Is the EDA Industry Ready for Cloud Computing?

Article *in* IETE Technical Review · October 2015

DOI: 10.1080/02564602.2015.1099056

CITATIONS
10

READS
2,889

3 authors:

Naresh Sehgal
Intel
51 PUBLICATIONS **468** CITATIONS
SEE PROFILE

John M. Acken
Portland State University
106 PUBLICATIONS **935** CITATIONS
SEE PROFILE

Sohum Sohoni
Arizona State University
36 PUBLICATIONS **311** CITATIONS
SEE PROFILE

IETE Technical Review

ISSN: 0256-4602 (Print) 0974-5971 (Online) Journal homepage: http://www.tandfonline.com/loi/titr20

Is the EDA Industry Ready for Cloud Computing?

Naresh Sehgal, John M. Acken & Sohum Sohoni

To cite this article: Naresh Sehgal, John M. Acken & Sohum Sohoni (2015): Is the EDA Industry Ready for Cloud Computing?, IETE Technical Review, DOI: 10.1080/02564602.2015.1099056

To link to this article: http://dx.doi.org/10.1080/02564602.2015.1099056

IETE TECHNICAL REVIEW, 2015
http://dx.doi.org/10.1080/02564602.2015.1099056

Is the EDA Industry Ready for Cloud Computing?

Naresh Sehgal[a], John M. Acken[b] and Sohum Sohoni[c]

[a]Platform Engineering Group, Intel Corporation, Santa Clara, CA, USA; [b]Electrical and Computer Engineering, Portland State University, Portland, OR, USA; [c]School of Computing Informatics and Decision Systems Engineering, Arizona State University, Mesa, AZ, USA

ABSTRACT

Many industries are increasingly adopting cloud computing. There are several electronic design automation (EDA) industry players, large and small companies, who have explored the idea of providing cloud-based very large-scale integration design tools and services. This paper briefly explores the history of EDA solutions and their growth path thus far, starting with standalone computer aided design (CAD) tools, through specialized EDA workstations, to integrated suites of tools and flows as currently provided by EDA vendors. A representative EDA flow and its steps are described to provide a basis for relating individual EDA tools to appropriate workload categories. Each step in the EDA design flow is then mapped to a cloud computing workload category. This mapping provides a basis for a decision on moving particular EDA design flow steps to a cloud computing environment. This paper also lists some capabilities currently offered by the public and private cloud providers as a basis for looking at the challenges and opportunities for migrating EDA solutions to cloud computing.

KEYWORDS

EDA; cloud computing; VLSI; design flows

1. Introduction

Electronic design automation (EDA) broadly refers to the software tools, methods, flows, and scripts that are used for very large-scale integration (VLSI) designs of silicon-based circuits, chips, and systems. The sheer number of transistors (now ranging into several billions) that can reside on a single piece of silicon prohibits a hand-crafted design and necessitates automation. The software technologies behind such automation have evolved over the last five decades, starting with university and large research lab tools, such as Magic [1] (for basic layout placement), SPICE (for circuit simulation) [2], SUPREM [3] (for IC fabrication process simulation), PISCES [3,4] (for semiconductor device simulation), SALOGS [5,6] (for logic simulation), TEGAS [7] (for logic simulation), MOSSIM [8] (switch level simulation), SCOAP [9] (for testability analysis), TMEAS [10] (testability analysis), ESPRESSO [11] (for logic minimization), SICLOPS [12] (for place and route), and Timberwolf [13] (for place and route) to several higher level logic synthesis solutions in 1980s. However, these technologies have since moved from academic institutions to many commercial companies that specialize in providing EDA tools and necessary support for design teams to deploy them. The growth rate for the EDA companies has been cyclical during the past decade due to various factors, and has opportunities for greater growth [14].

So many industries are evaluating moving to the cloud that researchers in Dublin have produced a systematic literature review of 23 selected studies, published from 2010 to 2013 [15]. The authors produced a database of migrations of legacy software to cloud, which provides useful information for the EDA companies considering such a move. Also, work has been done to identify performance bottlenecks in cloud computing [16]. The authors concentrate on network loading as affected by cloud workload. Moreno et al. [17] present an analysis of the workload characteristics derived from a production cloud data center and describe a method for analyzing and simulating cloud workloads. However, they point out that: "Finally, the workload is always driven by the users, therefore realistic workload models must include user behavior patterns linked to tasks." p. 210. We endeavor to meet this requirement by relating the general cloud computing benefits to the EDA industry specific workload characteristics. Successful utilization of cloud computing by the EDA industry, and the chip design industry that uses it, can create an opportunity for greater growth for both the EDA companies and the IC design community. Previous publications have discussed cloud workload classifications [18] and cloud security [19]. The contribution of this paper is to apply that analysis to the EDA industry.

2. Problem statement

The EDA field is already several decades mature and one wonders if there has been enough time to solve many, if

not all, of the key problems in the domain of silicon design automation (DA). The answer, however, is no, as the problems are not static in nature. Many of these problems are NP complete or NP-hard in nature, with heuristic solutions. With Moore's law of doubling the number of transistors that are available for integration every 18 months (current estimates are between 24 and 30 months), any heuristic solution tends to break in a couple of generations of semiconductor products, as the design solution space grows exponentially. This increases the cost of doing designs and is thus limiting the number of new commercial design starts, further putting a financial pressure on the EDA providers to recover their investments [20]. Thus, despite a growing electronics consumers market, the EDA industry enjoys a growth rate slower than its design customers, at 7.9% over the past one year [14]. Paradoxically, a large number of small design efforts can readily use existing tools only if the EDA cost comes down allowing a broader adoption. This is especially true for new emerging applications, such as Internet of things, where per unit silicon cost tends to be measured in cents instead of dollars. Broader access to large pools of inexpensive computing resources and availability of EDA tools to a broad number of current and potential users may result in reducing the cost of doing new designs. Therefore, more designs can be started by even smaller teams and companies [20], especially in the emerging economies. Hence, a clear and present question for the EDA industry is, "How can Cloud Computing enable new growth opportunities?" To address this question requires looking at the EDA workloads, and map them to cloud workload categories. The cloud workload mapping in this paper applies to both public and private clouds. However, EDA tools and flows have evolved over time, and will continue to do so. Hence, a brief summary of the EDA history is essential to understand the rationale behind our proposed mapping, and to keep it updated with any future tools and flows.

3. Brief history of EDA tools and flows

3.1. The Nascent years of the 70s

Although computer aided design (CAD) and DA activities started in early 70s, for the first decade, almost all of the development was within large corporate design centers, using proprietary solutions. By the mid to late 70s, CAD and DA solutions branched into different engineering domains, such as automotive design, VLSI circuit design, and printed circuit board design. The chips of that era were mostly hand-crafted and thus the number of logic gates was limited to what a human mind could comprehend in terms of functionality. Circuits were of the simple adder and multiplier types, and point solutions included both analysis (e.g. equation solvers) and synthesis (e.g., simple logic minimization) type of stand-alone tools. During all of the 70s, digital electronic CAD and DA tools were mostly for board level design, some IC chip tools were being developed in-house, such as at Intel.

3.2. The roaring 80s

In the early 80s, specialized workstations were created for electronic design by many companies, of which the top three were Daisy, Mentor, and Valid. Tools for interactive schematics and layout designs were becoming commonplace, even in the published papers during that era [21], including for individual design synthesis and graphics with batch processing for large analysis problems.

Later, that decade saw specialized high end workstations created to solve large design problems, such as Sun Workstations becoming a hardware platform standard for using CAD/DA/EDA tools. The EDA industry began to focus on IC designs, while the term CAD became more common for mechanical engineering. The chips of that era were able to reach and cross the 1 million transistor mark with Intel's N10 design in 1989. Very large analysis (such as design rule checking a whole chip or exhaustive digital simulation for design verification) was still performed on corporate main-frame computers while incremental or small-scale analysis and simulation were performed on high-performance engineering workstations at each engineer's desk.

3.3. Growing up in the 90s

This decade witnessed the young EDA industry growing up fast, with many startups and subsequent consolidation phases, as EDA tools moved from specialized to general purpose workstations. The solution vendors and their customers started to talk about flows and frameworks instead of individual point tools. Synthesis and analysis EDA programs were linked in automated script-driven flows with feedback loops. Heuristics-based solutions were used to address the size limitations for mostly NP-complete or NP-hard problems common in the EDA domain. There was an increasing interplay between automatic and human-interactive roles, especially in planning the layout and signal busses, to prune the search space for place and route tools.

The chips in this era routinely reached hundreds of millions of transistors, and tool limitations were overcome with the use of design hierarchies and cell libraries. The

latter also turned out to be a major productivity booster [22]. During the mid-90s, a trend included adding memory, and cache to on-die microprocessors, to speed up the data processing times.

3.4. Maturing into the first decade of 21st century

With the Internet boom came large data-centers and massively parallel user tasks, such as database searches. These clearly needed multi-core servers, instead of large single core processor designs, which also became a convenient way to overcome the power and complexity limitations of large out-of-order superscalar single-core designs. This trend benefitted from the concept of hierarchical and cell-based modular designs developed in the previous decade. EDA evolved to increase inclusion of system level concepts with the growth of system on a chip (SoCs), comprising of intellectual property (IPs), cores, and custom logic integration supported by the tools.

Using the new EDA solutions, large chip designs, such as Intel's Itanium microprocessors crossed the 1 billion transistor mark. This era also saw an added emphasis on low-power designs with the popularity of smart phones, tablet computing, and various new form-factors. As individual processors became faster, a number of consumer products evolved to put more functionality in software on standard hardware cores, thereby decreasing the need for many new design-starts in the industry. According to Prof. Michael Porter's competitive five forces theory, in any industry, it is desirable to have many customers instead of a few large customers to minimize the product pricing pressures [23]. This applies to the EDA industry as the uneven split of customers between high-end large designs done by a few companies and the smaller size designs by many companies presents a support and development opportunity [20].

3.5. 2010s till now, EDA stable

With the advent of the Internet, higher bandwidth of public and private networks, and consolidation of servers in large data-centers to save on their power needs and cooling requirements came the birth of cloud computing. Compute power required to complete the new chip designs in acceptable times exceeded the available compute capacity in many design companies, thus creating a natural problem and solution match, almost requiring the EDA tasks to move into the cloud. However, the lack of security, or at least a perception of lack of security, prevented movement of proprietary chip designs into public clouds [24]. Also, licensing models for EDA tools have not evolved much since the 90s, requiring user companies to buy yearly contracts for tool licenses, instead of pay as you go software rental models so popular in the cloud.

Both of these factors are contributing to the limit on how many simultaneous designs a company can afford to do, thereby also limiting the revenue growth of the EDA industry. Solving this dilemma provides an opportunity for the EDA and chip design industries to keep growing in tandem.

Furthermore, a relentless pressure to reduce the project schedules and costs is causing most design teams to adopt SoC methodologies [25]. This has led many EDA companies to venture into a new territory to provide large design components, often sold as IPs blocks, for specific functions, such as audio or video processing, USB support and even CPUs as building blocks for larger and more complex chips. This is one step away from an EDA company doing the complete designs itself, but then they may not want to directly compete with their tools' customers. These points indicate opportunities for EDA companies that cloud computing can potentially offer.

4. Cloud workload categories

This section lists and briefly describes various categories of cloud computing workloads from customer and vendor viewpoints. The categories were defined and described in detail in [18]. The category definitions apply to both public and private clouds. Careful definitions for workload categories are important for many uses. For example, Moreno et al. [17] suggest that a comprehensive study of workload task categories and details is essential to ensure a diversity of workload tasks when creating a model for aggregated analysis in large-scale production computing centers. Workloads categories are defined by shared characteristics of tasks within that category. Only those workload categories deemed relevant to EDA are described in this paper. The categories may have overlap, indeed might be identical, from other viewpoints. Some categories that are included are not technically cloud computing categories. However, because most EDA applications are not yet on the cloud, it is important to include these other categories. The categories are linked and related to underlying computer resources; however, they are not defined by those resources. The distinction is that the categories are qualitative user recognizable definitions, while the underlying resources and metrics are detailed quantifiable metrics cloud suppliers and computer system designers need.

To reiterate, the emphasized categories are an operational viewpoint of cloud computing workloads for the EDA industry. Although some of the related implementation resources (architectural, hardware, software, and infrastructure) concepts may be listed to help understand the categories, these implementation resource concepts are covered in another paper [24]. Also, although some low-level example hardware metrics may be cited for clarity in a particular category, those metrics are not described in detail in this paper. The ultimate goal of this paper is to demonstrate overlaps and gaps between the EDA industry's expectations and the SLAs offered by cloud providers, illustrating opportunities and areas for potential improvements.

The categories of computing workloads considered in this section include both cloud computing workloads and workloads that are not commonly considered cloud computing issues. For example, high-end desktop graphics processing is not a cloud computing category. Also, be aware that the cloud workload categories can also represent categories of jobs that could be run in a dedicated non-cloud, private computing environment. That being said, the categories in this section include: big streaming data, big database calculation, big database access, big data storage, many tiny tasks (ants), high-performance computing (HPC), highly interactive single person tasks, highly interactive multi-person jobs, single compute intensive jobs, private local tasks, slow communication, real-time local tasks, real-time geographically dispersed tasks, and access control.

The big database creation category is characterized by organizing a large amount of data requiring simple but massive computation efforts. For example, sorting and analyzing statistics for census data. Or another example is the offline keying for search keys of databases. Example suppliers are the US Census Bureau, Yahoo, and Google.

The big database search and access workload category is characterized by repeated interactive requests or queries submitted to a very large database. The customer satisfaction for this category is dependent upon the response time or latency. The key resources that limit the latency are the database server software and organization, the disk storage, the network bandwidth from the storage to the database server, and the server load. Example suppliers are Ancestry.com, the Mormon ancestry database, the US Census Bureau, Yahoo, and Google.

The big data storage workload category is characterized by the integrity of large amounts of data which is periodically updated, usually by small increments in a short period of time, and that occasionally requires a massive download or update. The end user sees this as an archiving or data backup resource. This workload category in the cloud is multiuser as opposed to internal solutions that would have the IT department handle archiving and retrieving. Example suppliers in this category are Rackspace, Softlayer, Livedrive, Zip cloud, Sugarsync, and MyPC. Primary objectives of this category are the integrity and security of the data. The speed (latency, throughput, storage, incremental processing, and complete restoration) is not the highest priority.

The big data transfer workload category is characterized by the large amounts of data that is being moved from one big data storage facility to another. Primary resource limitation is network bandwidth.

The big data calculation workload category is characterized by the relatively straightforward calculations on large amounts of data primary resource limitations which are number of processors, local data bus bandwidth, and big data storage access time. Another key resource for this category is cache utilization.

The in memory database workload category is characterized by the large amounts of data which is rapidly accessed. The end user sees this as a fast database. The objective of this category is primarily the size of the data. The speed (latency, throughput, storage, incremental processing, and complete restoration) is the highest priority.

The HPC workload category is characterized by problems requiring teraflops of computing power. High grid point count matrices of partial differential equations are in this category. The base resource is raw compute power and large memory; however, implementing this with multiple processes adds the resource requirements of network speed and software partitioning.

The highly interactive single person tasks workload category is characterized by a fast task with a single user, such as a video (online or local). The key aspect here is response time latency.

The single computer intensive jobs workload category is characterized by high-speed large single user tasks that have significant user interaction. An example can be digital artists working in a movie studio.

The private local tasks workload category is characterized by traditional single user tasks. Example tasks

Table 1. Characteristic computing resources for workload categories for the cloud [24].

Cloud computing workload category	User view or example providers	Limiting resources
Big streaming data	Netflix	Network bandwidth
Big database creation and calculation	Google, US census	Persistent storage, caching, computational capability
Big database search and access	US census, Google, online shopping and reservations	Persistent storage, network, caching
Big data storage	Rackspace, Softlayer Livedrive, Zip cloud Sugarsync, MyPC	Persistent storage, caching
Big data transfer	Copying and backup	
Big data calculation	Straightforward calculation on large amounts of data.	
In memory database	Redis, SAP	Main memory size, caching
Many tiny tasks (ants)		network, many processors
High performance computing (HPC)	AWS (Amazon), Cyclone^{TM} (SGI)	Processor assignment and computational capability
Highly interactive single person	Terminal access, server administration	Network (latency)
Highly interactive multi-person jobs	Collaborative online environment, e.g. Google Docs	Network (latency)
Single computer intensive jobs	EDA tools (logic simulation, circuit simulation, board layout)	Computational capability
Private local tasks	Offline tasks	Persistent storage
Slow communication	E-mail, blog	Network
Real-time local tasks	Any home security system	Network
Location aware computing	Travel guidance	Local input hardware ports
Real-time geographically dispersed	Remote machinery or vehicle control	Network
Access control	PayPal	Network
Voice or video over IP	Skype, SIP	Network

include word processing, spreadsheets, and data processing. These might be local for convenience or security. In the case of "thin" clients, these could run in the cloud.

The access control workload category is characterized by user-initiated requests where the response is to another server authorizing more activity. The local example is the password on your computer. Most systems rely on passwords and most security experts agree this is a very significant problem [25]. The local example connected to a network (which may be a cloud) is a bank ATM. The cloud example is online purchases. The resources required vary widely with the application and level of security desired. The cloud is inherently open and flexible. A key issue with access control is the conflict with privacy. The various cloud providers need to verify the identity of individuals and systems using and providing services. As Schneier says "Who controls our data controls our lives," [26] and so it is in the cloud. Data for access control include private data. Conglomerations of services (i.e. cloud provider) must present a trusted environment meeting both security and privacy.

Table 1 identifies a wide variety of workload categories with special emphasis on workload categories related to EDA. In these categories, some key resources are identified. Of course, all computing uses many computing resources. However, some categories of jobs use particular resources to a much greater extent than others. This resource allocation affects all of the different groups. For vendors, one wants to meet the customer needs promised in the SLA without overspending on capacity. For the customer, one does not want to purchase more capacity than is needed. For the computer manufacturer, one wants to enhance the hardware or software capability in exactly the way to actually improve the performance due to any bottlenecks. The table includes computing jobs that are not cloud related, for completeness.

5. Silicon design flow steps mapped to workload categories

A typical design flow has dozens of intermediate stages, with up to hundreds of tools generating thousands of design data and log files, requiring millions of CPU hours to complete the design. While actual design flows and tool versions may differ across individual projects, making it hard to share and maintain a consistent design environment, there is a desire to share EDA tool licenses between many design teams to optimize the software cost. An example design flow is shown in Figure 1, and as an illustration, the mapping of some individual steps to the cloud workload categories. Ghosh et al. [27] propose a structured approach for cloud workload analysis and migration aimed at generic web-based workloads, such as enterprise resource planning (ERP) and customer relationships management (CRM) solutions, whereas our approach below goes into the details of silicon design steps to accomplish mapping to cloud workloads.

Instruction set design involves high-level design decisions about the fundamental contract between hardware and software for the target design, the instruction set architecture (ISA), i.e. which machine-level instructions will the target architecture natively support. This is an important step, as all optimizations and design decisions follow from this step. For example, if an architecture does not need to natively support the division operation, the architects do not need to design division units for the

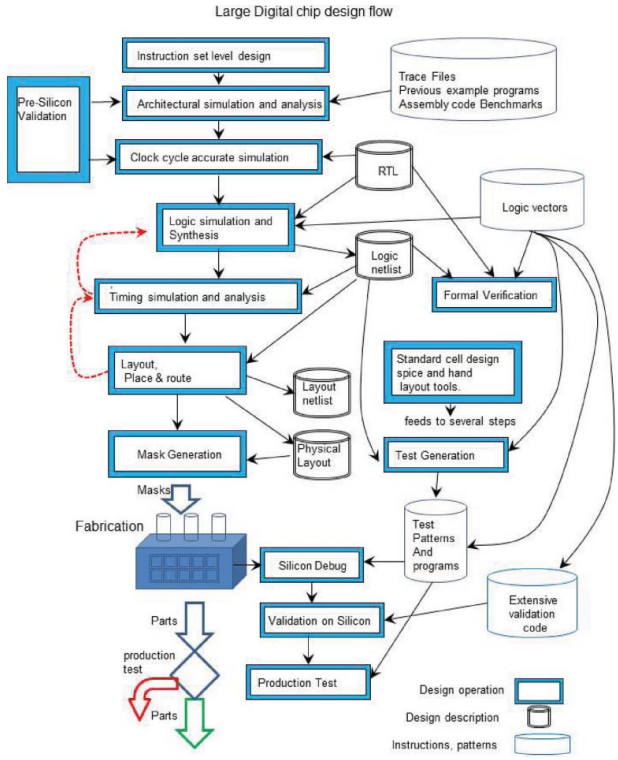

Figure 1: An example chip design flow with several EDA tools.

chip, and can thus use those resources elsewhere or reduce the number of transistors on the chip. If the target architecture needs to process large arrays of floating point elements, performing the same operation on all the elements of an array, the architects may choose to support this directly through the instruction set, by adding such instructions to the ISA. An example of such an addition is the set of streaming SIMD instructions (SSE) that Intel added to the Pentium 3 processor in 1999. Such decisions are made after careful examination of the performance of most popular applications on the existing architectures, and also by simulating the new architecture with new instructions included in the ISA. The latter can be done as part of the architectural simulation and analysis explained below. The former usually involves scripting a number of runs on the existing machines, and automatically gathering the data in a presentable form. Depending on the amount of data being collected, it could involve a moderate number of disk accesses, and depending on the tasks being studied the overall categorization could be compute-heavy, memory-heavy or disk-heavy. The bulk of these tasks maps to the HPC category. However, mapping this to the cloud, one must not only know the exact architecture and micro-architecture version of the machines that run these scripts, but these versions should match the ones that are of interest. In the current cloud computing model, this match-up might be typically difficult to achieve.

Architectural simulation and analysis is the exploration of microarchitectural design parameters and choices such as the size and organization of CPU caches, or how many arithmetic logic units (ALUs) need to operate in parallel to achieve the desired performance, and may also involve area and timing goals for a chip. Design architects often work with an instruction set simulator, to see how some sample applications will behave with the design configuration changes. Different simulators are used for exploring the design parameters at different levels. For example, full-system simulators can run unmodified operating systems and commercial applications while simulating the hardware structures in less detail compared to component or sub-system simulators that might simulate the hardware with the goal of understanding chip and power consumption. Sub-system simulators usually take traces of events as inputs, and do not have the ability to run high-level software directly. Full-system simulation involves the execution of a large, complex program running multiple independent simulations. What distinguishes this from the previous category (instruction set design) is that the simulators can run on machines with different underlying architectures, and the capabilities or configurations of these machines do not affect the simulation results – they will only affect the time it takes to finish the simulations. These tasks are thus well suited to run in the cloud, and map to the HPC category. Sub-system simulators, on the other hand, can be smaller in size and less complex, and can often be run interactively. Thus, these would match the highly interactive single person job category.

Clock cycle accurate simulation refers to the detailed functional and timing simulation of the proposed architectural design at the level of detail required to define the performance within each clock cycle. With both simulation and timing analysis, this checks and sets the boundary conditions for the time constraints on the lower level implementations to perform these functions. This requires massive amounts of simulation to ensure coverage of the various architectural decisions. The appropriate workload category can be either big data calculation or HPC. If the extensive simulations are performed with the results being stored and then the analysis is run, this is in the big data calculation category. On the other hand, if the analysis calculations are run immediately after the simulations, thus skipping storage of large amounts OS simulation data, this falls into the HPC category.

Logic simulation and synthesis refers to the step of detailed logic simulation of the system using the gate level implementation including accurate timing models.

The timing models include feedback from the later step of synthesis and place and route. This includes clock and power tree synthesis. This step requires extensive computer time for both the flattening and compilation of the circuit and the simulation of the circuit. This falls into the big data calculation category.

Timing simulation and analysis is the check for timing violations for the low-level implementation of the design. Depending upon how much of the circuit is analyzed, this can be in either of two categories. When all cases for the whole design is checked, this falls into the workload category of big data calculation. If a small subcircuit of the design is being checked quickly, this is in the highly interactive single person category.

Pre-silicon validation refers to running software OS and associated test content on the model of a chip, before silicon is ready. The idea here is to find any bugs early and avoid the cost of a silicon re-spin. This can be done on an emulation model, derived from register transfer level (RTL) description, or another high-level model description or some combination thereof. This involves downloading the RTL model on a large field programmable gate arrays-based server boxes and running them, say at a speed of several hundreds of MHz. These server boxes are specialized and tend to be very expensive, hence small design teams cannot often afford them. If these boxes are stored in a private or public cloud, then these can be time-shared at an effective cost, thereby helping to reduce the overall product design cycle. This step maps well to HPC in cloud.

Formal verification refers to the process of applying mathematical postulates and theorems to the high-level model or its representative code for ensuring that critical logic bugs are detected early without having to run an exhaustive set of simulation tests. Current generation of chips have many states giving rise to complex combinations of interactions, some of which may result in a deadlock that can be only detected after a long series of events. Such a situation cannot be always anticipated or tested with all available test vectors, but may get exposed in the field usage. Thus, formal verification adds yet another capability to the arsenal of validation team by mathematically ensuring that the flow of events and interactions will be error free. However, it is also computationally challenged and methods for this are still evolving. Formal verification usually works at an IP level, so it maps well to the single computer intensive jobs category.

Standard cell designs are done on local machines using spice and hand layout tools. These tasks require real-time editing of a layout figure by mask designers, or what-if simulations of circuit timing with interactive device size changes. These steps are performed on engineering workstations, or local terminals with graphics displays directly connected to nearby servers. If these servers are moved to be a public cloud, it is likely to introduce higher latencies due to network delays, which may be intolerable to the human designers. This design task maps well to the highly interactive single person category. This is not to imply that a large cell library can be designed by a single person. Large complex standard cell libraries require large teams and a great amount of compute power. However, from computer workload categories perspective that translates to multiple highly interactive single person tasks.

Layout place and route refers to the process of planning, placing, and connecting design objects. Planning is a top-down process, whereas the placement and routing are done bottoms up, with only the rectangular shape of a lower level block and its pins visible at the next level up. Transistors at the lowest level of hierarchy in a standard cell, and then these cells in a higher level block such as an ALU. Then these higher level blocks are placed and routed in yet another abstract level function, such as to implement a CPU or image signal processing unit on the silicon. Such a hierarchical arrangement limits the complexity and number of blocks to be placed and routed to a manageable size in a chip having billions of transistors. Planning is an interactive process which is mapped to highly interactive single person, whereas place and route tools are batch mode, and can be mapped to the category of single computer intensive jobs. Some physical synthesis jobs can take a week or more to complete. A number of such jobs are launched in parallel to run on a server farm, and map well to HPC in cloud.

Mask generation requires some additional processing. With smaller geometries on current silicon designs, there is a need to do some adjustment to account for these dimensions being even smaller than the wavelength of light used for lithography process during chip manufacturing. The geometric steps are collectively called optical proximity correction and are very computationally intensive in nature. Hence, divide and conquer strategy is used to split the final layout in little pieces, which are then sent to a large number of servers for parallel processing, and then integrated back into a final layout database. Hence, mask generation qualifies as an HPC category with big database creation and calculations.

Silicon debug is performed after the chip is manufactured and back. The chip is tested as the power, performance, and even some functional bugs can only be

found with actual silicon instead of with EDA models. Also, the chip after a successful power on can run user applications at full speed vs. simulation models often running at greatly reduced speeds. Thus, the whole HW and SW stack can be tested on an actual chip, exposing some flaws that may require further silicon spins. If better models can be created, then there is a potential for cloud computing to shine without waiting for the silicon to arrive. However, currently, there is no substitute for the post-silicon validation. This is best mapped to a single person interactive task.

The tasks of pre-silicon validation and mask generation clearly fall in the realm of cloud computing, if the IP security can be assured. Clearly, security is a boundary condition for any design IP to go into a public cloud. EDA tasks tend to have elastic computing needs, thus pay as you go scheme works well, and sometimes bringing in more computing to these tasks will tend to finish them sooner, resulting in faster run times and reaping economic benefits of cloud computing.

Production test is the application of a fixed test program to each chip manufactured in order to identify defective chips and discard them. The creation of the test program for production test requires significant computation for fault simulation and test vector generation. The fault simulation and test generation can be split into several high computation parallel jobs. This falls in the category of big data calculation.

There are many types of workloads that can be run in the cloud. The formal definition of cloud computing provided by National Institute of Standards and Technology (NIST) is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The NIST cloud model is composed of five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service), three service models [software as a Service (SaaS), platform as a Service (PaaS), infrastructure as a Service (IaaS)], and four deployment models (private cloud, public cloud, community cloud, hybrid cloud) [28].

Big data is a relative term. Large databases in the EDA industry are small compared to some in other industries. Of the three categories of big data (big data storage, big data transfer, and big data calculation), EDA projects are most dependent upon big data transfer – specifically, getting the large design database from the customer into the cloud and then getting the modified big data back to the customer.

Some workload categories are, and hence some steps in the design flow, best performed on a local computing resource such as a desktop workstation. Other workload categories benefit from the availability of large amounts of computing capacity, whether calculation or storage. Because the different tools utilized at the different steps of the design flow fall into different workload categories, the decision on what resources should be allocated is different during the different steps of the design process. Therefore, the decision to utilize cloud computing for the EDA design flow is more appropriately made at the individual steps rather than for the whole process. An interesting example is data mirroring for a multi-site design team. For some steps in the design flow, data mirroring maps to the big data storage category. For other steps in the design flow, data mirroring maps to the highly interactive multi-person jobs. The allocation of resources when needed and released when not needed is exactly what cloud computing solves, and is a problem for a multi-category process such as the EDA design flow. In summary, a one size fits all answer to the EDA design tool move to the cloud is replaced by a tool-by-tool or step-by-step decision based upon workload category and customer needs and requirements. The mapping of the workload categories provides a basis for some of the broad adoption issues, concerns, and considerations described in the next sections.

6. Other considerations for adoption of EDA in cloud

Beyond the workload category mapping as described in the previous section, there are some broad considerations that the EDA industry must address to enable a migration to the cloud. The considerations for EDA moving to the cloud include tool licensing, information security, cloud provider drive, past attempts to use the cloud, tool delivery mechanism, and customer demand.

First, consider the licensing of EDA tools, which often prohibits sharing a tool between different geographical sites of the same company. An EDA tool at run-time must renew its token from a central server at regular time intervals. A cloud by definition has distributed computing, this is a hindrance to share the same EDA tool across different servers or data-centers within a logical cloud. The current EDA licensing model can be in harmony with the cloud computing paradigm of pay as you use when the EDA industry players address the appropriate licensing mechanism [29].

A second consideration is that information security in the cloud must be solved to the satisfaction of all stakeholders in the EDA industry [30]. Information security can be viewed as including three functions: access control, secure communications, and protection of private data. Access control includes both the initial entrance by a participant and the reentry of that participant or the access of additional participants. The secure communication includes any transfer of information among any of the participants. The protection of private data includes storage devices, processing units, and even cache memory. These and more information security issues especially related to the cloud are described in [19]. An example security problem is denial of service attacks. Ficco and Rak [31] point out the need for cloud providers to monitor, detect, and provide host-based countermeasures. From the user viewpoint, security is an important part of the problem. For example, Ghosh et al. propose a framework to allow a customer to evaluate cloud providers using several factors, including different security issues [27]. The EDA industry does not in and of itself pose new information security issues [32]. However, the cloud suppliers and the EDA vendors must work together with the EDA user community to find a solution that meets the needs of all stakeholders.

A third consideration for EDA vendors is whether the cloud suppliers even want EDA tools in the cloud [33]. Both Amazon and Rackspace have publicly stated a desire to attract HPC workloads in their data-center, to showcase their prowess and increase revenue. The list of cloud service providers (CSPs) that may be potentially interested in EDA is fairly long [34] and spread across the world. Their services range from SaaS, PaaS , and IaaS, including hosting customers' servers in CSP owned data-centers. Of this list, the most relevant is IaaS because a typical EDA company and its users will come with their own software and data, and need physical or virtual servers to host them. Figure 2 shows how extra compute resources can reduce time-to-market.

A fourth consideration is that moving to cloud has been attempted in recent years. Several large EDA companies have already experimented with offering cloud services using their own data-centers, with mixed responses from customers. Synopsys is an example with a verification-on-demand solution using Verilog compiler simulator (VCS), which provides massively scalable verification via Amazon Web Services. According to Synopsis website, VCS [35] is the first EDA product to become available on a public cloud. While this does not demonstrate that EDA in the cloud is a success or failure, it does show that the work has started.

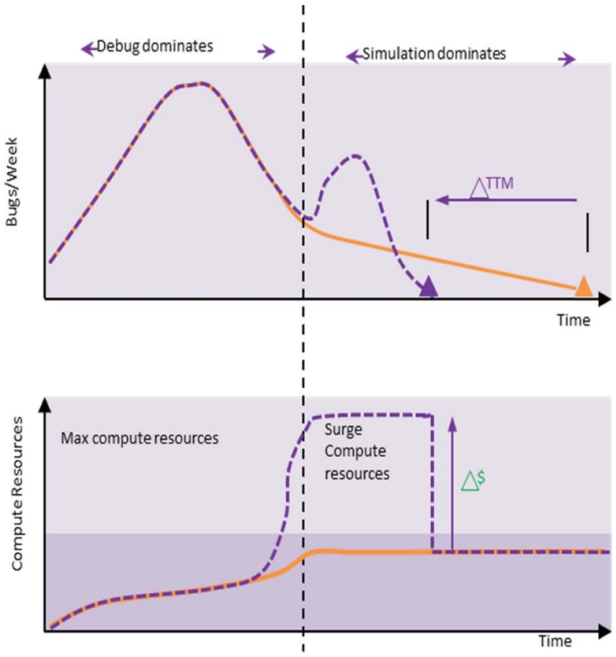

Figure 2: Design teams can reduce time-to-market by adding "surge" computing resources [35].

A fifth consideration is how the EDA tools are offered to the end users. As it often happens in the software industry, some companies are aggregators of others' tools. This model may work well for EDA in the cloud. Transim Technology Corporation hosts several cloud-based engineering applications [36]. Productrinica is the company that launched the cloud testing service aimed at design verification engineers, fabless company and any institutions with limited resources [37]. The service is a pay per use approach applied to IC prototype testing, debugging, and trouble shooting.

A sixth consideration is the EDA customers' demands. Cadence and a startup Nimbic (purchased by Mentor) found limited customer demand for moving to the cloud [38]. However, Synopsys recognizes the demand for EDA tools in the cloud [35]. A model which may satisfy EDA customers' demand is a hybrid one, with some of the EDA tools being in the cloud and others in the captive data-centers. As Figure 3 shows, 60% of participants in a recent Design Automation Conference survey predicted that EDA licensing will move to the Cloud model in the near future.

A seventh consideration is that the current EDA tools and flows were not designed with cloud computing in mind. Therefore, to fully utilize cloud computing, these may need to be redesigned. Potential benefits of redesigning the EDA tools for cloud may include an order of magnitude improvement. This clearly requires further

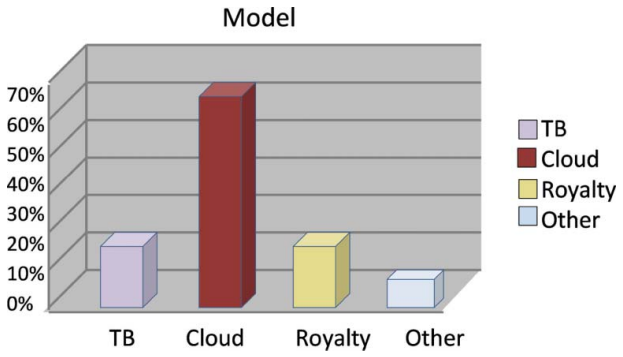

Figure 3: EDA tool licensing model [TB: TimeBased, Cloud, Royalty (on IC production) or other (mostly combination)] [30].

study and experiments on adopting EDA algorithms for cloud computing [39].

Our paper has concentrated on the case study of the EDA industry adoption of cloud computing. An excellent analysis of other issues for adoption of cloud computing by small- and medium-sized enterprises described that the larger barriers to adoption are non-technical [40]. First and foremost, it is worth noting that they found that most companies see the benefits of adopting cloud computing and most are interested in the adoption. The top seven reasons that companies gave for not adopting nor intending to adopt cloud computing solutions were security (including data loss), loss of control of data and infrastructure to manage data, lack of ability to measure cost-benefit analysis, availability and quality of service, data lock-in (inability to change providers), and data privacy and associated legal requirements. Their paper emphasizes that in some cases, such as security, adoption of cloud computing has actually improved the situation. This is particularly important for security because security is listed as a top concern both in the study and in the EDA industry. Their paper points out that non-technical key issues must be addressed to favor cloud computing development. Our paper proposes that the cloud computing adoption decisions should be made on a case-by-case basis for different categories of computing. We illustrate this through a case study of the EDA industry.

We have described broad considerations that must be addressed in order to take advantage of opportunities cloud computing might provide to the EDA industry. These considerations include licensing, security, cloud providers, past EDA attempts to move to the cloud, delivery mechanism, and customer demand. The approach to meeting these considerations may vary with different steps (hence workload categories) in the design process or with different vendors, customers, and cloud providers.

7. Summary

The EDA industry software, tool methodologies, flows, and scripts for VLSI design have evolved from individual programs on mainframe computers, through collections of tools on engineering workstations, to complete suites of tools with associated methodologies on networks of computers. EDA has had a significant impact on circuit and hardware design, which in turn has contributed to the vast progress in the field of computing. DA is one of the reasons why computer chips with upwards of billions of transistors can be designed, and one can say that the server farms that form the back end of the cloud would not have been around without the EDA industry. Thus, it is interesting to see whether cloud computing can, in turn, facilitate future growth of EDA and the EDA industry.

While cloud computing is often thought of as a monolithic entity, it comes in many flavors, and is comprised of several subsystems and services. Similarly, EDA tools are sometimes thought of as part of a system built with each tool having similar attributes. Based upon a previous categorization of cloud computing workloads, this paper maps the sub-tasks of an example silicon design flow to types of workloads. The mapping of workloads is applicable to both private and public cloud computing. This mapping can serve as an example for EDA companies and hardware design firms as they look to explore the cloud for hardware design tasks. Our method can potentially open new doors and customer bases for enabling EDA growth. This paper also provides examples of some early adopters, the issues they faced, and new emerging challenges, whether real or perceived [41]. Additionally, some considerations are mentioned, such as licensing and delivery mechanisms that go beyond the mapping of tasks to workloads. The major contribution of this paper is a proposed method for mapping EDA tools to cloud computing categories to facilitate the decision of which EDA tools are candidates for moving to the cloud.

Acknowledgements

We are thankful for the support of Intel's design technology experts to review and approve this paper for external publication.

Disclosure statement

No potential conflitct of interest was reported by the authors.

References

[1] J. K. Ousterhout, G. T. Hamachi, R. N. Mayo, W. S. Scott, and G. S. Taylor, "Magic: A VLSI layout system," in *21st Design Automation Conference*, Albuquerque, USA, 1984, pp. 152–9.

[2] L. W. Nagel, *SPICE2: A Computer Program to Simulate Semiconductor Circuits*. Berkeley, CA: University of California, 1975, pp. 4–5.

[3] S. Beebe, F. Rotella, Z. Sahul, D. Yergeau, G. McKenna, L. So, Z. Yu, K. C Wu, E. Kan, J. McVittie, and R. W. Dutton, "Next generation stanford TCAD—PISCES 2ET and SUPREM OO7," in *IEDM 1994 Proceedings*, San Francisco, CA, 1994, pp. 213–6.

[4] M. R. Pinto, C. S. Rafferty, and R. W. Dutton, "PISCES-II - poisson and continuity equation solver," Stanford Electronics Laboratory Technical Report, Stanford University, Stanford, CA, 1984.

[5] J. M. Acken, and J. D. Stauffer, "Part 1: Logic circuit simulation," *IEEE Circuits Syst. Mag.*, Mar. 1979, pp. 9–14.

[6] J. M. Acken, and J. D. Stauffer, "Part 2: Logic circuit simulation," *IEEE Circuits Syst. Mag.*, Vol. 1, no. 2, pp. 3–12, Jun. 1979.

[7] S. A. Szygenda, "TEGAS-anatomy of a general purpose test generation and simulation at the gate and functional level," in *Proceedings of the 9th Design Automation Conference*, Las Vegas, NV, 1972, pp. 116–27.

[8] R. E. Bryant, "MOSSIM: A switch level simulator for MOS LSI," in *Proceedings of the 18th Design Automation Conference*, Nashville, TN, 1981, pp. 354–61.

[9] L. H. Goldstein, "Controllability/observability analysis of digital circuits," *IEEE Trans. Circuits Syst.*, Vol. 26, no. 2, pp. 685–93, 1979.

[10] J. Grason, "TMEAS, a testability measurement program," in *Proceedings of the 16th Design Automation Conference*, San Diego, CA, 1979, pp. 156–61.

[11] G. D. Hachtel, L. Hemanchandra, R. Newton, and A. Sangiovanni- Vincentelli, "A comparison of logic minimization strategies using ESPRESSO: An APL program package for partitioned logic minimization," in *Proceedings of the International Symposium on Circuits and Systems*, Rome, 1982, pp. 42–8.

[12] B. Preas, and C. W. Gwyn, "Methods for hierarchical automatic layout of custom LSI circuit masks," in *Proceedings of DAC*, Las Vegas, NV, 1978, pp. 206–12.

[13] C. Sechen, and A. Sangiovanni, "Timberwolf 3.2: A new standard cell placement and global routing package," in *23rd DAC*, Las Vegas, NV, 1986, pp. 432–9.

[14] Available: http://www.edac.org/initiatives/committees/mss

[15] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research: A systematic review," *IEEE Trans. Cloud Comput.* Vol. 1, no. 2, pp. 142–57, Jul.–Dec. 2013.

[16] H. Guan, R. Ma, and J. Li, "Workload-aware credit scheduler for improving network I/O performance in virtualization environment," *IEEE Trans. Cloud Comput.*, Vol. 2, no. 2, pp. 130–42, Apr.–Jun. 2014.

[17] I. S. Moreno, P. Garraghan, P. Townend, and J. Xu, "Analysis, modeling and simulation of workload patterns in a large-scale utility cloud," *IEEE Trans. Cloud Comput.*, Vol. 2, no. 2, pp. 208–21, Apr–Jun. 2014.

[18] W. D. Mulia, N. Sehgal, S. Sohoni, J. M. Acken, C. Lucas Stanberry, and David J. Fritz, "Cloud workload characterization," *IETE Tech. Rev.*, Vol. 30, no. 5, pp. 382–97, 2013.

[19] N. K. Sehgal, S. Sohoni, Y. Xiong, D. Fritz, W. Mulia, and J. M. Acken, "A cross section of the issues and research activities related to both information security and cloud computing," *IETE Tech. Rev.*, Vol. 28, no. 4, pp. 279–91, 2011.

[20] Available: http://www.eetimes.com/document.asp?doc_id=1326801

[21] W. Joyner, "EDA: The first 25 years," presented at the 50th Design Automation Conference, Jun. 2013. Available: www.dac.com

[22] N. Sehgal, C. Y. R. Chen, and J. M. Acken, "An object-oriented cell library manager," in *Proceedings of ICCAD*, San Jose, CA, 1994, pp. 750–3.

[23] M. E. Porter, *The Five Competitive Forces that Shape Strategy, Harvard Business Review*. 2008. Available: http://hbr.org/2008/01/the-five-competitive-forces-that-shape-strategy

[24] A. Kuehlmann, R. Camposano, J. Colgan, J. Chilton, S. George, R. Griffith, … D. Singh. "Does IC design have a future in the clouds?" in *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, Anaheim, CA, 2010, pp. 412–4.

[25] R. Rutenbar, "EDA: The second 25 years," presented at the 50th Design Automation Conference, Jun. 2013. Available: www.dac.com

[26] B. Schneier, "The hidden battles to collect your data and control your world," Mar. 2015, Available: https://www.schneier.com/book-dg.html

[27] N. Ghosh, S. K. Ghosh, and S. K. Das, "SelCSP: A framework to facilitate selection of cloud service providers" *IEEE Trans. Cloud Comput.*, Vol. 3, no. 1, pp. 66–79, Jan–Mar. 2015.

[28] NIST Special Publication 800-145. Sept. 2011. Available: http://www.nist.gov/itl/csd/cloud-102511.cfm

[29] J. Ralph, "Challenges in cloud computing for EDA," *Chip Des. Mag.*, Available: http://chipdesignmag.com/display.php?articleId=4492

[30] L. Stok, "EDA: The next 25 years," presented at the 50th Design Automation Conference, Jun. 2013. Available: www.dac.com

[31] M. Ficco, and M. Rak, "Stealthy denial of service strategy in cloud computing," *IEEE Trans. Cloud Comput.*, Vol. 3, no. 1, Jan.–Mar.2015.

[32] R. Brayton, and J. Cong, "NSF workshop on EDA: Past, present and future," *IEEE Des. Test Comput.*, Vol. 27, no. 3, pp. 68–74, May/Jun. 2010.

[33] *EDAC Forecast Meeting*, San Jose, Mar. 14, 2013. Available: http://www.edac.org/events/2013-EDA-Consortium-Annual-CEO-Forecastand-Industry-Vision/video

[34] Available: http://talkincloud.com/tc100

[35] Available: http://www.synopsys.com/Company/Publications/SynopsysInsight/Pages/Art6-Clouds-IssQ2-11.aspx

[36] "Designing in the Cloud," *Electronic Specifier*, 7th November 2012. Available: http://www.electronicspecifier.com/design-automation/designing-in-the-cloud-

arrow-dr-uwe-knorr-transim-technology-es-design-magazine June 2015.
[37] M. Elliott, "Productonica: Cloud clears way to IC testing," *Electron. Specifier*, Nov. 2013. Available: http://www.electronicspecifier.com/test-and-measurement/advantestcx1000d-productronica-cloud-clears-way-to-ic-testing.
[38] Available: http://finance.yahoo.com/news/mentor-graphics-acquires-nimbic-inc-175400528.html
[39] C. Schwaderer, "EDA and the cloud," *Embedded Comput. Des.*, May 2014. Available: http://embedded-computing.com/articles/eda-the-cloud/
[40] S. Trigueros-Preciado, D. Perez-Gonzalez, and P. Solana-Gonzalez, "Cloud computing in industrial SMEs: identification of the barriers to its adoption and effects of its application," *Electron. Mark.*, Vol. 23, no. 2, pp. 105−14, Jan. 2013.
[41] Available: http://www.techradar.com/us/news/internet/cloud-services/security-is-a-red-herring-in-the-cloud-1163167

Authors

Naresh Sehgal is a senior program manager in Platform Engineering Group at Intel Corp. He has been with Intel for 27 years in various roles, including EDA development, Silicon design automation, Intel-HP alliance management and for launching virtualization technology on all Intel platforms. He holds a PhD in computer engineering from Syracuse University, and MBA from Santa Clara University. He holds 5 patents and authored more than 30 publications in the CAD domain.

E-mail: nareshksehgal@gmail.com

John M. Acken is a faculty member in the Electrical and Computer Engineering Department, Portland State University, Portland, OR. He received his BS and MS in electrical engineering from Oklahoma State University and his PhD in electrical engineering from Stanford University. Dr. Acken's projects include technology and devices for information security and identity authentication. He is a co-inventor on a patent titled: "Conditional Access and Content Security Method", US Patent Number 6,069,647, granted 30 May 2000. He has worked as an electrical engineer and manager at several companies, including the US Army, Sandia National Labs in Albuquerque, New Mexico and Intel in Santa Clara, CA. John's time in the US Army was in the Army Security Agency, which was the Army Branch of NSA during the Vietnam War.

E-mail: acken@pdx.edu

Sohum Sohoni is an assistant professor in the School of Computing Informatics and Decision Systems Engineering, Arizona State University, Mesa, AZ. He received his PhD in computer engineering from the University of Cincinnati in 2004 and his bachelors in electrical engineering from COEP, Pune University in 1998. Dr. Sohoni's research interests are broadly in the area of computer architecture and performance analysis. His research is in the area of cache memory performance of memory-intensive applications. He has recently expanded his research interests to include engineering education, and has published his work at ASEE's national conference and ASEE's Midwest section conference. He advises several undergraduate, MS, and PhD students.

E-mail: sohum.sohoni@asu.edu