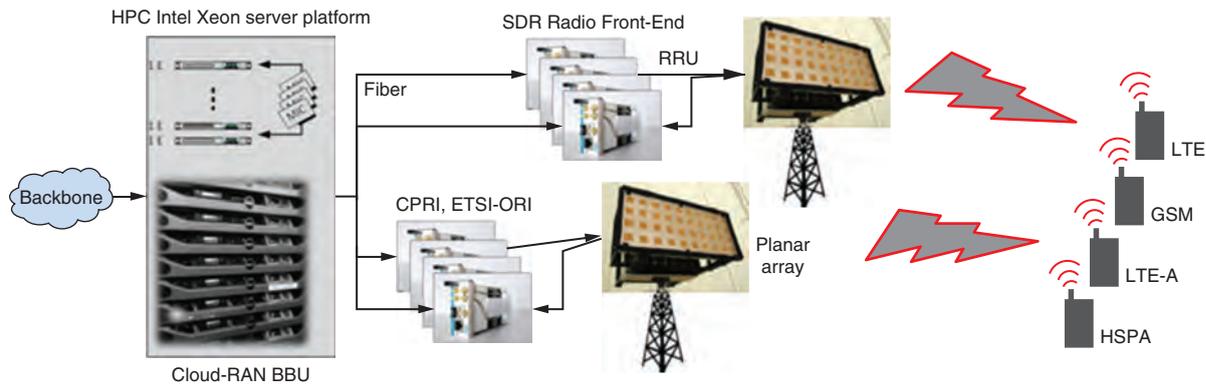


Figure 9: Server-based BBU (base band unit) and single RRU (remote radio unit) allowing multi-RAT processing in the same BBU.
(Source: Intel Corporation, 2014)

As an example, the signal processing complexity of the fourth generation radio standard LTE is shown, where the signal processing tasks can be shifted and split appropriately between BBU and RRU.

The red lines in Figure 10 depict splitting options between BBU and RRU in order to have a balance signal processing, either more centralized or more distributed at the remote radio unit (RRU).

1. Soft-bit fronthaul (softbits plus control information).
2. Subframe data fronthaul (frequency domain I/Q plus control).
3. Subframe symbol fronthaul (frequency domain I/Q).
4. CPRI/OBSAI/ETSI-ORI fronthaul (time domain I/Q).
5. Compressed CPRI/OBSAI/ETSI-ORI fronthaul (time domain I/Q).

“...splitting options between BBU and RRU in order to have a balance signal processing...”

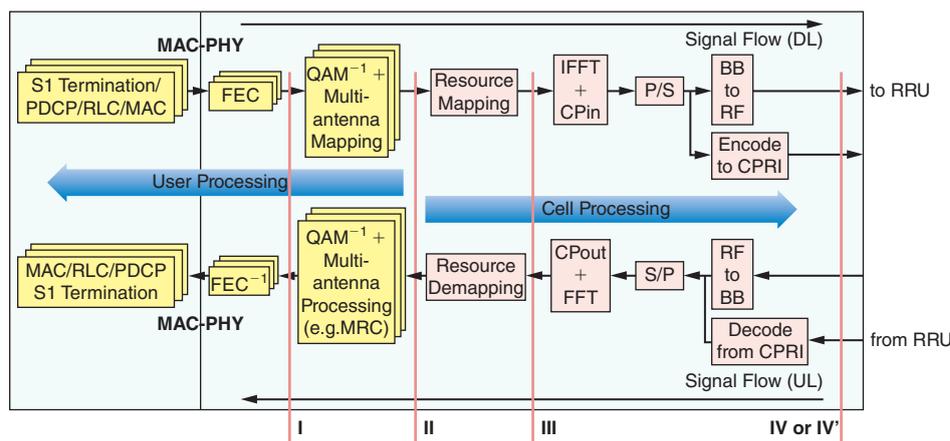


Figure 10: Server-based BBU and single RRU allowing multi-RAT processing in the same BBU.
(Source: Intel Corporation, 2014)

“...150 Gbps of data rate would be needed just to drive the interface between BBU and the three multiband multi-antenna RRUs at one macro site.”

“...real-time at a processing latency end-to-end on IP level of about 4–5 ms...”

Furthermore, this has an impact on communication data rate on the interface between BBU and RRU. The Common Public Radio Interface (CPRI) as an example requires 2.5 Gbps for a 20 MHz LTE carrier and two transmit and two receive antennas per RRU.

Taking into account that a macro site consists of three sectors minimum, with up to eight transmit and receive antennas, the interface communication data rate multiplies by 12. Further capacity increase per site can be obtained by carrier aggregation techniques, which can allow bonding up to 5–20 MHz LTE channels (2.5 Gbps \times 12 \times 5 = 150 Gbps of data rate would be needed just to drive the interface between BBU and the three multiband multi-antenna RRUs at one macro site). CPRI latency requirements are in the range of 100 μ s depending on the particular signal processing task to be performed, for example coordinated multi-point (CoMP) over X2.

Assuming as the chosen scenario that several dozen macro eNBs are connected to the cloud RAN BBU plus a number of small cells (a 3GPP compliant implementation would imply up to 3–10 small cells per macro sector), this would rationalize the tremendous amount of signal processing power to perform real-time baseband signal processing and real-time interconnections between BBU and many RRUs.

To provide some latency constraints within the LTE framework we have to process all cell data in real-time at a processing latency end-to-end on IP level of about 4–5 ms one way, since hybrid automatic repeat request (H-ARQ) acknowledgments are pre-scheduled for retransmission requests. Parallel signal processing can be done at most stages but some algorithms like MIMO equalizers can scale cubically with the number of antennas to be processed jointly, which is especially challenging when going for advanced cooperative signal processing like CoMP, or the number of pilots involved for advanced channel estimation all to be done as fast as possible in order to keep signal processing latency to a minimum.

Figure 11 depicts the signal processing chain of an OFDM receiver for a multi-antenna single-component LTE carrier in the down link. Considering that MIMO channel estimation can be processed in parallel per channel coefficient, this means 1200 OFDM subcarriers per antenna pair of N transmit and M receive antennas. After channel estimation and interpolation per antenna pair, the channel equalizer has to be calculated considering the spatial structure between the antennas. Therefore the next step of equalizer calculation can be parallelized along the subcarrier axis. Then the equalizer is applied onto the receive signals, which can be computed in parallel by matrix times vector multiplications per subcarrier again.

In the next step, user-related data has to be separated from the allocated OFDM symbols and subcarriers defining physical resource blocks (PRBs) allocated to each user. These data symbols are to be demodulated and decoded separately for each user, a scalable option to work in parallel depending on the number of users to be processed.

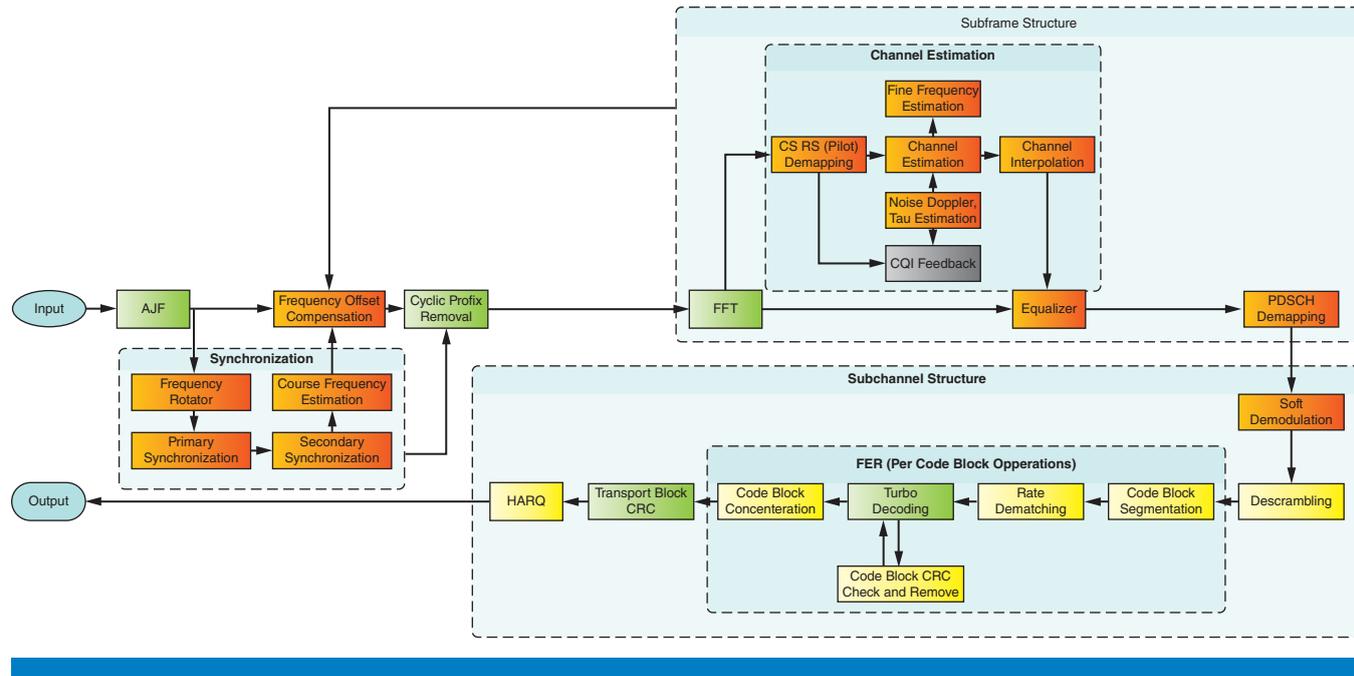


Figure 11: LTE real-time signal processing OFDM receiver chain
(Source: Intel Corporation, 2014.)

If such processing is to be extended towards more transmit and receive antennas, then crosswise signal processing is possible for performance enhancements but at a price of increased signal processing complexity. If this is done all at the same location, in the same computing hardware using an HPC architecture, these feature enhancements and hardware extensions can be realized on demand.

In order to provide some reasoning of state of the art approaches, most eNBs are software defined radio (SDR) based but with dedicated DSPs for various reasons: real-time constraints require low-level hardware implementation of the signal processing routines; dedicated hardware macros are available on these platforms, such as FFTs, turbo-decoder, and matrix-vector multiplications. Furthermore, energy consumption per base station affects operating expenses, therefore these DSPs are tailored for the specific application in wireless communications. On the downside, feature upgrades have to be within the available hardware processing capability available at the distributed eNBs. Further hardware upgrades on remote locations can become quite costly and in many cases require extensive changes in the software partitioning over several boards on vendor-specific interfaces.

As a current trend, some vendors consider clustering of such dedicated signal processing hardware in centralized shelves (HPC) in order to have the benefits of centralized inter-eNB signal processing and the option of adding more dedicated hardware if needed.

“...ideally suited for SMEs and R&D organizations...”

“...HPC allows flexibility in signal processing enhancements for evolutionary feature extensions and scalability in computing performance...”

“...small- to medium-size scalable HPC architectures providing new application space for SMEs and R&D centers...”

Conclusion

In this article the concept of a small-scale HPC architecture was introduced, which is ideally suited for SMEs and R&D organizations that have to deal with high-complexity signal and data processing with scalability in amount of data and available computing time.

The proposed HPC architecture is shown to be scalable in terms of hardware components including server blades and attached memory banks on the one hand and on the other hand showed linear scalability regarding processing performance benchmarked with state of the art tools.

Furthermore we showed the applicability of such scalable HPC architectures for various application spaces and illustrated the capabilities and advantages with three particular examples taken from the wireless system evaluation and signal processing domain, wireless channel modeling, wireless multicellular system-level analysis, and cloud-RAN based real-time signal processing for LTE. These examples show that HPC allows flexibility in signal processing enhancements for evolutionary feature extensions and scalability in computing performance, if more of the same or more in-depth processing is to be done at a specific location.

Summarizing the article highlights current trends, moving away from the two classic extremes of distributed small signal processing units and fully centralized big data centers towards small- to medium-size scalable HPC architectures providing new application space for SMEs and R&D centers in the wireless industry. The flexibility, scalability, and extendibility of such HPC architectures allow a new degree of freedom in CAPEX and OPEX optimization for a ubiquitous application space based on commercial-of-the-shelf hardware.

References

- [1] Joseph, E., J. Wu, S. Conway, and S. Tichenor, “Benchmarking industrial use of high performance computing for innovation,” White Paper, Council on Competitiveness, 2008.
- [2] Quadriga, <http://quadriga-channel-model.de>
- [3] <http://www.netlib.org/Linpack/>
- [4] <http://www.netlib.org/blas/>
- [5] Jaeckel, S., L. Raschkowski, K. Börner, L. Thiele, F. Burkhardt, and E. Eberlein, “QuaDRiGa - Quasi Deterministic Radio Channel Generator, User Manual and Documentation,” Fraunhofer Heinrich Hertz Institute, Tech. Rep. v1.0.5–171, 2013.
- [6] Jaeckel, S., K. Börner, L. Thiele, and V. Jungnickel, “A Geometric Polarization Rotation Model for the 3D Spatial Channel Model,”

- IEEE Transactions on Antennas and Propagation*, vol. 60, no. 12, pp. 5966–5977, December 2012.
- [7] Jaeckel, S., L. Raschkowski, K. Börner, and L. Thiele, “QuaDRiGa: A 3-D Multicell Channel Model with Time Evolution for Enabling Virtual Field Trials,” submitted to *IEEE Transactions on Antennas and Propagation*, 2013.
- [8] Kyösti, Pekka, Juha Meinilä, Lassi Hentilä, et al. IST-4-027756 WINNER II D1.1.2 v.1.1:WINNER II channel models. Technical report, 2007.
- [9] Baum, D. S., J. Hansen, and J. Salo. “An interim channel model for beyond-3G systems,” *Proc. IEEE VCT '05 Spring*, 5:3132–3136, 2005.
- [10] 3GPP TR 25.996 V6.1.0, “Spatial channel model for multiple input multiple output (MIMO) simulations (Release 6),” Tech. Rep., Sep. 2003. Online Available: <http://www.tkk.fi/Units/Radio/scm/>.
- [11] Chi, C. C., M. Alvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl: Parallel Scalability and Efficiency of HEVC Parallelization Approaches, *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE TCSVT, Special Issue on Emerging Research and Standards in Next Generation Video Coding, vol. 22, issue 12, pp. 1827–1838, 2012.
- [12] <http://www.intel.com/content/www/us/en/architecture-and-technology/many-integrated-core/intel-many-integrated-core-architecture.html>.
- [13] <http://www.intel.com/content/www/us/en/processor-comparison/processor-specifications.html?proc=52576>.
- [14] http://www.theregister.co.uk/2010/03/16/intel_xeon_5600_launch/
- [15] <http://ark.intel.com/products/64583>.
- [16] <http://www.3gpp.org/>.

Author Biographies

Lars Thiele (lars.thiele@hhi.fraunhofer.de) received the Dipl.-Ing. (M.S.) degree in electrical engineering from the Technische Universität Berlin in 2005. He joined the Fraunhofer Heinrich Hertz Institute (HHI) in September 2005. In 2013 he received the Dr.-Ing. (PhD) degree from the Technical University of Munich (TUM). He has contributed to receiver and transmitter optimization under limited feedback, performance analysis for MIMO transmission in cellular ODFM systems, fair-resource allocation, and CoMP

transmission under constrained CSIT. Lars has authored and coauthored about 50 conference and journal papers as well as a couple of book chapters in the area of mobile communications. He leads the System Level Innovation research group at Fraunhofer HHI and is actively participating in the GreenTouch Consortium.

Thomas Wirth (thomas.wirth@hhi.fraunhofer.de) received a Dipl.-Inform. (M.S.) degree in computer science from the Universität Würzburg, Germany, in 2004. In 2004 he joined Universität Bremen, Germany, where he worked in the field of robotics. In 2006 he joined HHI's WN department as senior researcher with the focus on real-time implementation for future wireless SDR prototypes. Since 2011, Thomas is head of the Software Defined Radio (SDR) group with the interest on algorithms for baseband processing including PHY and MAC as well as cross-layer design techniques for optimized video transmission over wireless systems.

Michael Olbrich (michael.olbrich@hhi.fraunhofer.de) is studying electrical engineering at the Technische Universität Berlin. Currently, he is working towards the Dipl.-Ing. degree (M.Sc.) at the Berlin Institute of Technology and the Fraunhofer Heinrich Hertz Institute (HHI). From 2003 until 2007, he was with Siemens Mobile (later Nokia Siemens Networks) and joined HHI in 2008. His research interests include performance evaluation of communication systems, MIMO-OFDM transmission, and high-performance computing.

Thomas Schierl (thomas.schierl@hhi.fraunhofer.de) received the Dr.-Ing. degree in Electrical Engineering (passed with distinction) from Berlin University of Technology (TUB) in October 2010. He is head of the research group Multimedia Communications in the Image Processing Department at Fraunhofer Heinrich Hertz Institute (HHI), Berlin. Thomas is the co-editor of various IETF RFCs and various MPEG standards. In 2007, Thomas visited the Image, Video, and Multimedia Systems group of Prof. Bernd Girod at Stanford University, CA, USA for different research activities. Thomas' research interests include system integration of video codecs, delivery of real-time media over mobile IP networks such as mobile media content delivery over HTTP, and real-time multimedia processing in cloud infrastructures.

Thomas Haustein (thomas.haustein@hhi.fraunhofer.de) received the Dr.-Ing. (Ph.D.) degree in mobile communications from the Technische Universität Berlin in 2006. In 1997, he joined HHI in Berlin, where he worked on wireless infrared systems and radio communications with multiple antennas and orthogonal frequency division multiplexing. He focused on real-time algorithms for baseband processing and advanced multiuser resource allocation. In 2006, he joined Nokia Siemens Networks, where he conducted research for LTE and LTE-Advanced. He is currently the head of the Wireless Communication and Networks Department at Fraunhofer HHI.

Valerio Frascolla earned his MSc in electrical engineering in 2001 and his PhD in electronics in 2004. He worked as research fellow at Ancona University, Italy, then moved to Germany, joining Comneon in 2006 and Infineon Technologies in 2010. Since 2011 he has been funding and innovation manager at Intel Mobile Communications, acting as facilitator of research collaborations using Agile methodologies and focusing on the program management of publicly funded projects and innovation activities. He is author of several peer-reviewed scientific publications and has been an invited speaker at international events. Email: valerio.frascolla@intel.com.