WHITE PAPER

Optimize Physical Verification Cost of Ownership with Elastic CPU Management

Authors

Chris Grossmann
Principal Engineer,
Synopsys Inc.

Manoz Palaparthi
Product Manager,
Synopsys Inc.

Introduction and Motivation

For physical verification, advanced process technology nodes create implementation challenges. Design sizes have gotten larger and required rules from foundries have become more numerous in count (thousands) and more complex (hundreds of discrete steps). For these reasons, physical verification tools have been able to span these jobs not only across multiple CPUs on a single physical compute host, but also across multiple networked hosts. A DRC or LVS job for today's leading-edge designs, with billions of transistors, can run for multiple days utilizing many hundreds of CPU cores. Physical verification tools are scalable—adding more resources might speed up the run. However, it's also possible that the job has some serial dependencies and cannot run faster. If this is the case, adding more compute resources will NOT speed up the run and will instead waste resources.

Any wastage of compute resources results in real loss to the user. In a cloud environment, users can end up paying by-minute rates for resources that are not used. For an on-premise environment, the resources that are occupied and idle represent resources that other users cannot use. Idle resources take up licenses that could be used for other jobs. Excessive run time also causes loss for the user. The time that is spent waiting for results is time that cannot be spent debugging those results. Therefore, users always want their runs to finish as fast as reasonably possible. However, the details of the causes and cures to the scheduling complexity of work items is not available to most users. They would require deep knowledge of not only the foundry rules and the way the rules are implemented in the runset but also the complicated way in which those rules are going to interact with the data. In other words, it is practically impossible for the user to predict how many resources will be needed to optimize both run time and resource usage.

IC Validator Elastic CPU Management

Synopsys IC Validator™ is a modern physical verification tool, architected for massive and efficient distributed processing. IC Validator recognizes the dependency graph, the current resources and the command queue. It uses this information to identify when adding more compute resources will make the run finish faster. It also uses this information to release resources that are no longer needed. IC Validator's dynamic elastic CPU management capability can automatically add and release CPU resources to a DRC or LVS job on the fly depending on the job requirements. The elastic CPU feature seamlessly works with LSF, SGE, and other popular job queuing systems. It can be used in different types of compute networks, including on-prem and cloud.

User Benefits

The IC Validator elastic CPU management technology delivers significant value in the design flow both in resource/cost optimization and in accelerating design closure to meet tape-out schedules.

- **Efficient resource management before tape-out:** Full chip implementation and signoff before the tape-out is an intensive phase of the design flow, where multiple design steps such as ECOs, timing signoff and physical verification are run in parallel. These steps are competing for a limited set of resources. Using IC Validator elastic CPU, designers can release unused or under-utilized CPUs from a physical verification job and re-deploy them for other jobs. This greatly accelerates the design closure time and enhances the ability to meet the tape-out schedule.
- **Lower cost of ownership on the cloud:** An IC Validator elastic CPU job uses up to 40% fewer compute resources, while maintaining similar performance versus a traditional DRC or LVS job. The compute savings directly translate to cost savings on the cloud, where users are billed per hour for compute and storage.
- **Start jobs early and finish early:** Requests to batch systems that ask for more resources can be slower to launch since they require the batch system to acquire all the hosts. IC Validator can start immediately with the minimum resources, with more resources being added as needed. As a result, a job could finish sooner, since it was able to start running sooner.

Consider Figure 1. With the design and run set fixed, the run is performed in several different configurations as described below.

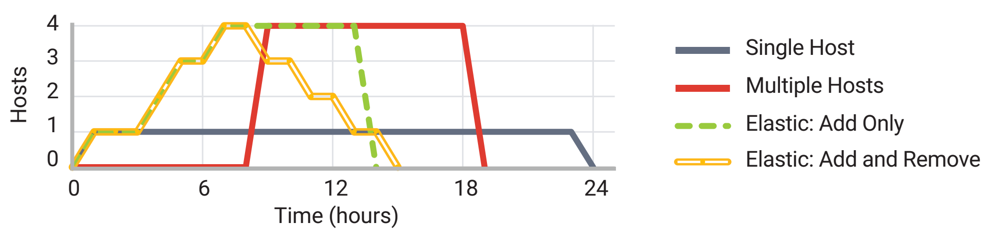

Figure 1: Run Configurations

- **Single Host:** A single host job is relatively easy to launch, since obtaining a single machine is easy. But the turn-around time of this type of job is very long
- **Multiple Hosts:** This type of request is harder to fulfill since the probability that all the machines are available at the same time is very low. This type of job wastes a lot of time waiting for the request to be fulfilled. The turn-around time of the actual job is the best, but the waiting time is variable and can be extremely long, which negates the fast turn-around time.
- **Elastic CPU Add Only:** The turn-around time of this job is comparable to that of the job with multiple hosts, and the waiting time is greatly reduced since each host is added as it becomes available. But the total resource utilization (visualized as the area under the curve) is high and comparable to the multiple hosts job. The tool does not use all the cores throughout the job, and these cores could have been used by some other user.
- **Elastic CPU Add and Remove:** The turn-around time of this job is comparable to that of the previous two cases. But its total resource utilization is much reduced since it can remove hosts from the job when they are not needed.

Results on Real Designs

Figure 2 and Table 1 show the results of IC Validator elastic CPU for DRC checking of a large 7nm design. This DRC job uses 1200 CPUs. The job has some serial dependencies, and during later stages, many allocated CPUs are left idle. IC Validator recognized the job dependencies on the fly and automatically added CPUs when needed and released CPUs when not utilized. The result is that the elastic CPU job finished at the same time and used significantly lower (40%) CPU hours, saving 40% compute costs on the cloud.

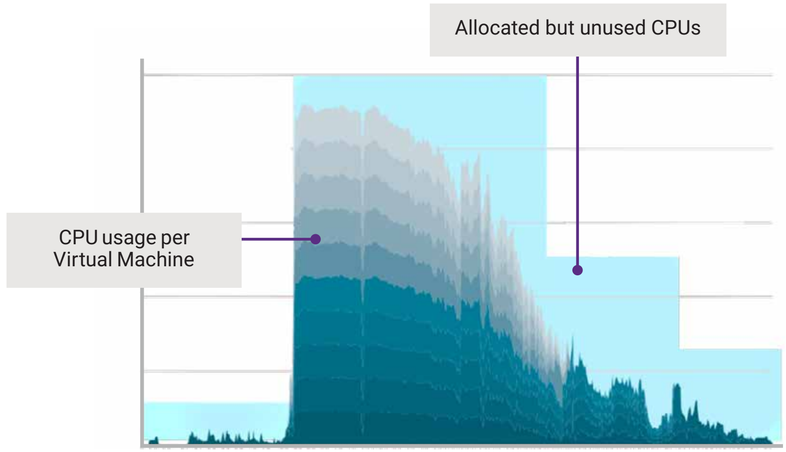

Figure 2: CPU Utilization of Elastic CPU DRC Job

	CPUs	Runtime (hours)	Allocated CPU Hours
Traditional	1200	11.5	13,718
Elastic	1200 peak	12	**8,198**

Table 1: Elastic CPU DRC Job—Runtime and CPU Hours

Shown below in Figure 3 and Table 2 are the results of IC Validator elastic CPU for LVS checking of a 5nm design. This particular LVS job has several serial dependencies and IC Validator recognized these dependencies and automatically added released CPUs based on the job demands. In the end, the elastic CPU job was completed with only 5% additional runtime, while using 50% few CPU hours.

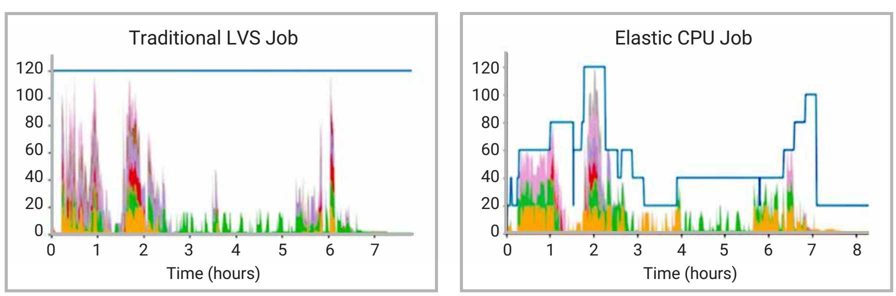

Figure 3: CPU Utilization of Elastic CPU LVS Job

	CPUs	Runtime (hours)	Allocated CPU Hours
Traditional	120	8	943
Elastic	120 peak	8.4	**410**

Table 2: Elastic CPU LVS Job—Runtime and CPU Hours

Conclusion

IC Validator elastic CPU management is a paradigm shift for compute resource usage and management for physical verification jobs. The elastic CPU automation in IC Validator allows the user to spend less time thinking about how to assign resources to the job. In doing so, the resources used will be minimized and runs will start and finish sooner. All of this results in lower resource costs, less license usage and more time spent debugging violations.

08/06/21.CS732834617-optimize-physical-verification-cost-wp.

Pub: May 2021