

# TOPSAW Sawing Optimization Analysis Using Grid Computing

Ashwin Bommathanahalli<sup>2</sup>, Maoyuan Xie<sup>2</sup>, Zhifeng Yun<sup>2</sup>, Sun Joseph Chang<sup>1</sup>, Zhou Lei<sup>1</sup>, Gabrielle Allen<sup>2</sup>

<sup>1</sup>*School of Renewable Natural Resources, Louisiana State University, U.S.A*

<sup>2</sup>*Center for Computation & Technology, Louisiana State University, U.S.A*

*ashwin@cct.lsu.edu; mxie@cct.lsu.edu; zyun@cct.lsu.edu; xp2601@lsu.edu; zlei@cct.lsu.edu; gallen@cct.lsu.edu*

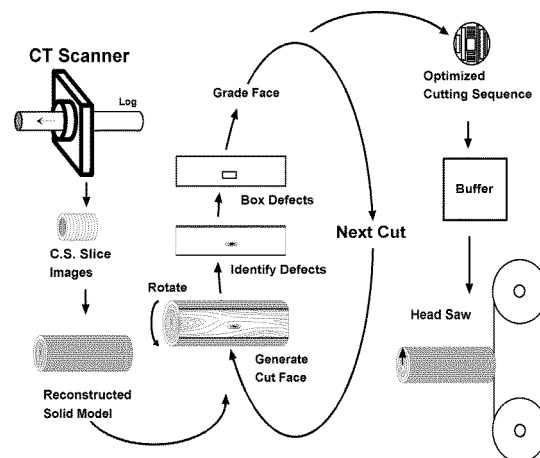
## Abstract

*TOPSAW program was developed by Dr. S. Chang and his team and has been successfully used in the replication of sawmill cutting to determine the optimal cutting pattern for a log to get the maximum value for lumber in terms of dollars based on National Hardwood Lumber Association grading rules. Currently Live sawing, which is one direction sawing, can be simulated for only one rotational orientation and one depth of opening cut using a single workstation. The optimal rotational orientation and depth of the opening cut significantly affects the value of lumber produced [14]. The optimal rotational orientation and the opening position outperformed the mean of 432 possible combinations by 14.7 percent. However the computational time involved in simulation of all the 24 different log orientations and 18 different ways to choose the opening depth for a log proves costly and is not effective for sawmill industry. In order to cut the simulation time we collaborate the TOPSAW and Grid Computing technology to leverage this sawing optimization. In this paper we propose a problem solving environment which makes TOPSAW Grid-enabled. The applicability of this solution is shown by a case study.*

## 1. Introduction

TOPSAW program reconstructs a virtual log from the CT scan images of the log. As shown in Figure 1, firstly the entire length of the log is scanned with the CT scanner to acquire its cross-sectional images. These images are then combined to construct the 3-D virtual log on the computer. From there it generates the full-length cut-faces at various rotational angles and depths. The software then identifies the internal defects in the log and boxes the defects on the cut-face and grades the generated boards according to the

National Hardwood Lumber Association (NHLA) grading standards and eventually determines the value of each board it generates. Then before sending the log to sawmill for actual cutting the optimal cutting pattern determined is stored in the memory and a barcode corresponding to the optimal cutting sequence is attached to the scanned log. The total value of the boards estimated by the software matched to 97% of the value produced at the sawmill. This laid the foundation for TOPSAW to be used as a practical tool to analyze different sawing patterns and determine the value of lumber thus produced (Guddanti and Chang 1998). In the end the cutting pattern for each of the 24 different log rotations for each of the 18 depths are determined. Finally the optimal cutting pattern for the whole simulation process is determined and also the orientation angle and the depth of the opening cut which produced this optimal value is found out.



**Figure 1.** A schematic of the TOPSAW sawing optimization system.

By implementing this technology we can substantially increase not only the BF per volume of

lumber but also value per volume of lumber. The impact - more income generated with less trees cut, more boards produced and more efficiently produced. By using this technology we gain close to 14.7 percent in value of lumber thus produced otherwise which would have gone waste (Chang, Cooper, Guddanti et al.). The enhanced value recovery promised by TOPSAW comes from a combination of larger lumber volume and better lumber grade. In addition to the potential to increase the profit of individual companies, the TOPSAW also has profound implications on the nation's conservation of forest resources. For the hardwood sawmill industry as a whole the enhanced value recovery means that American hardwood sawmill industry can convert logs into lumber more efficiently than their counterparts in other countries. As such, American hardwood sawmill industry would become more globally competitive. The enhanced global competitiveness would ensure that more logs would be processed into lumber in the United States. More lumber and fewer logs would be exported. The TOPSAW system would then translate into more US economy and additional jobs for her people. Beyond its impact on global competitiveness, the TOPSAW affects the resource conservation for the nation. For a nation efficient log conversion means that fewer logs would be cut to satisfy the market needs. As a result, fewer trees will be harvested. More trees would be left out in the woods, thereby extending the log supply and enhancing the resource conversation for the nation.

However TOPSAW optimization requires a powerful computational tool in order to realize the goal of achieving the substantial gain in lumber value. Also computational tools should be cheap so that the sawmills can afford it and gain substantial profits from it. The TOPSAW program consumes considerable number of processor cycles for running optimizations to determine the optimal cutting pattern. The replication of cutting is repeated for every 15 degrees of log rotation, 18 different ways to choose the depth from to start ( $18^2$ ), i.e.,  $24 * 18 = 432$  simulations.

The total simulation time for one sample data (CT scan image of the log) was around 9000 seconds using a single workstation. Considering the sawmill industries which processes 400 logs per 8 hour shifts the amount of computational time involved is huge. We propose to use High Performance Computing to leverage the sawing optimization and at the same time keep the cost of computational resources at the minimal for it to be practical.

The main objectives of this study are:

1. To show that TOPSAW can be collaborated with the grid.
2. To decrease the simulation time by parallelizing the optimization and distributing the computation on different nodes.
3. To test the parallelization of TOPSAW on the grid and check the workloads of the nodes.
4. To show the scalability of TOPSAW on grid when more features are added to TOPSAW.

## 2. Challenges

### 2.1. Sawmill challenges

The main challenge facing sawmills is the cost involved in buying a CT scanner. Although the economic feasibility of adapting an X-ray CT scanner or NMR scanner for lumber production has been studied before (Chang, 1989, Hodges et al. 1990) a simplified version of the analysis is presented here. With a 5 year loan, the annual mortgage payment would be \$751,387 at 8% interest. When the interest rate is 16%, the figure increases to \$916,228. As long as annual expense for other items such as salaries, electricity bills, maintenance, parts and other expendables stays below \$583,774 the TOPSAW system would be viable proposition.

**Table 1. Optimal Cutting Pattern: Live Sawing 15° log orientation WANE = 50%**

Stage	Thickness(in)	Value(\$)
1	5	1.34
2	4	9.51
3	6	19.25
4	8	21.25
5	8	21.73
6	5	11.03
7	8	24.05
8	8	9.78

Total \$ 118.31566

Time 17.83 seconds

**Table 2. Optimal Cutting Pattern: Live Sawing for 345° log orientation WANE = 50%**

Stage	Thickness(in)	Value(\$)
1	4	2.32
2	5	7.68
3	6	13.42
4	6	15.44
5	6	16.48
6	6	23.18
7	6	21.37

8	8	21.15
9	4	1.94

Total 122.985687  
Time(s) 19.879999

## 2.2. TOPSAW challenges

If sufficient computational resources are available, it can be possible to include all the 24 rotational angles (with 15 degree increments) for each depth of opening cut (with increments of 1) on live sawing since it results in a higher gain. The gain is close to 14.7 percent and it will result in substantial gain for the sawmill industries. However, there are challenges:

**2.2.1. Computing Time.** We obtained the results using a single workstation for live sawing for one rotational orientation and a depth of opening cut at a time. It can be seen that live sawing for single rotational orientation and the depth can be easily achieved using a single workstation but Live sawing for different log orientations and depths cannot be practically achieved using single workstation. Here are the details: Table 1 shows the optimal cutting pattern for Live sawing. As shown the optimization was performed for 15 degree log orientation and 14 number depth (each depth number is one-eighth of an inch i.e., 14 number depth = 14 \* 0.125 inches) on the virtual log generating \$118.315 worth of lumber. The simulation time for this case is 17.3 seconds. This time was achieved using a single workstation. This is practical since it takes less than 20 seconds to simulate on a single workstation for the log in our case.

**Table 3. Effect of various depths of opening cut on 195° log orientation**

Depth (Number x one-eighth of in)	Total Value(\$)
1	120.156
2	123.508
3	126.750
4	119.126
5	120.156
6	126.750
7	119.126
8	120.156
9	126.750
10	119.126
11	120.156
12	126.750
13	119.126
14	120.156

15	126.750
16	119.126
17	120.156
18	126.750

Maximum = 126.750  
Minimum = 119.156  
Mean = 122.010  
Variance = 15.410  
Standard Deviation = 3.9255

Table 2 shows the optimal cutting pattern for Live sawing for 345 degree log orientation and 14 depths. The result was obtained when the simulation was performed on a single workstation. The optimal cutting pattern generated in this case produced \$122.98 worth of boards of various thicknesses. This result in 4% higher gain compared to the previous case. The total time for the simulation was just under 22 seconds. Thus it is required to simulate the TOPSAW for as many orientations as possible. If we consider the log orientations to be simulated starting from 15 degree to 360 degree in 15 degree increments, we have to run the simulations 24 times for various angles.

Moreover from Table 3 when we consider simulating for different depths for a randomly chosen angle, we see that the maximum value obtained was \$126.750 and the minimum was \$119.126. Thus the simulation has to be performed not only for all the angles but also for various depths. The simulation time for doing this will be close to 9000 seconds depending on the log. Furthermore we can make sure not to lose the optimal cutting solution by excluding other angles of log orientation and depths. However the technology is no longer practicable because the simulation time is too costly. The solution to the problem can be Cluster Computing or Grid Computing. Our solution to the problem must be driven by cost-performance concerns rather than absolute performance.

## 2.3. Grid challenges

By implementing Grid Computing technology, multiple jobs can be executed at the same time in numerous computational resources on the Grid

**2.3.1. Workload allocation.** With Grid Computing, sufficient computational resources may be available for a single user. But, not all computational resources in the Grid environment can be accessible, due to authorization. And, not all available computation resources have the same computation power as well as same existing workload. How to efficiently

dispatch jobs is a key consideration for workload allocation strategy.

### 3. Grid computing

Computational simulations are becoming increasingly important since some simulations involve enormous computations which cannot be handled by even the most powerful supercomputers today. A solution is to bring multiple computers together connected through a computer network(Cluster Computing) or bring a number of already connected groups of computers to perform the computations(Grid Computing), based on the requirements of the application.

The present problem in our case can be solved using cluster computing but we are looking for a long term solution. In other words when there are new features added to the existing TOPSAW software to include Bi-directional and Grade sawing for all the different rotational orientations and depths, its hunger for computational resources will increase. In other words simulation time increases exponentially from live sawing to bi-directional to grade sawing. In this scenario, the clusters can hardly suffice to the computational needs while keeping the costs in mind. Moreover we are looking for a long term solution for the ever increasing demand for computational resources. The problem is aggravated when the TOPSAW software includes more features to fetch higher gain, technical maintenance and upgrading of the dedicated cluster will again be a cost issue.

Grid Computing provides a very effective solution to the problem, making it an economical and also very powerful computational resource. Unlike Cluster computing, Grid computing consists of heterogeneous computing resources distributed geographically all connected through network usually the Internet. The main goal of Grid Computing is to put the idle processors to work and achieve high throughput. The result of Grid Computing is that tremendous computing power is made available to all those who have access to the grid. Many industries are making use of grid computing like automotive industries for collaborative design and data intensive testing, military applications which involve simulation of thousands of entities, etc. We present Grid Computing as a solution to our problem. We make use of grid computing for several reasons:

1. Grid computing can be a powerful tool in any research computational arsenal. Most grids provide additional computational resources at no additional cost to the

researcher. For IT professionals, they provide a way to divert some computational demand to lower-cost systems, freeing up valuable time on high-end computational resources.

2. Since it is a shared resource, Grid is affordable and cost-effective to maintain. Grid computing is a way to share computing resources within and among organizations. Now, attention has shifted to the wider application of grid infrastructures in both commercial and government settings. Technology watchers believe grids will eventually become a staple of mainstream computing. Although some progress has occurred among enterprise data centers, that vision remains largely unfulfilled. One of the issues that block further adoption of grids include dearth of grid-enabled applications.
3. Grid computing wins over cluster computing when it comes to Scalability: Whenever TOPSAW software is upgraded to include more features its appetite for computational resources increases, cluster computing then becomes no longer affordable as more nodes need to be included to satisfy the computational needs and maintenance issue arises. In this case grid computing solves the problem economically and over a very long run.

### 4. TOPSAW-G: Collaborating TOPSAW and Grid

The authors present a Grid-enabled, problem-solving environment TOPSAW-G to fulfill the challenges explained above. This Grid-enabled tool benefits sawmill industries in three aspects:

1. Considering not only simulation for a particular log orientation but we also simultaneously simulate for every 15 degree from the starting angle. So that we do not lose any optimal cutting pattern which is better than the other log orientation angles
2. Considering also the simulation for various opening cut depths (depth of the first cut) i.e., for every one-eighth of an inch, in eighteen different ways. Since the log's rotational orientation and the depth of the opening cut influences the value of lumber produced in sawing hardwood logs(Chang, Cooper, Guddanti et al) [14].

3. Equipping TOPSAW with supercomputing capabilities thus reducing the simulation time significantly.

4. Collaborating all the functionalities of TOPSAW into one.

#### 4.1. Usage scenario

Before running TOPSAW, it prompts us to provide the list of parameters- the CT scan data file of the log of interest, the wane percentage, number of blades to be used for sawing, the initial angle of orientation, and the depth of the opening cut. The whole procedure of TOPSAW-G includes:

1. A user logs into an end machine and activates the proxy with a valid GSI certificate which authorized this user to access Grid services.

2. The CT scan image file of the log should be stored in the working directory in the remote machine.

3. The user specifies all the parameters required by TOPSAW-G in the parameter file. Parameters like percentage of wane, log orientation, number of blades, and depth of opening cut.

4. The script file generates 432 PBS files, for 24 different log orientations and 18 different depths with the parameters specified to be submitted as jobs simultaneously in the working directory.

5. The simulation run produces the output files in the directory specified with the optimal cutting patterns for each log orientation.

6. A program then collects all the output files and determines the best solution among all the optimal cutting patterns of all the angles for various depths and is stored in a file as an input to the sawmill for actual cutting.

#### 4.2. System Realization

1. Data Preparation: in this first stage, a local directory, containing a work directory, is created on the local machine. The input CT log data & other input files required by TOPSAW-G would be saved in binary or plain text files in this directory.

2. Data Staging: in this second stage, the local directory built in the first stage is replicated on the remote storage site.

3. Task Farming: in this third stage, TOPSAW-G will assign the 432 jobs to 24 available computation resources, cutting the simulation time by a factor of 24 according to the workload allocation decision.

4. Job Execution: in this fourth stage, the required command files will be generated & submitted to the remote computational machines.

5. Results Retrieving: in this fifth stage, the computational nodes will write their results back to the remote data repositories and read all the output files and get the best solution from the 432 output files and puts it in a file.

6. Staging Out: in this sixth stage, this file containing the best optimal solution will be transferred back to the user's local machine for it to be sent to the actual sawmill cutting.

We used the Workload allocation strategy of "Collaborating Mechanical Design Phases Across A Grid" [16] for our case. Four major factors of a resource, including computational capability, architecture, numbers of job running and numbers of job queuing, are considered to dispatch simulations. The CPU number and CPU speed of a resource are critical factors for computational capability. The numbers of job running and queuing are essential for load balancing. The computational capability  $C_i$  of a resource  $i$  is measured as follows:

$$C_i = \text{CPU Number} \times \text{CPU Speed}$$

The workload allocation strategy aims to dispatch certain number of simulations to a resource according to its computational capability and current workload. The number of simulations  $N_i$  to be dispatched to a resource  $i$  is calculated by using following equation:

$$N_i = \frac{W}{\sum_{k=1}^n C_k} - \frac{M_i}{C_i} + \left( \sum_{k=1}^n \frac{M_k}{C_k} \right) \cdot \left( \frac{C_i}{\sum_{k=1}^n C_k} \right)$$

where  $W$  is the total simulation number;  $n$  is the number of all available resources;  $C_k$  is the computational capability of a resource  $k$ ;  $M_k$  is the current workload in a resource  $k$ .

#### 5. Case Study

To achieve flexibility and portability, we adopted Perl scripts for the functionality implementations. We generated the PBS scripts for the different log orientations and depths. We used Portable Batch System(PBS) for scheduling our batch jobs. The perl script for generating the PBS scripts prompts the user to enter the parameter values by specifying the usage as follows:

```
Usage:
<ImageFile> <Wane> <NumberOfBlades>
<DepthOfInitialCut> <InitialAngle>
```

After entering the values: inv3 50 1 1 15 it generates

the PBS files for those parameter values. Here is a sample example of the PBS files generated by the perl script for all the log orientations where m is the angle, and n is the depth number:

```
#!/bin/sh

#PBS -N topsaw_15_1
#PBS -l nodes= 24 : ppn= 2
#PBS -l walltime = 1:00:00
#PBS -o topsaw_15_1.o
#PBS -e topsaw_15_1.e

cd
/home/topsaw/projects/commandline/bin/
./topsaw inv3 1 50 1 15
-----
.....
-----
#!/bin/sh
#PBS -N topsaw_m_n
#PBS -l nodes= 24 : ppn= 2
#PBS -l walltime = 1:00:00
#PBS -o topsaw_m_n.o
#PBS -e topsaw_m_n.e
cd
/home/topsaw/projects/commandline/bin/
./topsaw inv3 1 50 n m
-----
.....
-----
#!/bin/sh
#PBS -N topsaw_360_18
#PBS -l nodes= 24 : ppn= 2
#PBS -l walltime = 1:00:00
#PBS -o topsaw_360_18.o
#PBS -e topsaw_360_18.e

cd
/home/topsaw/projects/commandline/bin/
./topsaw inv3 1 50 18 360

The perl file which generated all these
input files also generates an
executable file which looks like this:
qsub topsaw_15_1.pbs

qsub topsaw_30_1.pbs
.....
.....
.....
qsub topsaw_m_n.pbs
.....
.....
.....
qsub topsaw_360_18.pbs
```

When the user executes the file, the jobs will be assigned to 24 different nodes and generates 432

output files for the corresponding submitted jobs. A program then reads through all the output files and finds out the optimal cutting pattern for each angle. Then it finds the best solution amongst them and displays it. The output looks like this: The Optimal cutting pattern is obtained by aligning the log at 195° rotation and choosing the opening cut at depth number 1 to produce lumber worth \$126.750.

## 6. Summary and Future Work

A Grid-enabled tool for sawmill industry is suggested in the paper. In the beginning, a brief introduction of TOPSAW program including its various functionalities, impact on sawmill industry in particular and US economy in general is presented, followed by a typical scenario.

At last, a case study for the scenario used previously, is discussed in detail, by employing open source engineering software packages. This project is currently in progress. The tool is being developed to include more features to maximize lumber production at the School of Renewable Natural Resources, Louisiana State University. More features are expected to be added into the TOPSAW in the future to enhance the tool's strength in maximizing lumber production and gain. The authors believe that this specific tool itself has significant practical value in the sawmill industry. We highlight an approach for Grid technologies applications in sawmill industry with distributed software packages involved.

## Acknowledgements

This project is sponsored by the U.S. Department of Energy (DOE) under Award Number DEFG02-04ER46136 and the Board of Regents, State of Louisiana, under Contract No. DOE/LEQSF(2004-07).

## 8. References

- [1] I. Foster, C. Kesselman, S. Tuecke, "The anatomy of the Grid: enabling scalable virtual organizations," *International J. Supercomputer Applications*, 15(3), 2001.
- [2] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, "A security architecture for computational Grids," *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pp. 83-92, 1998.
- [3] I. Foster, C. Kesselman, "Globus: A toolkit-based Grid architecture," *The Grid: Blueprint for a New Computing Infrastructure*, pp. 259-278, 1999.

- [4] J. Novotny, M. Russel, O. Wehrens, "GridSphere, A protal framework for building collaborations," *IEEE Concurrency and Practice*, April, 2004.
- [5] D. Thain, T. Tannenbaum, M. Livny, "Condor and the Grid," *Grid Computing: Making the Global Infrastructure a Reality*, John Wiley, 2003, ISBN: 0-470-85319-0.
- [6] Z. Lei et al. "ResGrid: A Grid-aware toolkit for reservoir uncertainty analysis," *Proceedings of CCGrid2006*, the IEEE Computer Society Press, Singapore, May 2006.
- [7] Andre Merzky, Shantenu Jha, Thilo Kielmann, Tom Goodale, "Simple API for Grid application," <http://forge.gridforum.org/projects/saga-rg/>, 2006. Linux Taken to the Extreme, *Linux Magazine*, Vol. 1, No.1, pp. 56-59.
- [8] G. Allen, K. Davis, et al. "The Grid application Toolkit: towards generic and easy application programming interfaces for the Grid," *Proceedings of the IEEE*, Vol. 93, No. 3, pp. 534-550, March 2005.
- [9] T. Sterling, J.D. Becker, D. Savarese, J.E Dorband, U.A. Ranawak, C.V. Packer, "A parallel workstation for scientific computation," *Proceedings, International Conference on Parallel Processing*, 1995.
- [10] M. May, "Piles of PCs," *American Scientist*, Vol. 26, pp. 128-129, March 1998
- [11] "Supercomputers: Plug and play," *The Economist*, 21 November 1998.
- [12] F. M. Hoffman and W. W. Hargrove, "Cluster computing: Linux taken to the extreme," *Linux Magazine*, Vol. 1, No. 1, pp. 56-59.
- [13] S.J. Chang "Live, bi-directional, and grade sawing of a virtual red oak log," *Proceedings of Scan Tech, International Conference*, LAS Vegas, Nevada 2005
- [14] S. Guddanti, S.J. Chang, "Replicating sawmill sawing with TOPSAW using CT images of a full-length hardwood log," *Forest Products Journal*, 48(1), pp. 72-75, 1998.
- [15] S.J. Chang, C. Cooper, S. Guddanti, "Effects of the log's rotational orientation and the depth of the opening cut on the value of lumber produced in sawing hardwood logs," *Forest Products Journal*, Vol. 55, No. 10, pp. 49-55, 2004.
- [16] M. Xie, F. Zhou, S. Acharya, G. Allen, T. Kosar, Z. Lei, A. Tabbal, "Collaborating mechanical design phases across a Grid," *IMSCC07, International Conference*, University of Iowa, 2007