EVALUATION OF SOLVERS ON ACOUSTIC WAVES LF-DOMAIN MODELING

Hussain AlSalem

geohussain@berkeley.edu

Berkeley UNIVERSITY OF CALIFORNIA

INTRODUCTION

Challenge: Current IDR (iterative Krylov subspace) solver can be slower than Direct SuperLU for multi-RHS when forward modelling acoustic waves in Laplace-Fourier (LF) domain for 2nd and 4th order Finite-Difference (FD) discretization schemes.

Objective: Find a compromise of when direct SuperLU Solver is better than IDR and vice-versa:

(1) Strong and weak scaling comparisons between the two solvers.

(2) Maximum problem size that SuperLU can handle on fully packed nodes (default memory per core).

(3) Advantages of using unpacked nodes (large memory per core) on Edison for SuperLU.

GOVERNING EQUATIONS

The first-order system can be recast as a second-order equation in pressure after elimination of the particle velocities in the frequency-domain 3D acoustic first-order velocity-pressure system, that leads to a generalization of the Helmholtz equation:

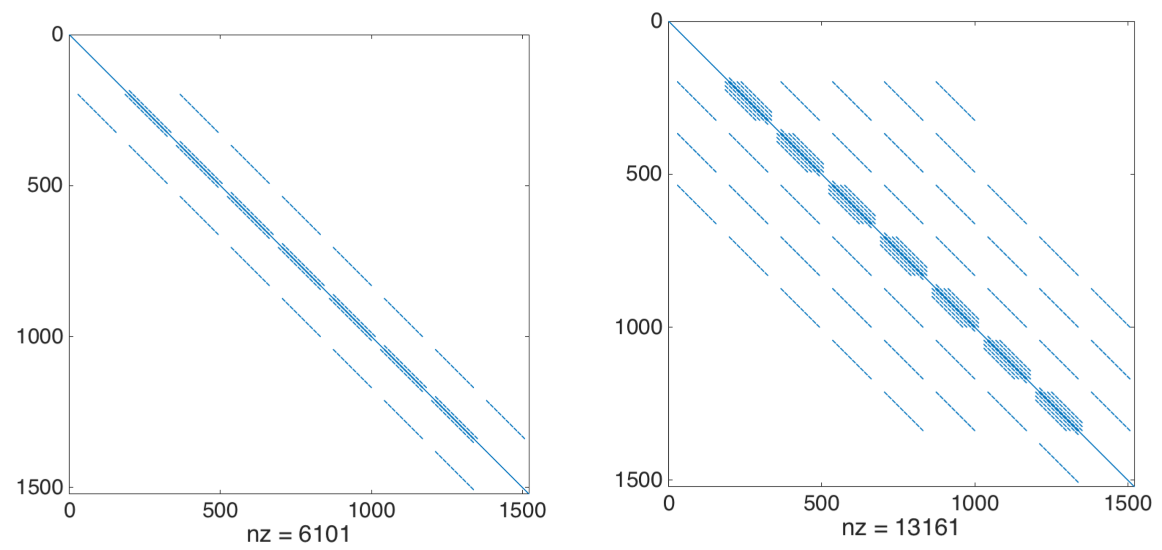
$$\frac{\omega^2}{\kappa(\mathbf{x})}p(\mathbf{x},\omega) + \frac{\partial}{\partial x}b(\mathbf{x})\frac{\partial p(\mathbf{x},\omega)}{\partial x} + \frac{\partial}{\partial y}b(\mathbf{x})\frac{\partial p(\mathbf{x},\omega)}{\partial y} + \frac{\partial}{\partial z}b(\mathbf{x})\frac{\partial p(\mathbf{x},\omega)}{\partial z} = s(\mathbf{x},\omega). \quad (1)$$

Equation 1 can be recast in matrix form:

$$[\mathbf{M}+\mathbf{S}]\mathbf{u} = \mathbf{A}\mathbf{u} = \mathbf{b}. \quad (2)$$

2ND VS 4TH ORDER SCHEMES

A matrix for both second and fourth order schemes:



REFERENCES

[1] Li, Xiaoye S and Demmel, et. al On SuperLU user's guide In 2011.

[2] Sonneveld, Peter and van Gijzen, Martin B On IDR (s): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations In 2008.

PARALLEL SUPERLU ALGORITHM

for *block* $K = 1$ **to** N **do**
- Factorize $A(K:N,K) \rightarrow L(K:N,K)$ (may involve pivoting)
- Compute $U(K,K+1:N)$ (via a sequence of triangular solves)
- Update $A(K+1:N,K+1:N) \leftarrow A(K+1:N,K+1:N) - L(K+1:N,K)\cdot U(K,K+1:N)$ (via a sequence of calls to GEMM)

The algorithm is chosen for parallel SuperLU:

(1) The sparsity structure can be determined before numerical factorization because of static pivoting.

(2) All the GEMM (general matrix multiplication) updates to the trailing submatrix are independent and so can be done in parallel.

SUPERLU PROS & CONS

Advantages of using SuperLU:

(1) Can solve for multiple RHS (sources) cheaply once matrix **A** is factorized.

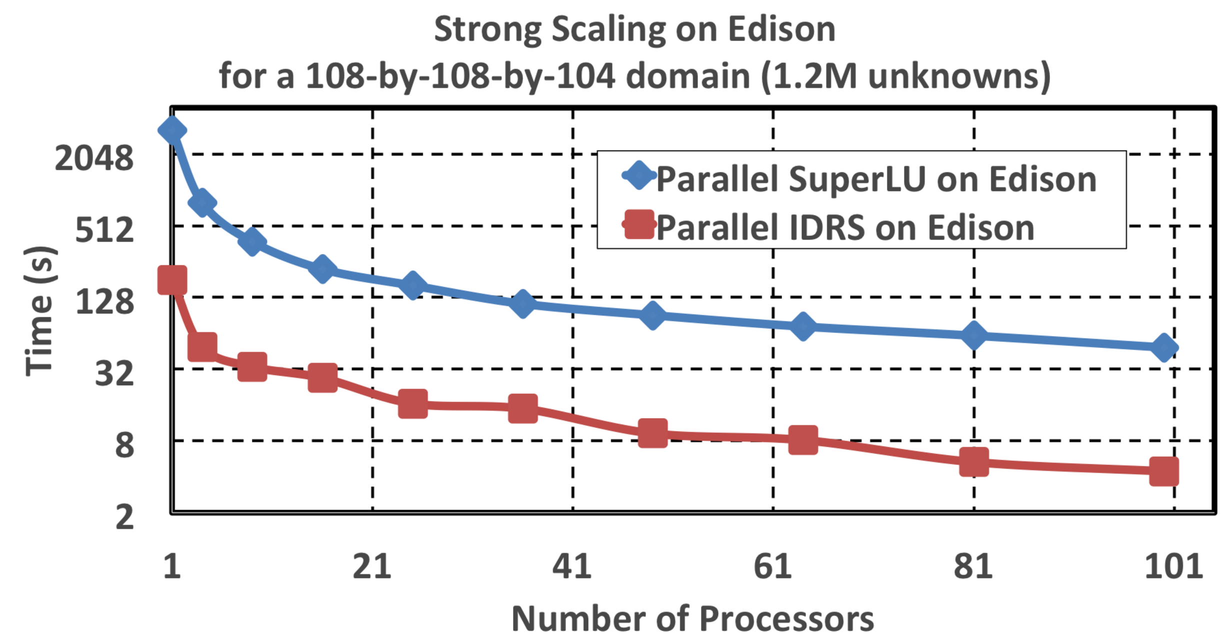
(2) Scales better when using multiple processors compared to the current IDR(s) method.

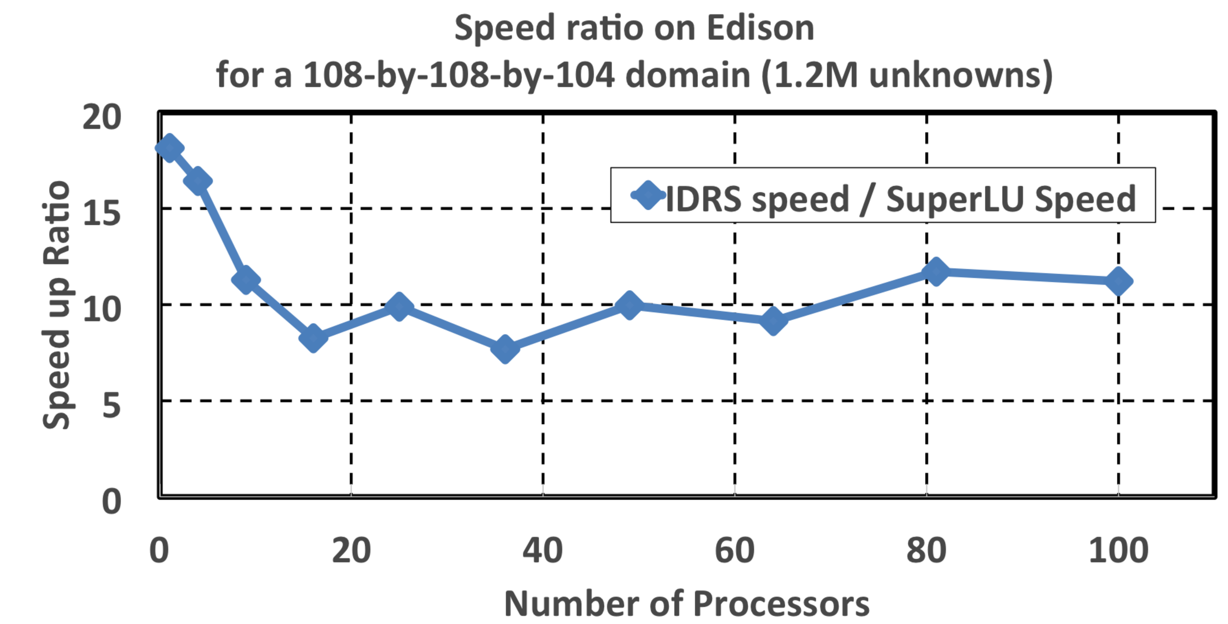
Drawbacks of using SuperLU:

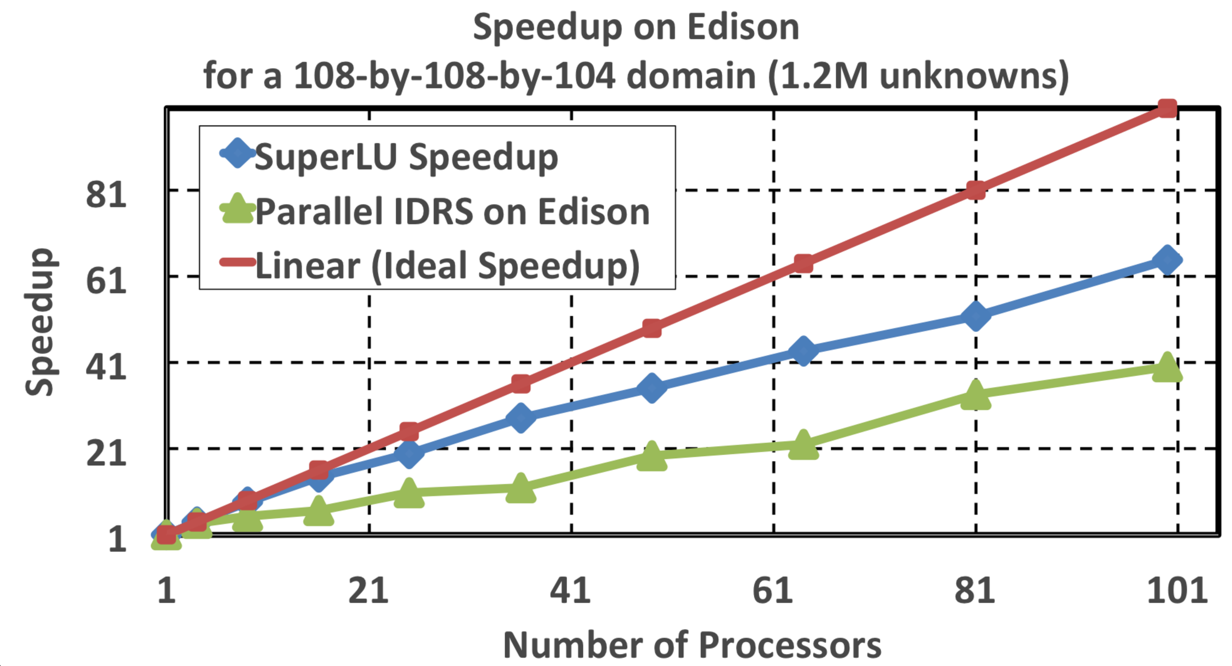
(1) Out of memory (OOM) errors during fill-in for large matrices.

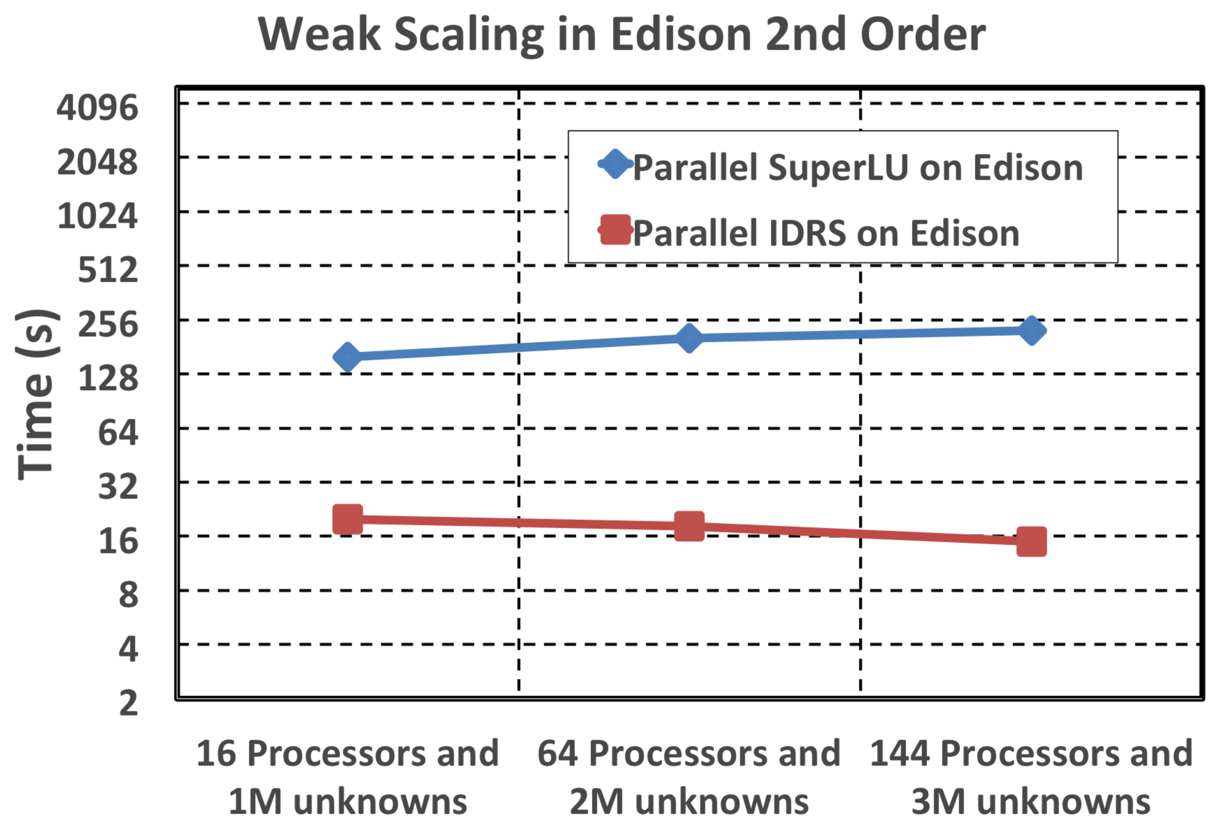
(2) SuperLU is generally slower than iterative methods when solving for one RHS.

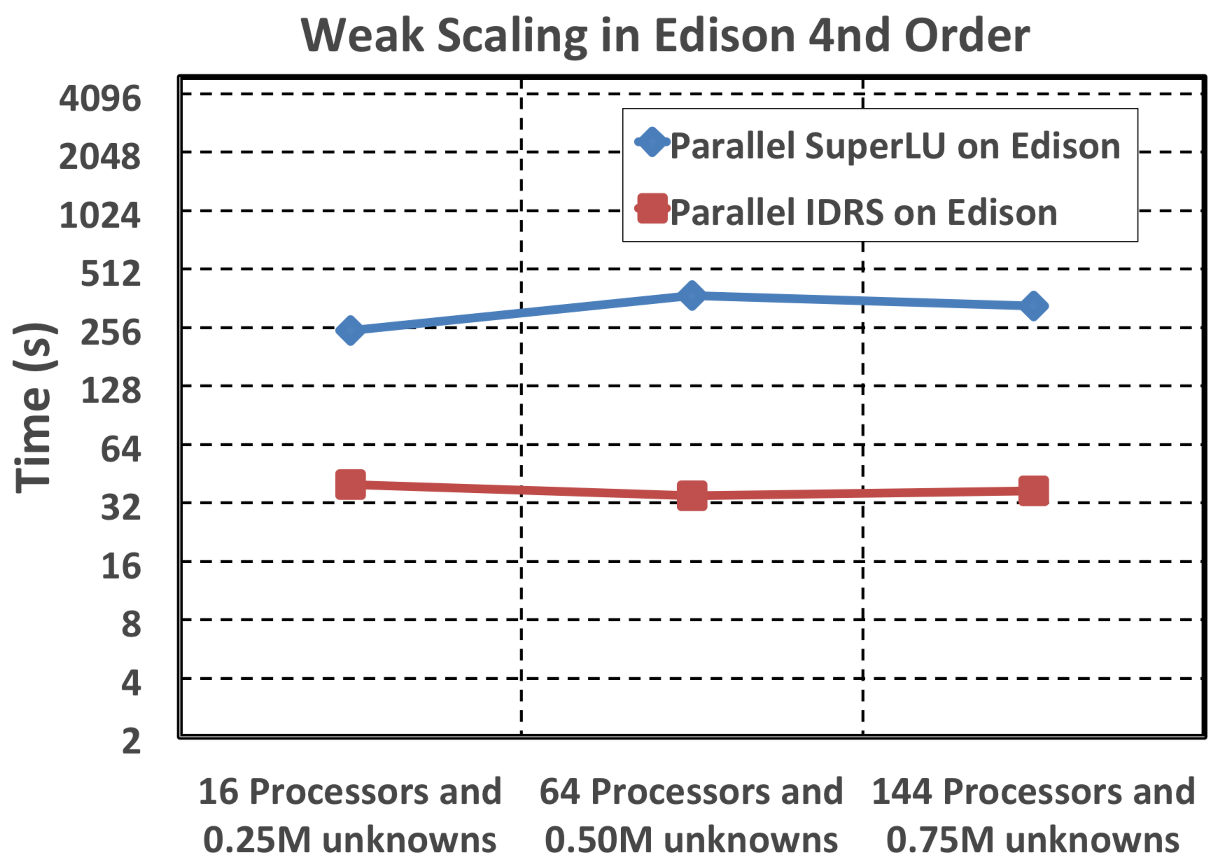
PARALLEL RESULTS ON FULLY PACKED NODES



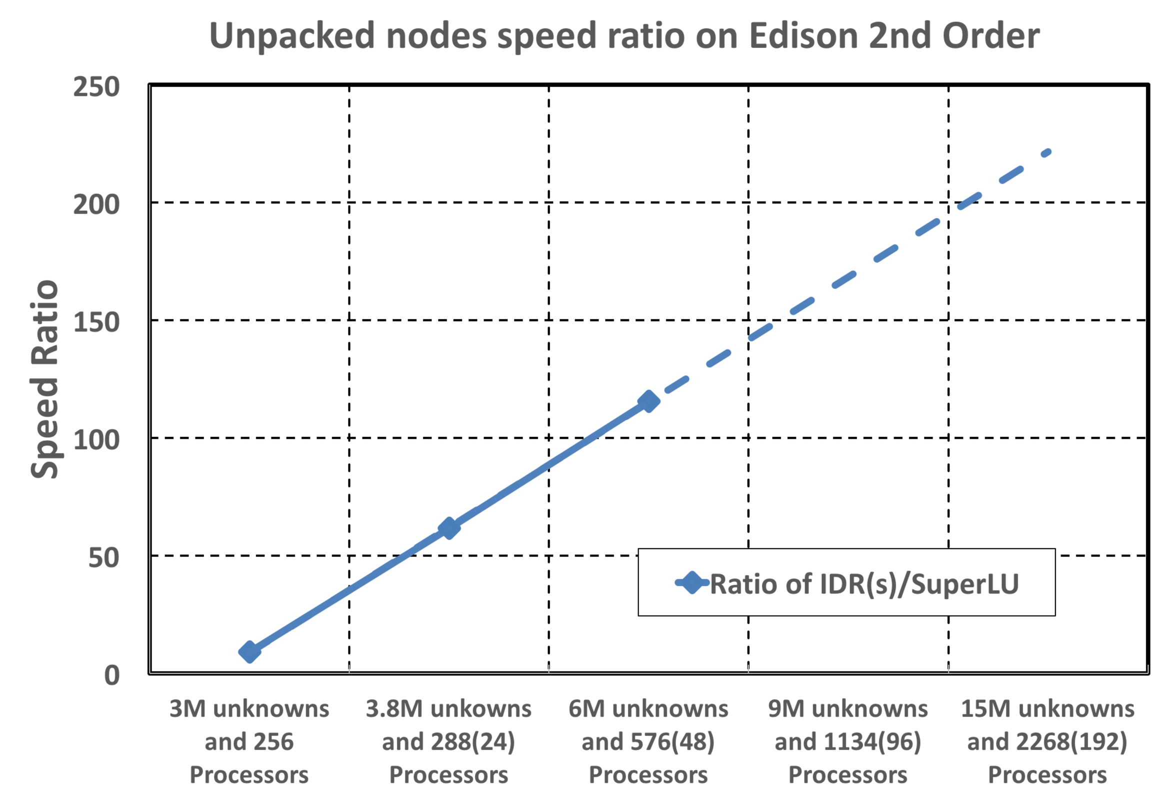








UNPACKED NODES RESULTS



OBSERVATIONS

(1) For medium-sized problems and using fully packed nodes (2.67GB per core), IDR(s) is 10 times faster than SuperLU.

(2) For large-sized problems and using unpacked nodes (32GB per core), IDR(s) is 100 times faster than SuperLU.

(3) SuperLU runs out of memory for medium-sized problems with fourth order discretization. Imagine the maximum size it can handle for elastic problems!

(4) SuperLU is more efficient in strong-scaling.