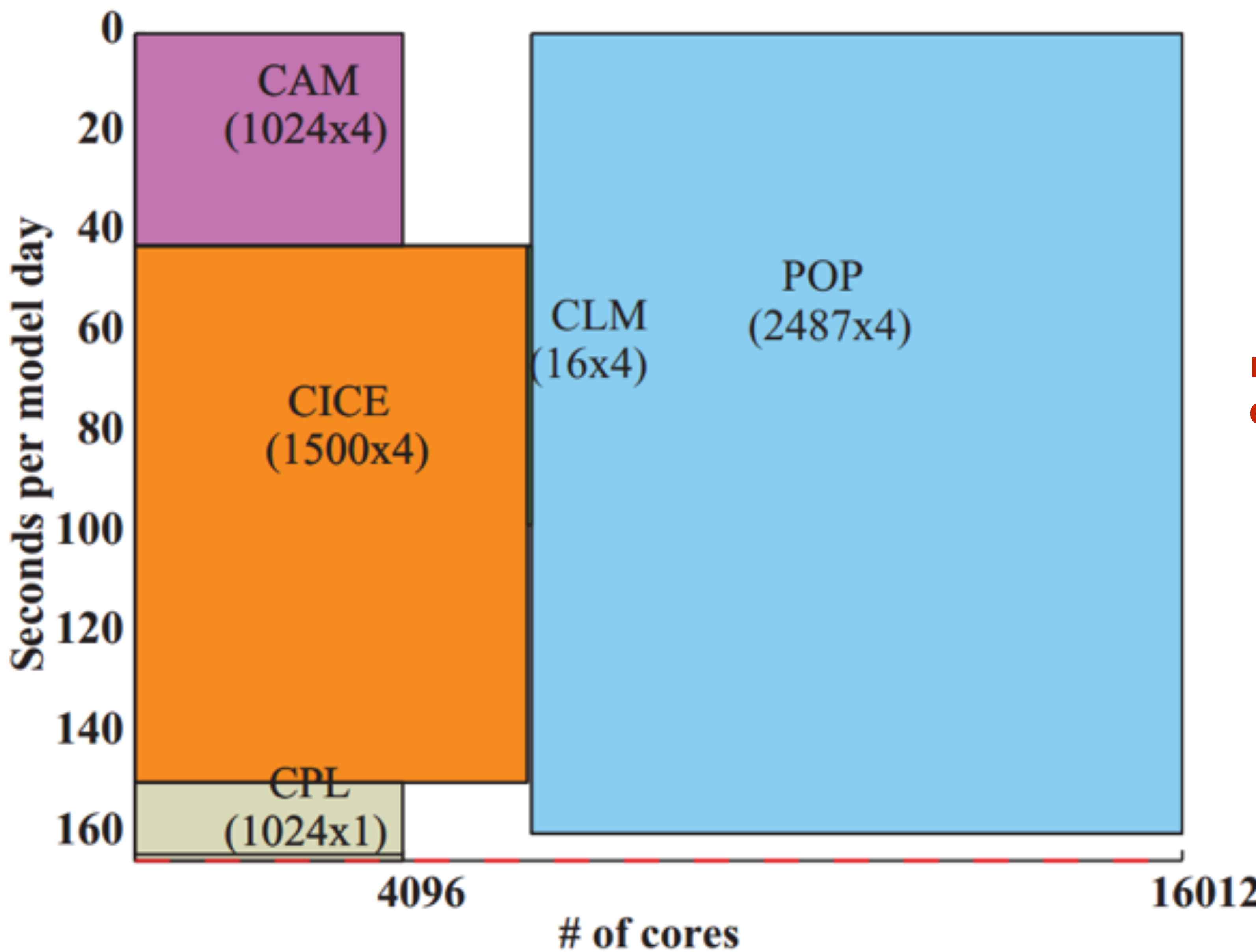
# A Scalable Barotropic Solver for the Parallel Ocean Program

Yong Hu[1], Xiaomeng Huang[1], Yu-Heng Tseng[2], Allison Baker[3], Frank Bryan[2], and John Dennis[3]

1 Ministry of Education Key Laboratory for Earth System Modeling
Tsinghua University, China
2 Oceanography Section, Climate and Global Dynamics Division
3 Computational and Information Systems Laboratory
National Center for Atmospheric Research, USA

# POP dominates in CESM

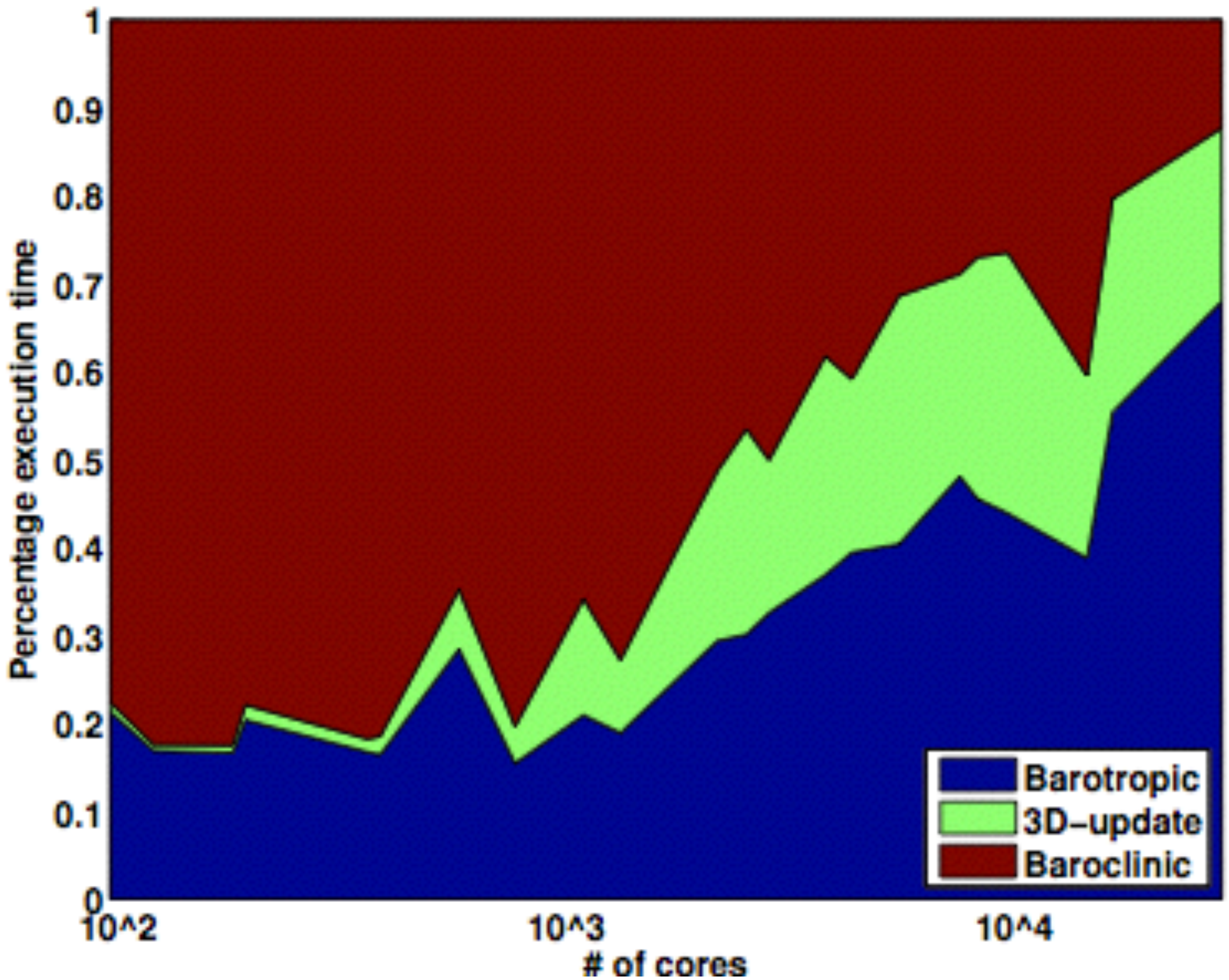



**resources : 58% to 66%**
**efficiency : 42% to 58%**

Computational performance of ultra-high-resolution capability in the Community Earth System Model. John Dennis, et al. HPCA'12

# Scalability of 0.1º POP

Baroclinic mode time dominates on a small number of cores,
Barotropic mode time dominates in a large number of cores (>5000)




Kraken, 99,072 core Cray XT5 system at the National Institute for Computational Science (NICS). Andrew Stone. 2011

# Barotropic mode in POP

- Require to solve an elliptic system for SSH in the barotropic mode.
- The elliptic problem is approximated as a linear system.

$$[\nabla \cdot H\nabla - \phi(\tau)]\eta^{n+1} = \psi(\eta^n, \eta^{n-1}, \tau) \longrightarrow Ax = b$$

- PCG (ChronGear) is used to solve this linear system.

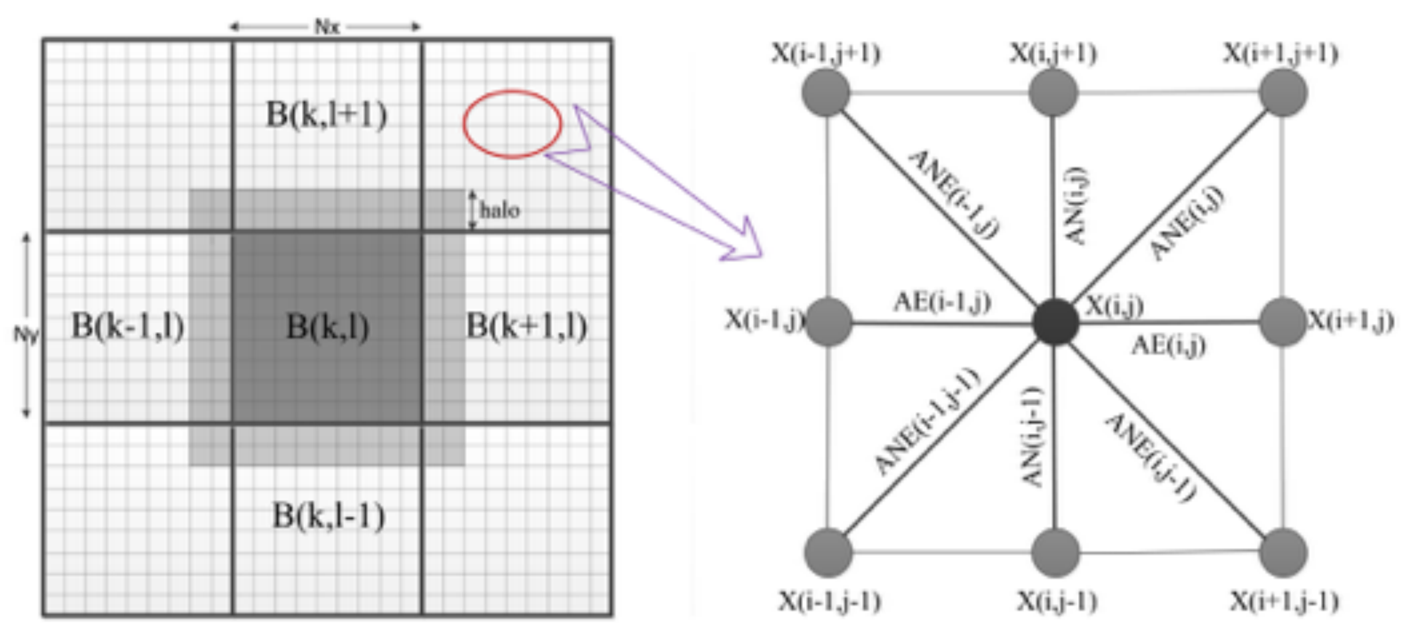
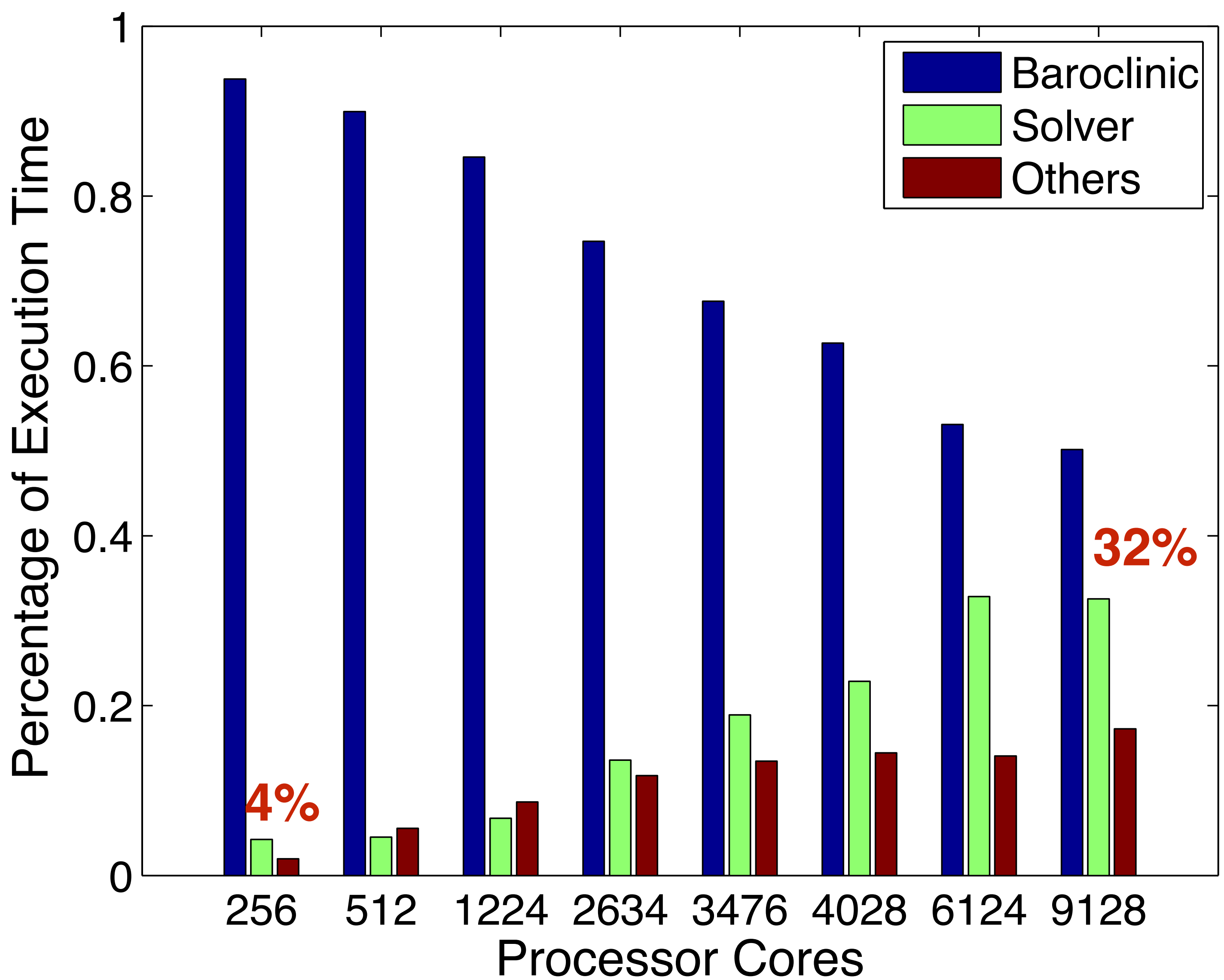



**Fig. 1.** Grid domain decomposition of POP

# Scalability of 0.1º POP on Yellowstone

ChronGear Solver does not scale well on large number of processes.




Yellowstone, at National Center for Atmospheric Research (NCAR) . Yong Hu. 2014

# ChronGear Solver

**Algorithm 1** Chronopoulos-Gear Solver

**Require:** Coefficient matrix $\mathbf{B}$, preconditioner $\mathbf{M}$, initial guess $\mathbf{x}_0$ and $\mathbf{b}$ associated with grid block $B_{i,j}$

// *do in parallel with all processes*

$\mathbf{r}_0 = \mathbf{b} - \mathbf{B}\mathbf{x}_0$, $\mathbf{s}_0 = 0$, $\mathbf{p}_0 = 0$; $\rho_0 = 1$, $\sigma_0 = 0$, $k = 0$;

**while** $k \le k_{max}$ **do**

  $k = k + 1$;

  $\mathbf{r}'_k = \mathbf{M}^{-1}\mathbf{r}_{k-1}$; /* preconditioning */

  $update\_halo(\mathbf{r}'_{k-1})$; /* boundary communication */

  $\mathbf{z}_k = \mathbf{B}\mathbf{r}'_k$; /* matrix-vector multiplication */

  $update\_halo(\mathbf{z}_k)$; /* boundary communication */

  $\tilde{\rho}_k = \mathbf{r}^T_{k-1}\mathbf{r}'_k$;

  $\tilde{\delta}_k = \mathbf{z}^T_k\mathbf{r}'_k$;

  $(\rho_k, \delta_k) = global\_sum(\tilde{\rho}_k, \tilde{\delta}_k)$; /* global reduction */

  $\beta_k = \rho_k / \rho_{k-1}$;

  $\sigma_k = \delta_k - \beta_k^2 \sigma_{k-1}$;

  $\alpha_k = \rho_k / \sigma_k$;

  $\mathbf{s}_k = \mathbf{r}'_{k-1} + \beta_k \mathbf{s}_{k-1}$;

  $\mathbf{p}_k = \mathbf{z}_k + \beta_k \mathbf{p}_{k-1}$;

  $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{s}_k$;

  $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \mathbf{p}_k$;

  **if** $k \% n_c == 0$ **then**

    check convergence;

  **end if**

**end while**

## Computation

$$\mathcal{T}_p + 15\frac{\mathcal{N}^2}{p}\theta$$

θ : time unit per floating-point operation

$N^2$ : number of grids

p : number of processors

# ChronGear Solver

**Algorithm 1** Chronopoulos-Gear Solver

**Require:** Coefficient matrix $\mathbf{B}$, preconditioner $\mathbf{M}$, initial guess $\mathbf{x}_0$ and $\mathbf{b}$ associated with grid block $B_{i,j}$

// *do in parallel with all processes*

$\mathbf{r}_0 = \mathbf{b} - \mathbf{B}\mathbf{x}_0$, $\mathbf{s}_0 = 0$, $\mathbf{p}_0 = 0$; $\rho_0 = 1, \sigma_0 = 0$, $k = 0$;
**while** $k \le k_{max}$ **do**
  $k = k + 1$;
  $\mathbf{r}'_k = \mathbf{M}^{-1}\mathbf{r}_{k-1}$; /* preconditioning */
  $update\_halo(\mathbf{r}'_{k-1})$; /* boundary communication */
  $\mathbf{z}_k = \mathbf{B}\mathbf{r}'_k$; /* matrix-vector multiplication */
  $update\_halo(\mathbf{z}_k)$; /* boundary communication */
  $\tilde{\rho_k} = \mathbf{r}'^{T}_{k-1}\mathbf{r}'_k$;
  $\tilde{\delta_k} = \mathbf{z}^T_k \mathbf{r}'_k$;
  $(\rho_k, \delta_k) = global\_sum(\tilde{\rho_k}, \tilde{\delta_k})$; /* global reduction */
  $\beta_k = \rho_k / \rho_{k-1}$;
  $\sigma_k = \delta_k - \beta_k^2 \sigma_{k-1}$;
  $\alpha_k = \rho_k / \sigma_k$;
  $\mathbf{s}_k = \mathbf{r}'_{k-1} + \beta_k \mathbf{s}_{k-1}$;
  $\mathbf{p}_k = \mathbf{z}_k + \beta_k \mathbf{p}_{k-1}$;
  $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{s}_k$;
  $\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \mathbf{p}_k$;
  **if** $k \% n_c == 0$ **then**
    check convergence;
  **end if**
**end while**

Boundary Communication

$$4\alpha + \left(\frac{8\mathcal{N}}{\sqrt{p}}\right)\beta$$

α : communication latency

β : transfer time per byte

# ChronGear Solver

**Algorithm 1** Chronopoulos-Gear Solver

**Require:** Coefficient matrix $\mathbf{B}$, preconditioner $\mathbf{M}$, initial guess $\mathbf{x}_0$ and $\mathbf{b}$ associated with grid block $B_{i,j}$
// *do in parallel with all processes*

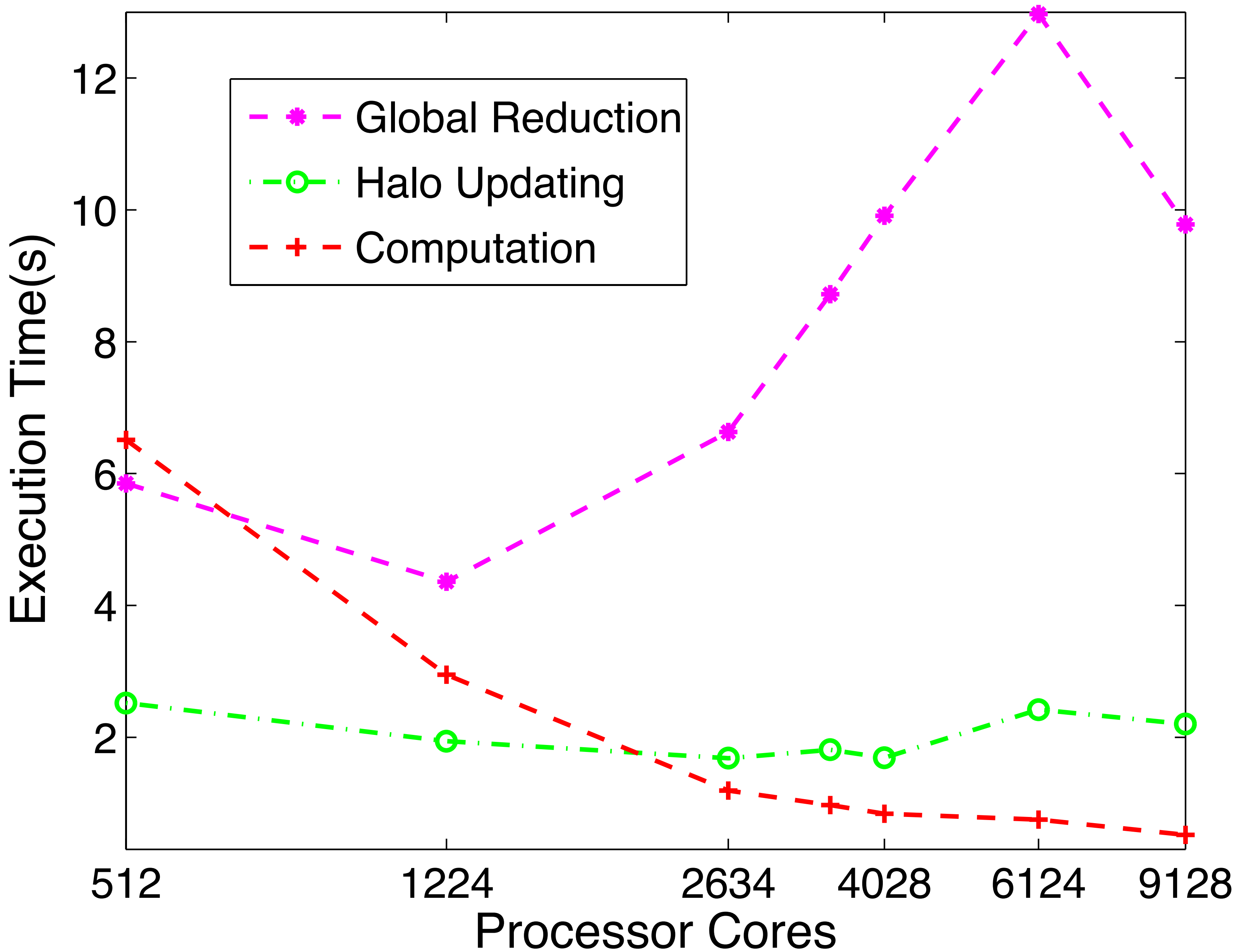
$\mathbf{r}_0 = \mathbf{b} - \mathbf{B}\mathbf{x}_0,\ \mathbf{s}_0 = 0,\ \mathbf{p}_0 = 0;\quad \rho_0 = 1, \sigma_0 = 0,\ k = 0;$
**while** $k \leq k_{max}$ **do**
$\quad k = k + 1;$
$\quad \mathbf{r}'_k = \mathbf{M}^{-1}\mathbf{r}_{k-1};$ /* preconditioning */
$\quad$ *update_halo*$(\mathbf{r}'_{k-1});$ /* boundary communication */
$\quad \mathbf{z}_k = \mathbf{B}\mathbf{r}'_k;$ /* matrix-vector multiplication */
$\quad$ *update_halo*$(\mathbf{z}_k);$ /* boundary communication */
$\quad \tilde{\rho_k} = \mathbf{r}^T_{k-1}\mathbf{r}'_k;$
$\quad \tilde{\delta_k} = \mathbf{z}^T_k\mathbf{r}'_k;$
$\quad (\rho_k, \delta_k) = global\_sum(\tilde{\rho_k}, \tilde{\delta_k});$ /* global reduction */
$\quad \beta_k = \rho_k / \rho_{k-1};$
$\quad \sigma_k = \delta_k - \beta_k^2\sigma_{k-1};$
$\quad \alpha_k = \rho_k / \sigma_k;$
$\quad \mathbf{s}_k = \mathbf{r}'_{k-1} + \beta_k\mathbf{s}_{k-1};$
$\quad \mathbf{p}_k = \mathbf{z}_k + \beta_k\mathbf{p}_{k-1};$
$\quad \mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k\mathbf{s}_k;$
$\quad \mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\mathbf{p}_k;$
$\quad$ **if** $k\%n_c == 0$ **then**
$\qquad$ check convergence;
$\quad$ **end if**
**end while**

Global Reduction

$$\mathcal{T}_g = \log p\alpha$$

# ChronGear Solver Components

Global reduction is the bottleneck of the ChronGear Solver in 0.1° POP.



# P-CSI Solver

**Algorithm 2** Preconditioned Stiefel Iteration solver

**Require:** Coefficient matrix $\mathbf{B}$, preconditioner $\mathbf{M}$, initial guess $\mathbf{x}_0$ and $\mathbf{b}$ associated with grid block $B_{i,j}$; Estimated eigenvalue boundary $[\nu, \mu]$;

// *do in parallel with all processes*

```
α = 2/(μ−ν), β = (μ+ν)/(μ−ν), γ = β/α, ω_0 = 2/γ;   k = 0;
r_0 = b − B x_0; x_1 = x_0 − γ^{-1} M^{-1} r_0; r_1 = b − B x_1;
while k ≤ k_max do
   k = k + 1;
   ω_k = 1/(γ − (1/(4α^2)) ω_{k−1});        /* the iterated function */
   r'_{k−1} = M^{-1} r_{k−1};                /* preconditioning */
   Δx_k = ω_k r'_{k−1} + (γω_k − 1) Δx_{k−1};
   x_k = x_{k−1} + Δx_{k−1};
   r_k = b − B x_k;                          /* matrix–vector multiplication */
   update_halo(r_k);                         /* boundary communication */
   if k%n_c == 0 then
      check convergence;
   end if
end while
```

$\alpha = \frac{2}{\mu-\nu}$, $\beta = \frac{\mu+\nu}{\mu-\nu}$, $\gamma = \frac{\beta}{\alpha}$, $\omega_0 = \frac{2}{\gamma}$; $k = 0$;

$\mathbf{r}_0 = \mathbf{b} - \mathbf{B}\mathbf{x}_0$; $\mathbf{x}_1 = \mathbf{x}_0 - \gamma^{-1}\mathbf{M}^{-1}\mathbf{r}_0$; $\mathbf{r}_1 = \mathbf{b} - \mathbf{B}\mathbf{x}_1$;

**while** $k \le k_{max}$ **do**

  $k = k + 1$;

  $\omega_k = 1/(\gamma - \frac{1}{4\alpha^2}\omega_{k-1})$; /* the iterated function */

  $\mathbf{r}'_{k-1} = \mathbf{M}^{-1}\mathbf{r}_{k-1}$; /* preconditioning */

  $\Delta\mathbf{x}_k = \omega_k \mathbf{r}'_{k-1} + (\gamma\omega_k - 1)\Delta\mathbf{x}_{k-1}$;

  $\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta\mathbf{x}_{k-1}$;

  $\mathbf{r}_k = b - \mathbf{B}\mathbf{x}_k$; /* matrix–vector multiplication */

  *update_halo*$(\mathbf{r}_k)$; /* boundary communication */

  **if** $k\%n_c == 0$ **then**

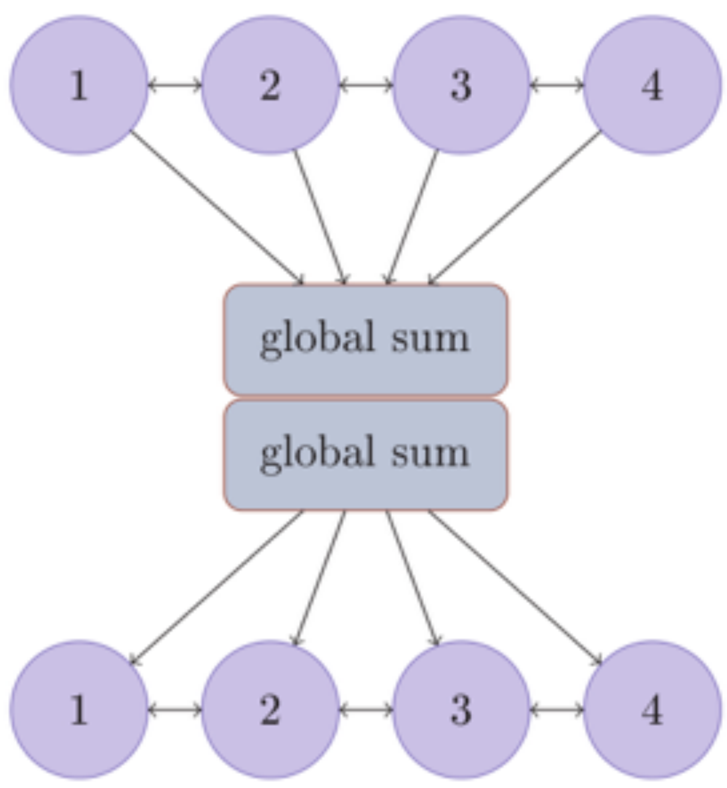
    check convergence;

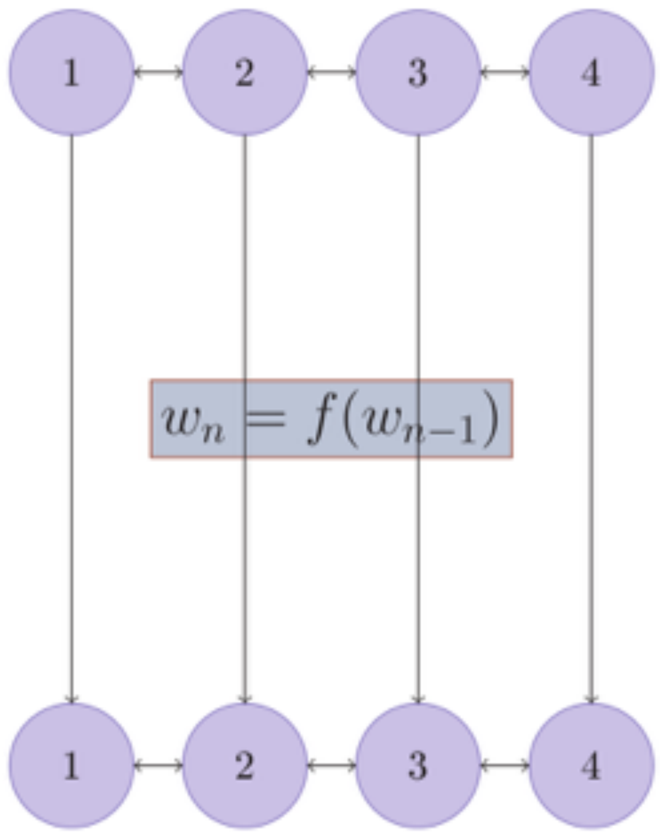
  **end if**

**end while**

# P-CSI .vs. ChronGear

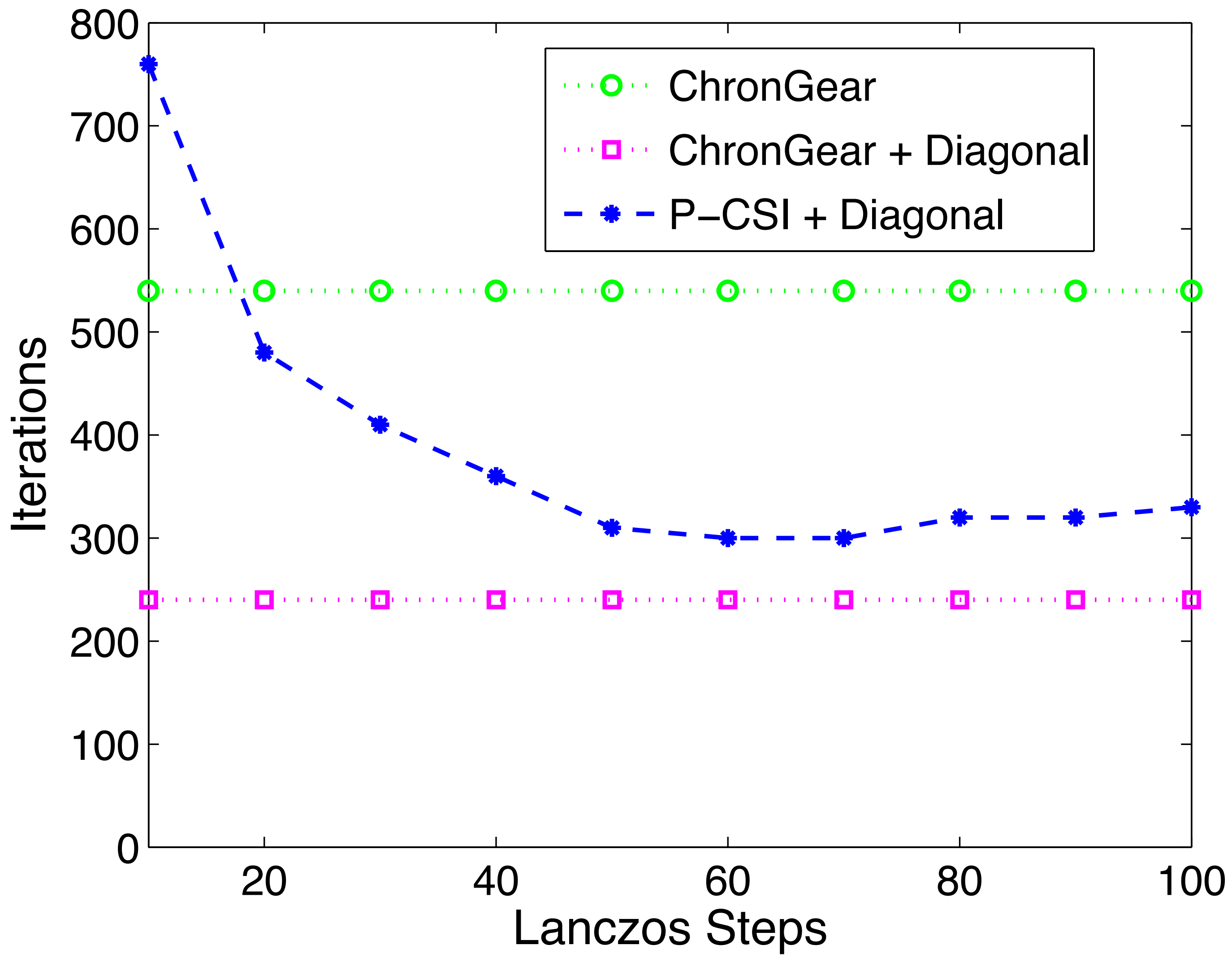
- Unlike ChronGear, P-CSI has no global reduction
- P-CSI needs slightly more iterations, but less computation in each iteration
- P-CSI requires two extreme eigenvalues



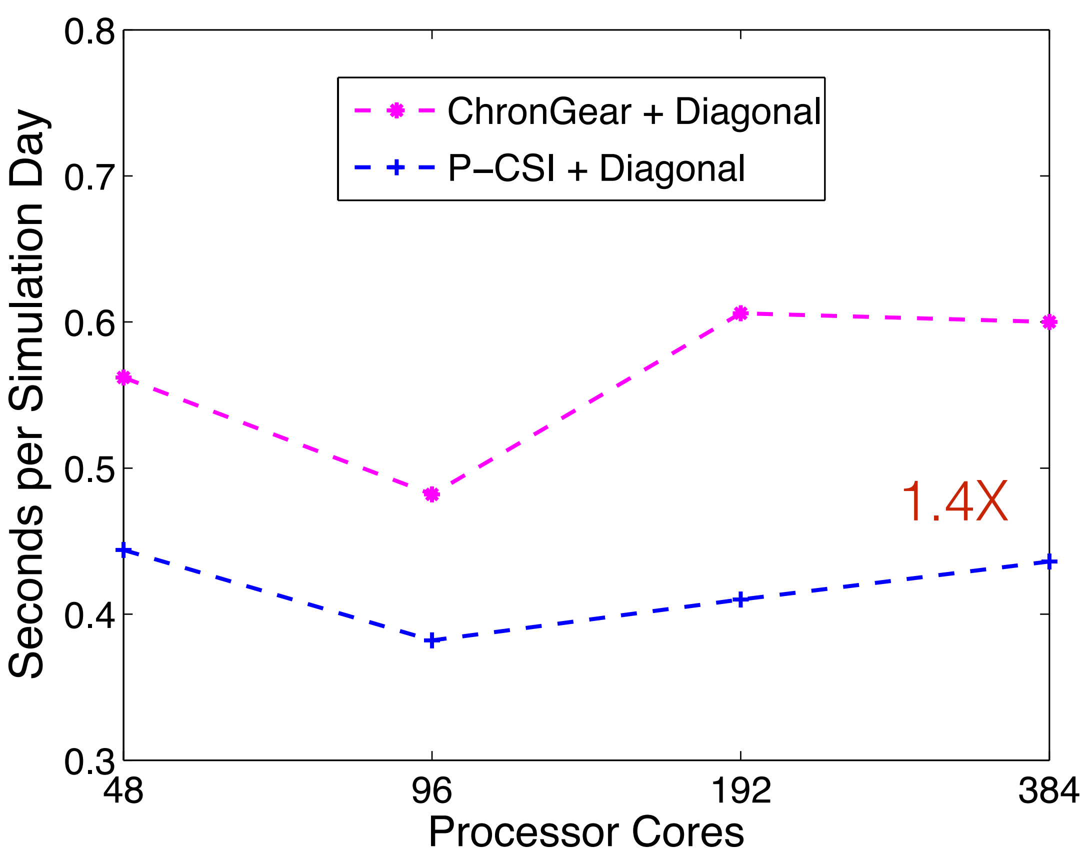
# Eigenvalue Estimation

- Lanczos method to construct a tridiagonal matrix T
- T has eigenvalues close to $M^{-1}A$
- Estimation needed only once, extra overhead less than one barotropic step.

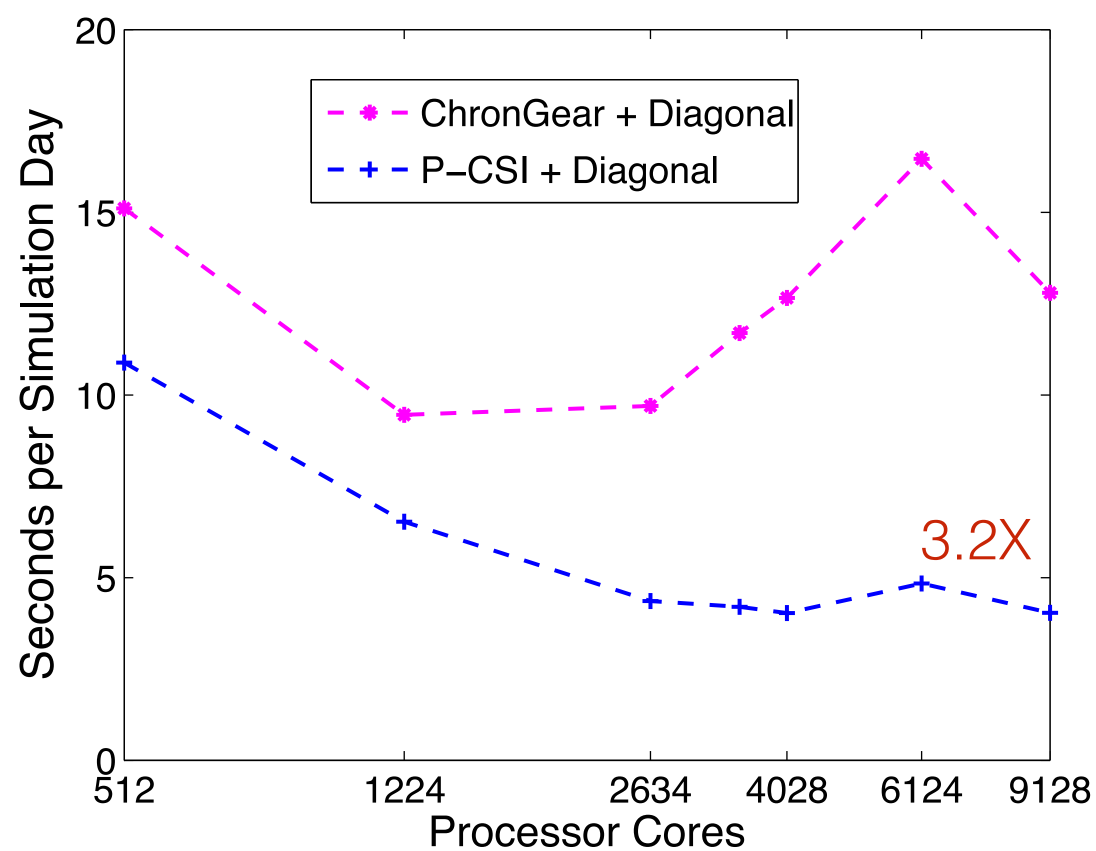


# Scalability on Yellowstone

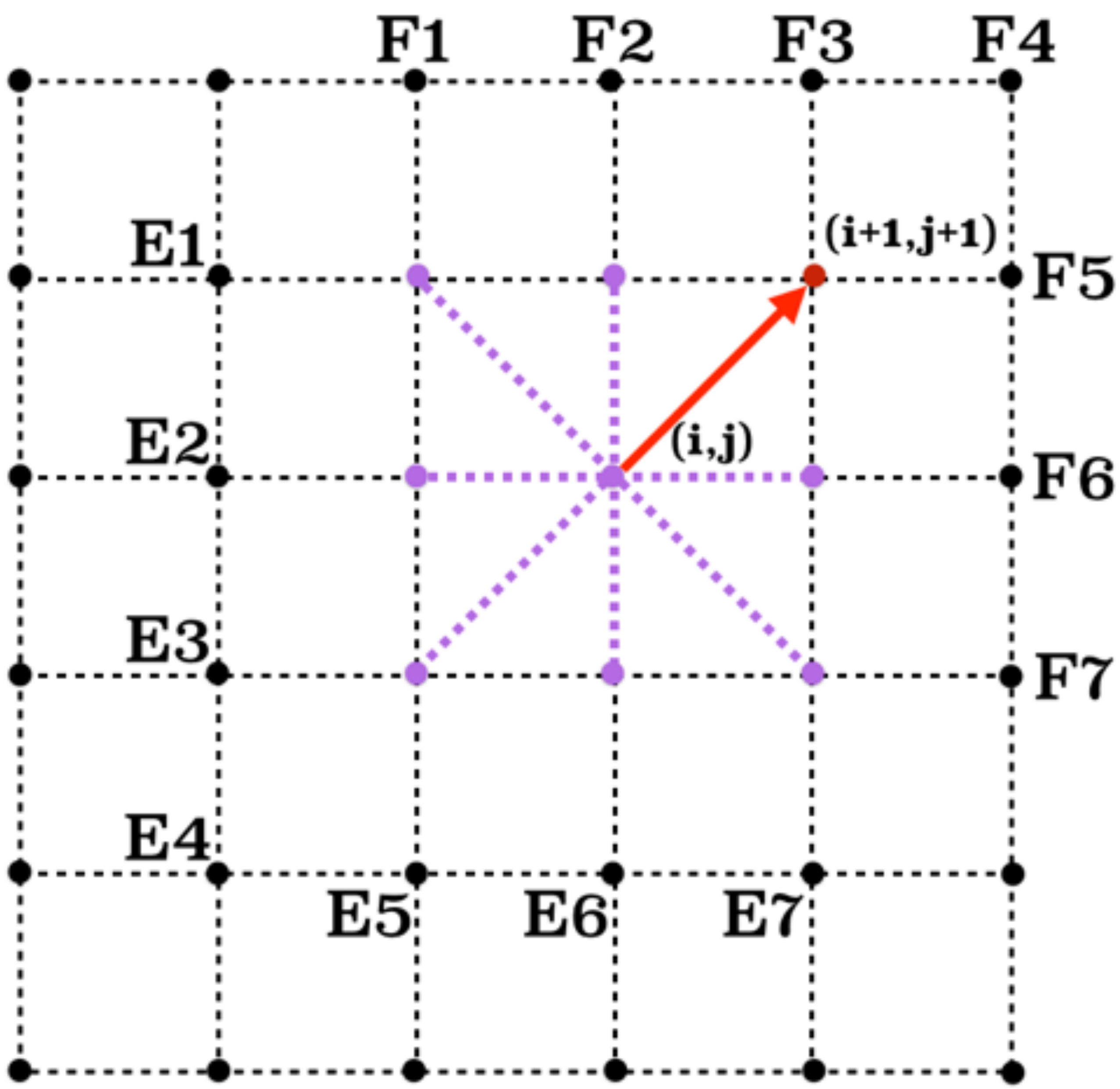
## 1 degree POP




## 0.1 degree POP



# Error Vector Propagation (EVP) Preconditioning




$$
\begin{gathered}
A^0_{i,j}\eta_{i,j} + A^e_{i,j}\eta_{i+1,j} + A^n_{i,j}\eta_{i,j+1} + A^{ne}_{i,j}\eta_{i+1,j+1} \\
+A^{ne}_{i-1,j}\eta_{i-1,j+1} + A^e_{i-1,j}\eta_{i-1,j} + A^{ne}_{i-1,j-1}\eta_{i-1,j-1} \\
+A^n_{i,j-1}\eta_{i,j-1} + A^{ne}_{i+1,j-1}\eta_{i,j-1} = \psi_{i,j}
\end{gathered}
$$

$$
\begin{gathered}
\eta_{i+1,j+1} = (1/A^{ne}_{i,j})(\psi_{i,j} - A^0_{i,j}\eta_{i,j} - A^e_{i,j}\eta_{i+1,j} \\
-A^n_{i,j}\eta_{i,j+1} - A^{ne}_{i-1,j}\eta_{i-1,j+1} + A^e_{i-1,j}\eta_{i-1,j} \\
-A^{ne}_{i-1,j-1}\eta_{i-1,j-1} - A^n_{i,j-1}\eta_{i,j-1} - A^{ne}_{i+1,j-1}\eta_{i,j-1})
\end{gathered}
$$

Roache 1995, *Elliptic marching methods and domain decomposition*.

# Preconditioning and Iterations

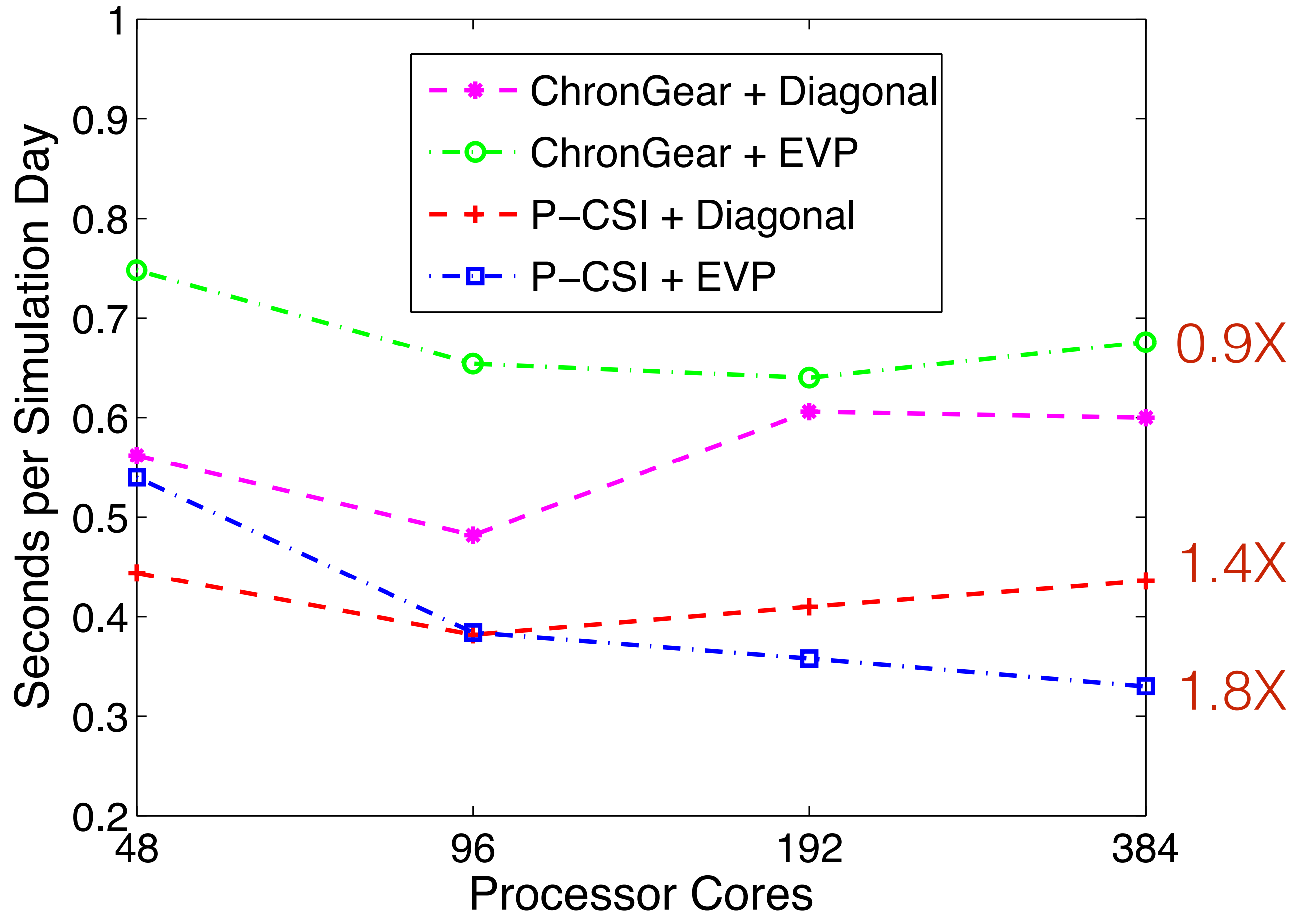
$$\begin{bmatrix} B_i^{nw} & B_i^{n} & B_i^{ne} \\ B_i^{w} & B_i & B_i^{e} \\ B_i^{sw} & B_i^{s} & B_i^{se} \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} B_1^{-1} & & \\ & \ddots & \\ & & B_{m^2}^{-1} \end{bmatrix}$$

EVP preconditioning reduces the iteration number to about one-third

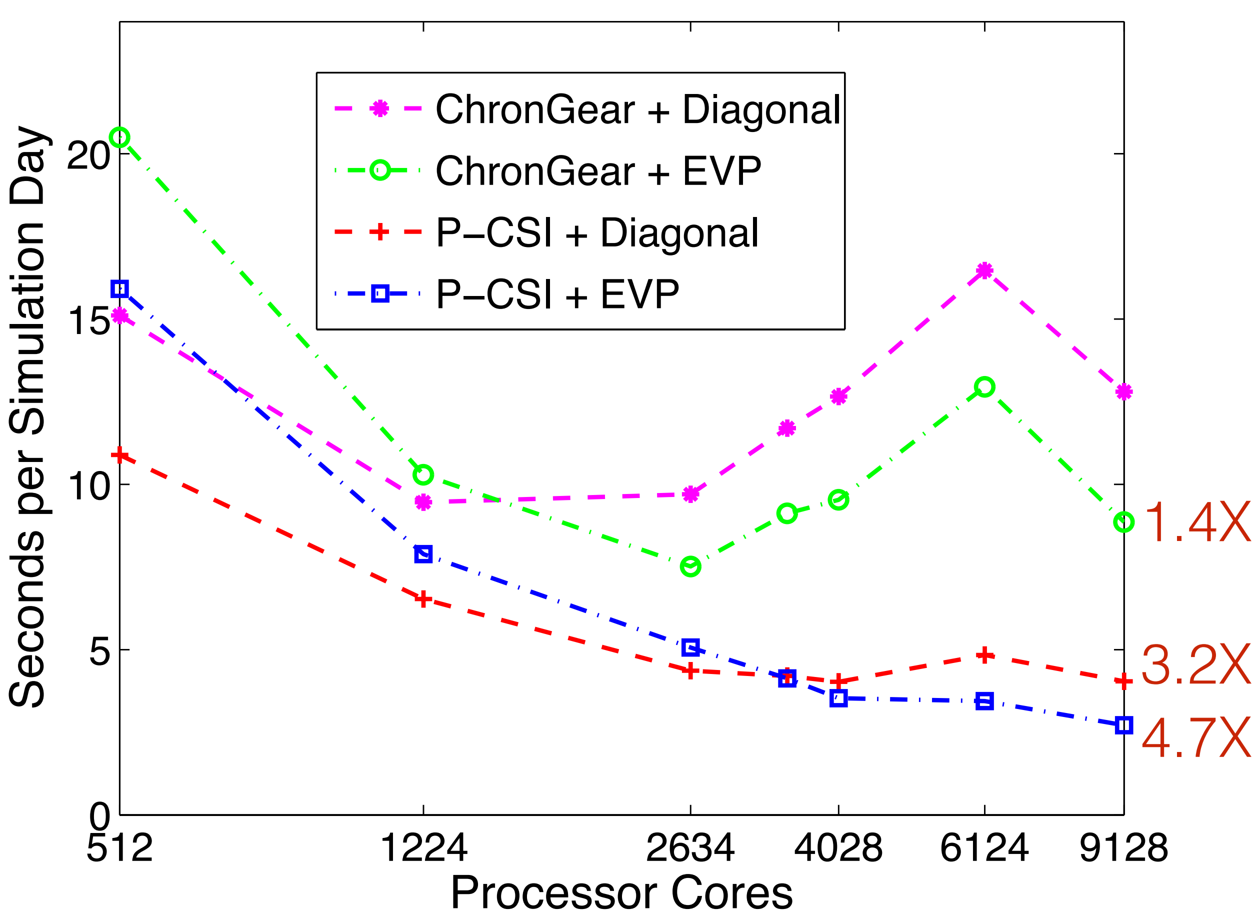
| Resolution | 1 degree | | 0.1 degree | |
|---|---|---|---|---|
| Solvers | ChronGear | P-CSI | ChronGear | P-CSI |
| NONE | 540 | 570 | 150 | 160 |
| DIAG | 240 | 300 | 100 | 120 |
| EVP | 160 | 200 | 50 | 60 |

# Scalability on Yellowstone

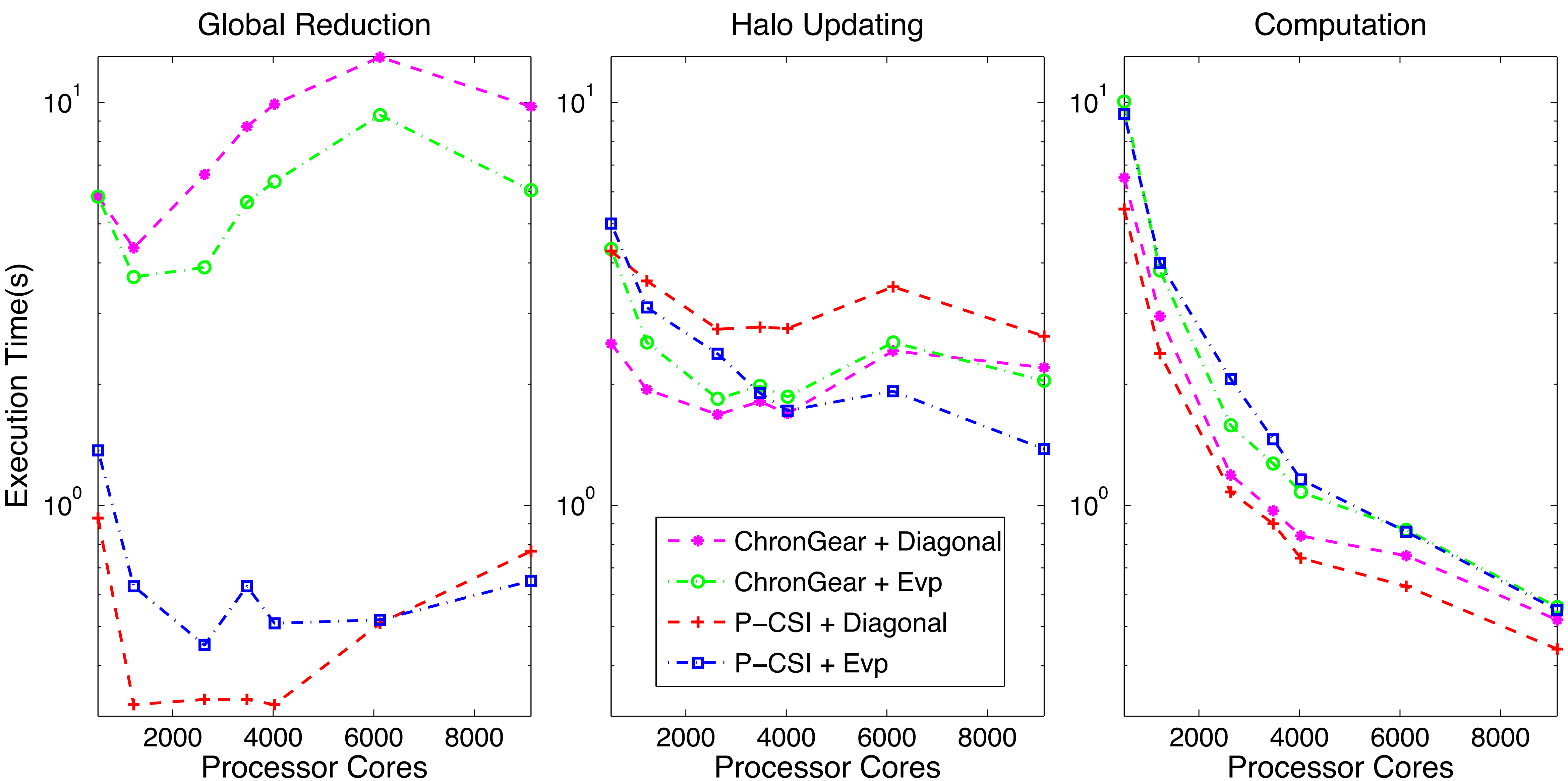
## 1 degree POP




## 0.1 degree POP



# Solver Component



# Simulated Years for Solver Per wall-clock Day

Simulation rate without considering the I/O and initialization in 0.1 degree POP.

| Number of cores | 512 | 1224 | 2634 | 3476 | 4028 | 6124 | 9128 |
|---|---|---|---|---|---|---|---|
| ChronGear | 0.71 | 1.69 | 3.31 | 3.83 | 4.28 | 4.72 | 6.02 |
| ChronGear+Evp | 0.70 | 1.68 | 3.42 | 3.99 | 4.53 | 5.08 | 6.69 |
| P-CSI+Diagonal | 0.72 | 1.72 | 3.58 | 4.36 | 5.07 | 6.15 | 7.75 |
| P-CSI+Evp | 0.71 | 1.70 | 3.54 | 4.36 | 5.12 | 6.38 | 8.10 |

35%

Q & A

Thanks!