Big Data on a Shoestring

Darshan Thaker, WIN
Huan Jin, WIN
Srikanth Kandukuri, WIN

Abstract – Panamera, the next generation WIN Broadband Plasma (BBP) tool, has a 30X increase in data requirements for a critical NanoPoint (NP) feature. Additionally, due to the strict time-to-market and cost constraints of the Panamera program, the traditional avenues of developing new hardware/systems were closed to us. To overcome this challenge, we redesigned the data flow through Software and Algo and maximized the reuse of existing hardware resources. Specifically, we developed innovative software techniques to manage data transfer, borrowed existing ideas from the EDA industry to reduce data size and judiciously broke the low-coupling rule, to aggressively optimize data lifecycle.

I. Introduction

The Pixel to Design Align (PDA) feature, a critical component of NanoPoint software on WIN tools, aligns wafer images to the design during defect detection. Design data computation is performed during a setup phase, called clip extraction, in units called 'clips' and the performance metric is clips/sec. Current generation WIN tools can achieve a performance of 140 clips/sec; driven by Algo requirements (which are beyond the scope of this paper), clip extraction performance for next generation PDA needs to be increased to approx. 4400 clips/sec while the budgeted time to complete PDA Setup remains unchanged. [1] [2]

This project had strict time-to-market and RPC (raw parts cost) constraints which meant we could neither develop/design new hardware nor could we write software from scratch. Consequently, we addressed this problem by redesigning the data flow through software and algo and by adding a few off-the-shelf components to the Image Computer Cluster (IMC) to extract required performance from already existing hardware.

Clip extraction is performed by the DbbService which runs on a single server with 32 cores, 256 GB of RAM and a large RAID array providing 30 TB of storage. Fortunately, WIN BBP tools have an Image Computer Cluster (IMC) with 32 servers each with 32 cores; also the PDA Setup step is not performed in parallel with defect detection, so we could use the available compute resources on the IMC for clip extraction. The budgeted time for completing PDA Setup is 10 minutes [1]. The primary problem is thus transformed into the following problems: A) Reduce design data size to manageable amounts B) Rapid transfer of design data onto the IMC and C) Minimize the lifecycle of extracted design clips. [3]

The rest of this paper gives a general overview of our solutions to the three problems mentioned above.

II. Solutions

Table 1 gives an idea of how large design data can be and how its size compares with the image data that forms the bread-and-butter for WIN BBP tools. The first part of our solution therefore addresses the problem of design data size.

Use Case	Data Size	Max. Time	Transfer Mechanism
Defect Detection	Full wafer image @ 50nm px. 29 TeraBytes	30 min	Dedicated and specialized HW
Next Gen PDA Setup	Full die design. 20 TeraBytes	10 min	Software with off-the-shelf HW

Table 1: Consider the standard WIN use case of defect detection, where massive amounts of data are transferred by specialized hardware. For PDA Setup, we will have to work with similar amounts of data, in far less time, without developing new hardware. This table details the similarities and differences between these use-cases.

A. *Compression and Hierarchy*

Very large design data files are an on-going problem in the EDA industry and standard techniques have been developed to manage them. This is the Oasis format [4] which incorporates a combination of compression and a hierarchical data format to dramatically reduce the storage needs of design data. We incorporated the Oasis format in our software which gave us orders of magnitude reduction in design file sizes, as shown in Table 2.

Customer	Optimized Design Data Size
Customer-A	22 GB
Customer-B	35 GB
Customer-Ba	31 GB
Customer-C	56 GB
Customer-Ca	33 GB
Customer-D	32 GB

Table 2: Design data sizes at various customer sites after Oasis format optimization

When compared to image data generated by WIN tools, there are two main reasons that Oasis format yields such great data size reduction. The first is that from the perspective of WIN tools, design data is invariant to any operation we perform; therefore we simply consider it immutable and proceed accordingly. Secondly, a single use-case will consume at-most one/twelfth of design data. So its default state can be a highly compressed, hierarchical format; we extract and consume only what is necessary.

B. *Cascading Pipelines and a Hypercube*

The overall requirement is that PDA Setup complete in 10 minutes. To achieve the best possible performance on the IMC, the design data should be transferred into memory of each IMC node before computation (extracting design clips) can begin. The budget for this transfer is 3 minutes. A further constraint is that design data, for IP reasons, has to reside on a server called the DbbIMC. We cannot make copies of it on any IMC disk. After optimization for data size, we are left with the following problem: transfer 64GB of data from the disk on DbbIMC to memory of 16 nodes in the IMC within 3 minutes.

The RAID configured hard drives on the DbbIMC have a read speed of 0.4 GB/sec for small fragmented files and a maximum read speed of 1 GB/sec for large sequential reads. This could be increased by using a large RAID with more disks, but as mentioned earlier, we were constrained by cost. There exists an InfiniBand network on the IMC that can allow data transfers of 1.4 GB/sec using the standard TCP/IP stack and IPoIB. If we spent extra time and wrote additional infrastructure software that used RDMA (Remote Direct Memory Access), we could increase the IB bandwidth to 3.5 GB/sec. This was also not a viable option. Therefore, we developed two different techniques; the first is the Cascading Pipelines when we were constrained by slow disk speed due to small fragmented files. This technique, elaborated in **Figure 1**, uses NFS mounts on IMC nodes in a cascading fashion and the size of the pipeline is approximately equal to the number of files to be copied; effectively this simulates a broadcast over InfiniBand.

The second technique is the Hypercube, based on backplane topology of supercomputers of the 1980s [5] to transfer large files 20GB or more each. As detailed in **Figure 2**, the hypercube results in a geometric progression. Assuming each node can communicate with six neighbors, all required data can be copied to 16 IMC nodes in 4 steps. This technique is ideally suited to large sequential files such as tar archives.

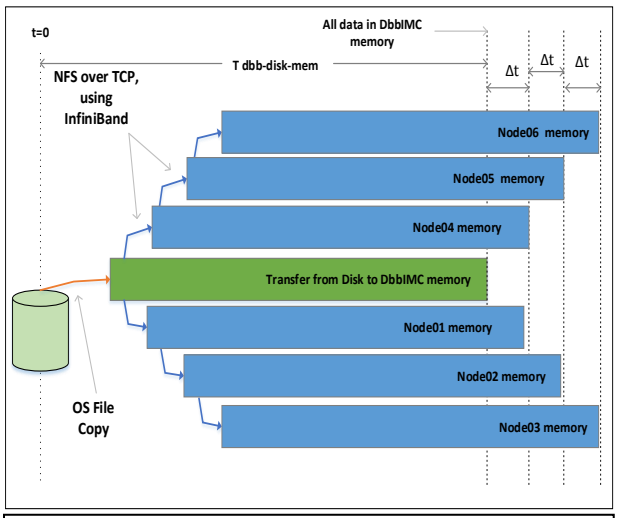

Figure 1: The Cascading Pipeline technique for copying a large number of small files (approx. 20MB per file). There are cascading NFS mounts between machines and the overall performance is bound by the smaller of the disk read speed on the DbbIMC or twice the InfiniBand transfer speed. This is because each node is performing simultaneous reads and writes over IB, thus requiring twice the bandwidth. For reasons of space and clarity, only 6 IMC nodes are shown in this figure.

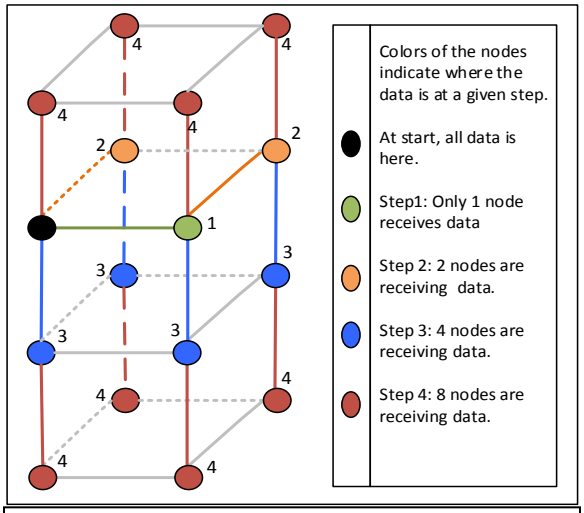

Figure 2: The Hypercube for transferring large files from a single source while making maximum use of network resources. Data is transferred in a geometric progression, if there are **N** nodes that need data, the transfer can be completed in '**x**' steps such that $\mathbf{2^x = N}$. The 16 nodes and their edges are colored based on the step during which they receive data; As soon as a node receives data, it participates as a sender in the next step. This technique requires that all nodes have equal network bandwidth to each other; the type of network is immaterial.

C. *Data Lifecycle*

For any piece of data, an object, a stream or a variable, we define its lifecycle as the time between when it was first created until the time it is no longer in use and therefore can be cleaned up. While this is a simple concept; the longer the piece of data exists, the greater the complexity in managing it. Particularly, if data is allowed to move between machines, to grow, to shrink and allowed to evolve. Therefore, when dealing with the 30X increase in data that is a consequence of next generation PDA, minimizing the data lifecycle is an essential and non-trivial part of the design process.

The current PDA design practices good data encapsulation and low-coupling between different components. While this has served us well so far, as Figure 3A shows, it has the drawback of long data lifecycles and requires data to move through transformations that have a measurable overhead. For next generation PDA, if we maintained the same data lifecycle, the overhead of managing it would overwhelm the system and render meaningless all other enhancements that have been discussed.

We chose to break a system engineering rule of thumb and increased coupling. We did this with much thought and only between Algo and DbbLite (a dynamically linked clip extraction library). As a result, most design data will be consumed in close proximity, spatially and temporally, to where it was produced. Figure 3B shows the design that is a result of this choice.

III. Conclusion and Continuing Work

We have demonstrated that careful and innovative software design choices can enable large increases in data requirements without developing new hardware. The ideas expressed in this paper are not specific to WIN tools but should find applicability in other KT products. Next generation PDA is still under development and it will be some months before it is released to our customers; our current design choices have given us a platform for further development.

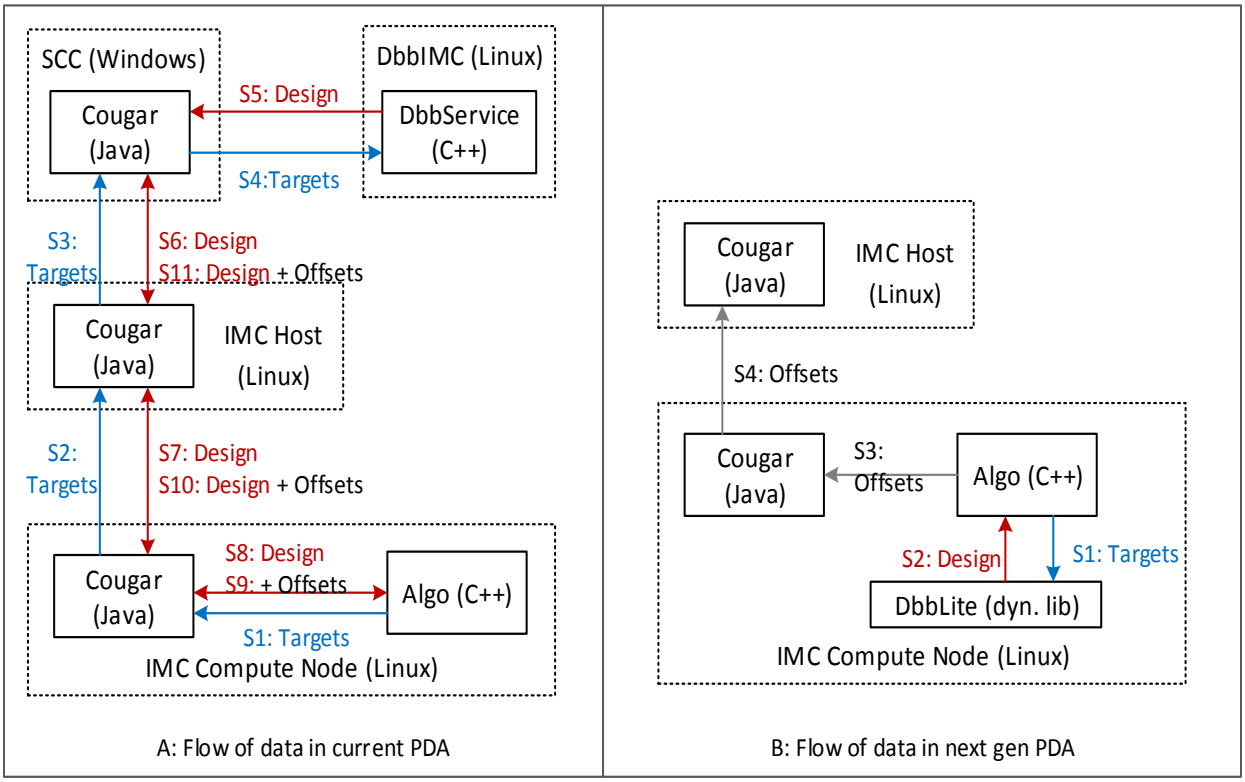

Figure 3: For PDA, the figure shows the relevant software components and the physical machines on which they reside in rectangles with solid lines and dotted lines respectively. The fundamental flow is as follows: Algo generates targets, DbbService extracts design clips for those targets and Algo then renders the design and produces offsets. Software manages the flow and packaging of data. In the figure, S1 is the first step which carries targets as data and then the flow continues as described earlier. Design data is shown with red arrows as its size is large. There is additional cost when data is translated between Java and C++.

Subfigure B performs the same work as subfigure A, but data production and consumption is as within close proximity. This greatly reduces the overhead of managing large quantities of data.

Acknowledgment

This work would not have been possible without the advice and guidance of Chetana Bhaskar, Lisheng Gao, Ashok Mathew and Anil Raman. We are also grateful to the tireless efforts of Santosh Bhattacharya and Pavan Kumar in ensuring the success of this project.

Bibliography

[1] D. Thaker and C. Bhaskar, "PDA Model for IMC," KT Document Center ID: KTDC-1103-605, 2015.

[2] S. Bhattacharya and L. Gao, "PDA Capacity Model," KT Document Center ID: KTDC-1103-320.

[3] D. Thaker, "SubPixel PDA RAD for Panamera," KT Document Center ID: KTDC-1103-467, 2015.

[4] SEMI, "SEMI P44-0211 - Specification for Open Artwork System Interchange Standard (OASIS ®) Specific to Mask Tools," 2010.

[5] Wikpedia, "Hypercube Graph," Wikipedia, 2015.