Optimize EMX Resource Usage with Machine Learning

Bin Wan

Skyworks Solutions, Inc.

Overview

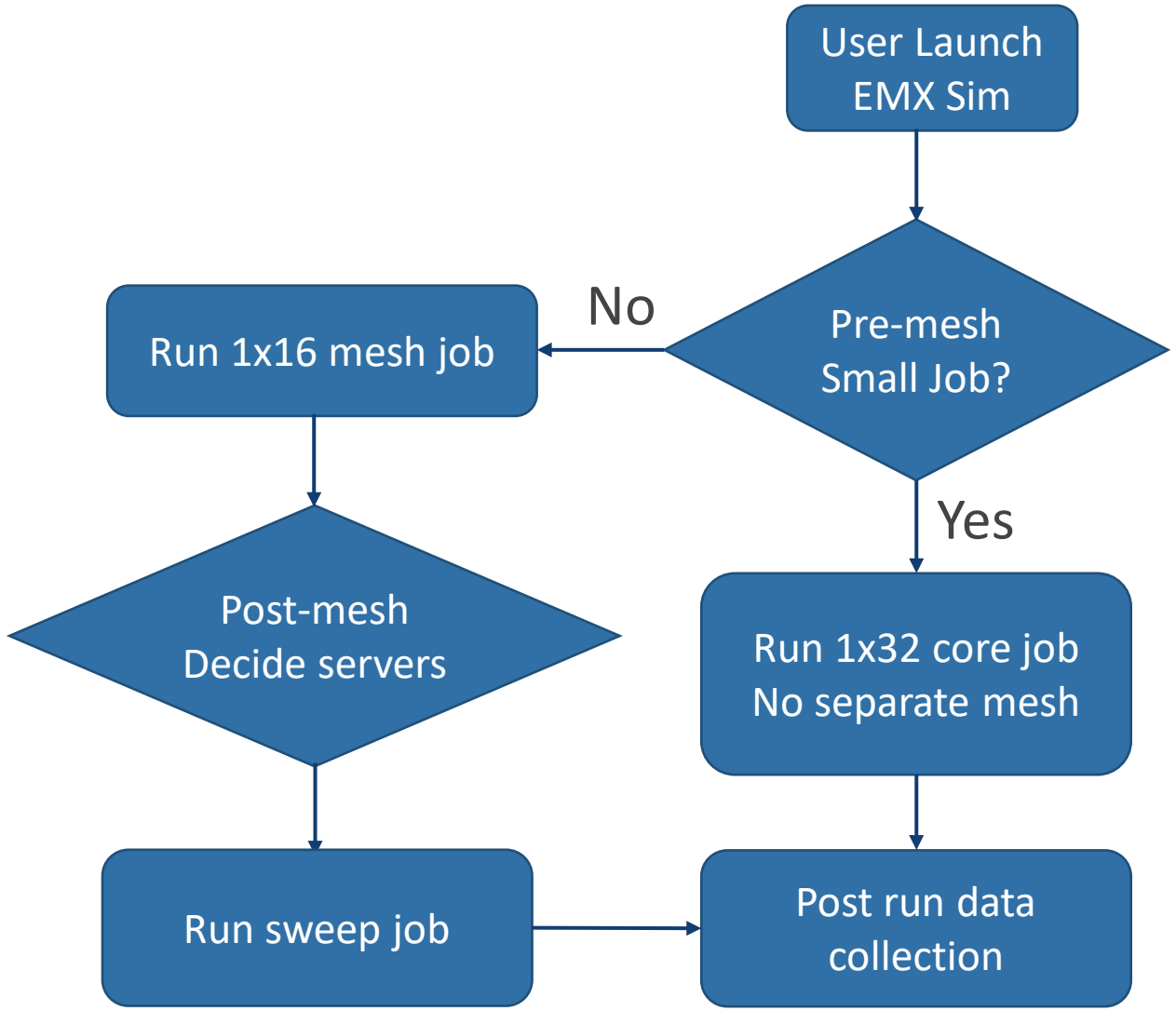
- EMX is easy to use, fast and accurate planar 3D EM solver
  - Skyworks internal GUI and design flow from Virtuoso layout editor
- Finding the right computing resource for EMX is a big challenge
  - Skyworks use LSF to manage computing resources
- Questions to answer before EMX actually runs:
  - Separate mesh job?
  - How many servers/cores to use?
  - How much memory per server?
  - What is the estimated run time?
- Machine learning algorithm that runs before each job starts to determine best resource allocation

Data Collection

- Meta data - used to filter sample points
  - PDK/project/library/cell/username
- Simulation setup
  - Number of ports
  - Number of frequency sweep points, maximum frequency
  - Important options, like --double-precision
- Pre-mesh geometry
  - GDS file size, # of shapes from GDS stream out log file
- Post-mesh geometry
  - Basis functions, vector elements, scalar elements
- Simulation finished results
  - Run time
  - Peak memory
  - LSF server statistics
    - CPU efficiency
    - Memory efficiency (Peak memory used / Max memory available)

EMX Simulation Flow

- Each EMX simulation can have up to 2 decisions, pre-mesh and post-mesh
- Pre-mesh decision will decide if the simulation should have separate meshing job, or just run combined mesh and sweep
- Post-mesh decision uses meshed results to determine sweep server configuration



SKYWORKS®

Pre-mesh Decision

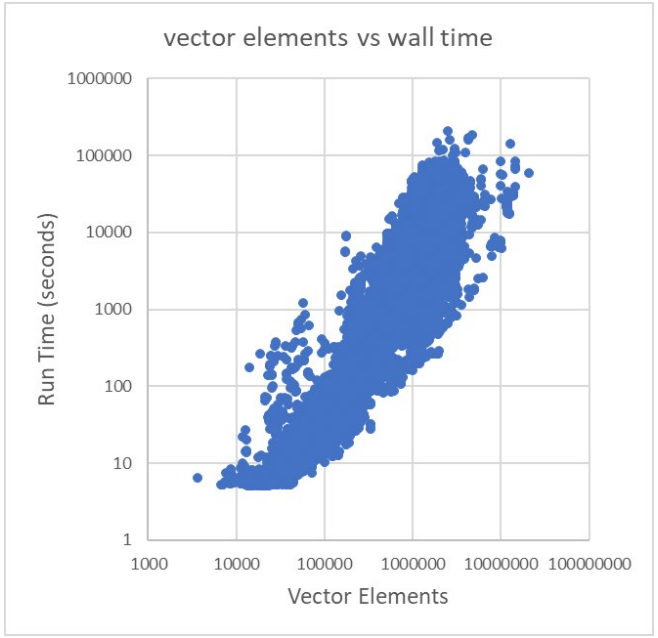
- Since EMX 2023.1, EMX can split simulation into meshing and frequency sweep
  - EMX can save mesh to a file and load an existing mesh
- Meshing tends to be very quick, and only runs on a few cores
  - Use server with best single core performance: order[cpuf]
- Pre-mesh algorithm is used to determine if the current job is big enough to have a separate mesh job
  - Use meta data, simulation setup, and GDS geometry to make an estimation on run time and memory usage
  - A small job does not benefit from a separate meshing process
    - Example: < 15 minutes estimated run time, < 200GB peak memory, AND < 32 ports
- If the job is determine to be small, run EMX with mesh and sweep combined, with 32 cores (1 base license)
- If the job is not small, submit EMX job with meshing only, to the fastest CPU server
  - --save-mesh=emx.mesh --stop-after=mesh

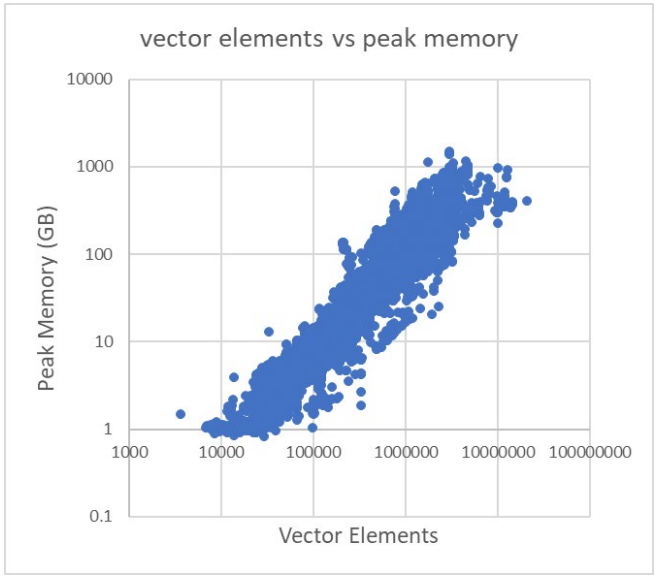
Post-mesh Decision

- Meshing elements can be extracted from EMX log file after meshing

```
Creating mesh...
20841 basis functions
23682 vector potential elements
16714 scalar potential elements
```

- Based Skyworks' experience, vector elements have the best correlation to run time and peak memory usage
  - Data filtering is still needed for best prediction
- Decide how many servers and cores per servers
  - Based on estimated CPU efficiency, peak memory ratio and server availability
- Launch EMX frequency sweep job
  - --load-mesh=emx.mesh

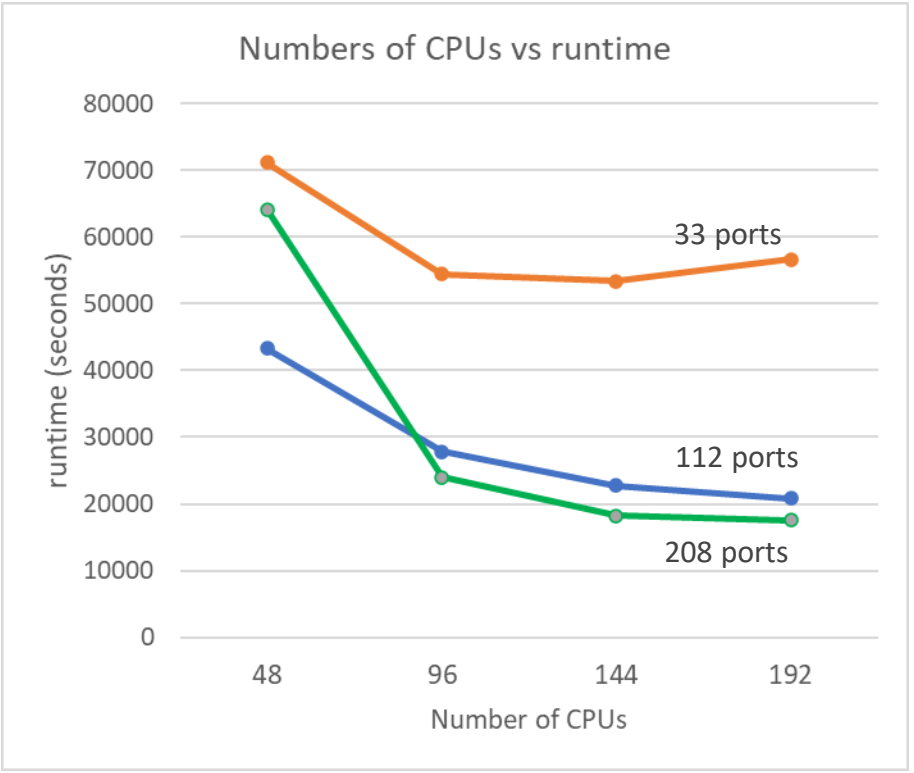




SKYWORKS®

Optimizing CPU Usage

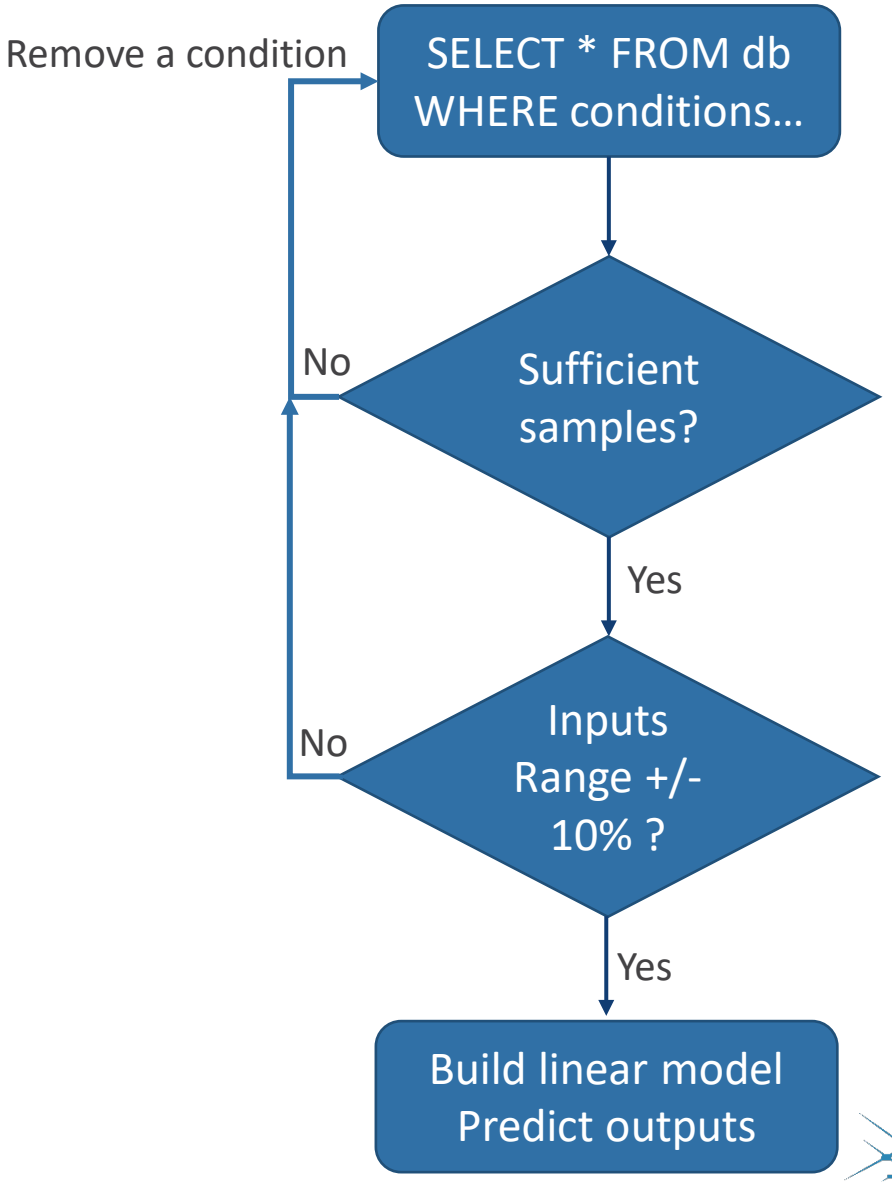
- Since EMX v6.1, EMX has the capability to run multiple servers with MPI
  - The primary server in MPI configuration is more heavily used than other servers
  - Prioritize single server over multiple servers
  - Based on Skyworks' observation, > 2 servers do not have good resource/performance trade off
- Look at CPU efficiency and memory efficiency from previous runs to decide how many servers/CPUs to use
  - CPU efficiency > 70% - allocate more servers if possible
  - CPU efficiency < 20% - do not allocate more servers
    - Some ports take much longer to solve than others
    - Reduce servers if under server shortage
  - Memory efficiency > 80% - job can use more memory
    - Allocated cores cannot be used if insufficient memory



SKYWORKS®

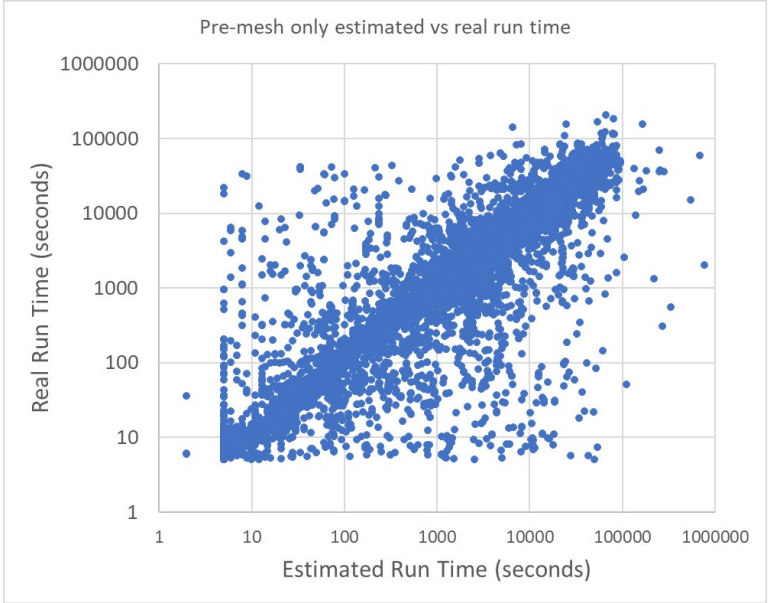
Prediction Algorithm

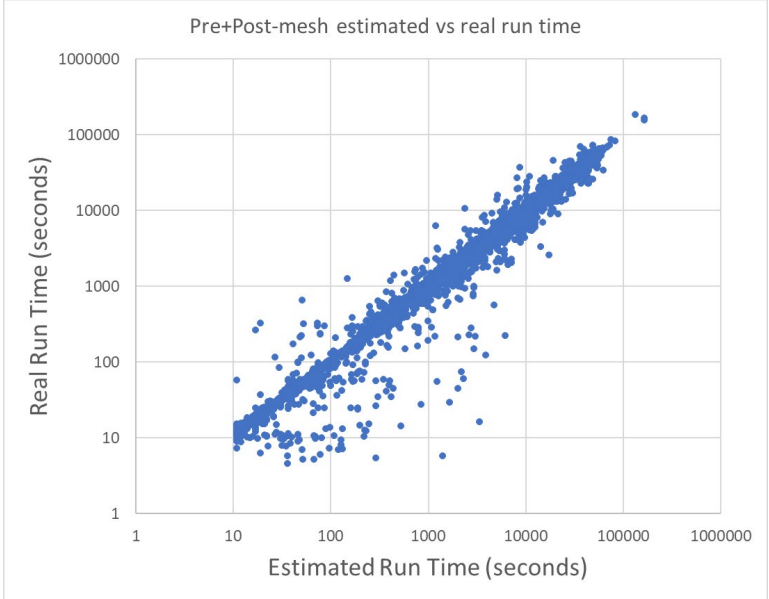
- Metadata filtering
  - Username, cell name, library name, project name, PDK
  - Filtered sample points greatly improves prediction accuracy
- Inputs
  - Number of ports, GDS size, mesh elements
  - Inputs needs to be +/- 10% from existing range
- Predicted outputs
  - Run time
  - Memory usage
  - Memory efficiency
  - CPU efficiency



Estimated Run Time

- Estimated run time can be fairly accurate with post mesh prediction
- Estimated run time can be used at LSF level to schedule jobs
  - bsub -We to pass estimated run time to LSF
  - Preempt jobs with longest remaining time
  - Delay start time for overnight jobs





SKYWORKS®

Implementation

- Pre-mesh and post-mesh algorithms coded in python, as web API
  - sklearn module used for linear regression model
- EMX setup GUI coded in python
  - Pre-mesh decision executed right after user clicks on run button
  - If separate meshing is decided, post-mesh python script is called after meshing to evaluate and launch sweep job
- MariaDB used to store EMX data collections
- LSF used for job management. Data collection done in LSF post execution trigger
- OpenMPI that came with EMX installation is used for multi-server jobs

Additional Considerations

- Unknown port count for native EMX blackbox flow
  - Blackboxed designs should always mesh in a separate job
- --double-precision argument will double memory usage
  - Default is on in EMX 2023.2
- Avoid sharing servers that run EMX job with another job
  - By default, EMX will use up to the larger of: 80% of the total memory, or the total memory minus 8 gigabytes
- EMX works best with physical cores, not hyperthreaded cores
- CPU factor can affect estimated run time
- EMX "recommended memory" message shows up too late and usually over-estimates
- Using all cores on one server may not be the best strategy
  - If memory efficiency is very high and CPU efficiency is very low, reduce cores per server (--parallel option) to save on HPC license

Summary

- Data collection
  - Meta data
  - Simulation setup
  - Pre-mesh and post-mesh geometries
  - Run performance
- Pre-mesh and post-mesh decisions
  - Pre-mesh decision determines if job is short and combines mesh and sweep together
  - Post-mesh decision sets sweep cores/servers based on CPU efficiency and memory efficiency
  - Filtering by metadata greatly improves prediction accuracy
- Estimated run time can be used to manage jobs with LSF