BandFFT parallelization of ABINIT

François Bottin & Gilles Zérah

CEA/DIF, BP 12 Bruyères-le-Châtel
Département de Physique Théorique et Appliquée
Service de Physique de la Matière Condensée

3rd International ABINIT Developer Workshop
Liège (Belgium), 29-31 January 2007

Parallelization: one of the main computational tasks

- Large supercell *ab initio* calculations are very time consuming.

Parallelization: one of the main computational tasks

- Large supercell *ab initio* calculations are very time consuming.
- Increase of the powerful and number of processors of supercomputers

Site	Computer	Proc.	Tflops
DOE/NNSA/LLNL (US)	BlueGene	131072	281
NNSA/Sandia National Laboratories (US)	Red Storm	26544	101
IBM T. J. Watson Research Center (US)	BGW	40960	91
DOE/NNSA/LLNL (US)	ASC Purple	12208	76
Bacelona Supercomputing Center (Spain)	Mare Nostrum	10240	63
NNSA/Sandia National Laboratories (US)	Thunderbird	9024	53
CEA/DAM (France)	TERA-10	9968	53
NASA/Ames Research Center/NAS (US)	Columbia	10160	52
GSIC Center (Japan)	TSUBAME	11088	47
Oak Ridge National Laboratories (US)	Jaguar	10424	43

Parallelization: one of the main computational tasks

- Large supercell *ab initio* calculations are very time consuming.
- Increase of the powerful and number of processors of supercomputers

Site	Computer	Proc.	Tflops
DOE/NNSA/LLNL (US)	BlueGene	131072	281
NNSA/Sandia National Laboratories (US)	Red Storm	26544	101
IBM T. J. Watson Research Center (US)	BGW	40960	91
DOE/NNSA/LLNL (US)	ASC Purple	12208	76
Bacelona Supercomputing Center (Spain)	Mare Nostrum	10240	63
NNSA/Sandia National Laboratories (US)	Thunderbird	9024	53
CEA/DAM (France)	TERA-10	9968	53
NASA/Ames Research Center/NAS (US)	Columbia	10160	52
GSIC Center (Japan)	TSUBAME	11088	47
Oak Ridge National Laboratories (US)	Jaguar	10424	43

- Aim: to use and to be efficient on these supercomputers.

1 Theoretical background

- *The Norm-Conserving method (NC)*
- *The Projector Augmented-Wave method (PAW)*
- *Locally Optimal Block Preconditioned Conjugate Gradient method (*LOBPCG*)*
- *The self-consistent loop (SCF)*

2 BandFFT parallelization

- *Hypothesis – Requirements*
- *Standard implementation*

3 BandFFT Results

- *Benchmarks*
- *Beyond the standard implementation - Optimisations*
- *Norm-conserving Results*
- *PAW results*
- *How to use the BandFFT parallelization*

4 Conclusion – Perspectives

- *Conclusion*
- *Perspectives*

The Norm-Conserving method (NC)

Hamiltonian

$$\tilde{\mathcal{H}} = -\frac{\Delta}{2} + \mathrm{v_{loc}} + \sum_{\mathrm{I}} \sum_{\mathrm{lm}\in\mathrm{I}} \frac{|\tilde{\mathcal{P}}^{\mathrm{I}}_{\mathrm{lm}}\rangle\langle\tilde{\mathcal{P}}^{\mathrm{I}}_{\mathrm{lm}}|}{\langle\tilde{\mathcal{P}}^{\mathrm{I}}_{\mathrm{lm}}|\tilde{\Phi}^{\mathrm{I}}_{\mathrm{lm}}\rangle} \quad \text{with} \quad \tilde{\mathcal{H}}\tilde{\Psi}_{\mathrm{nk}} = \epsilon_{\mathrm{nk}}\tilde{\Psi}_{\mathrm{nk}}$$

Density

$$\tilde{\mathrm{n}}(\mathbf{r}) = \sum_{\mathrm{nk}} \mathrm{f_{nk}} |\tilde{\Psi}_{\mathrm{nk}}(\mathbf{r})|^2$$

The Projector Augmented-Wave method (PAW)

Hamiltonian

$$\tilde{\mathcal{H}} = -\frac{\Delta}{2} + \mathrm{v_{loc}} + \sum_{\mathrm{I}} \sum_{\mathrm{ij}\in\mathrm{I}} |\tilde{\mathcal{P}}_{\mathrm{i}}^{\mathrm{I}}\rangle \mathrm{D}_{\mathrm{ij}}^{\mathrm{I}} \langle\tilde{\mathcal{P}}_{\mathrm{j}}^{\mathrm{I}}| \quad \text{with} \quad \tilde{\mathcal{H}}\tilde{\Psi}_{\mathrm{nk}} = \epsilon_{\mathrm{nk}} \mathcal{O} \tilde{\Psi}_{\mathrm{nk}}$$

Densities – Overlap operator

$$\hat{\mathrm{n}}(\mathbf{r}) \quad \text{and} \quad \rho_{\mathrm{ij}}^{\mathrm{I}} = \sum_{\mathrm{nk}} \mathrm{f_{nk}} \langle\tilde{\Psi}_{\mathrm{nk}}|\tilde{\mathcal{P}}_{\mathrm{i}}^{\mathrm{I}}\rangle\langle\tilde{\mathcal{P}}_{\mathrm{j}}^{\mathrm{I}}|\tilde{\Psi}_{\mathrm{nk}}\rangle$$

$$\mathcal{O} = \mathcal{I} + \sum_{\mathrm{I}} \sum_{\mathrm{ij}\in\mathrm{I}} |\tilde{\mathcal{P}}_{\mathrm{i}}^{\mathrm{I}}\rangle (\langle\Phi_{\mathrm{i}}^{\mathrm{I}}|\Phi_{\mathrm{j}}^{\mathrm{I}}\rangle - \langle\tilde{\Phi}_{\mathrm{i}}^{\mathrm{I}}|\tilde{\Phi}_{\mathrm{j}}^{\mathrm{I}}\rangle) \langle\tilde{\mathcal{P}}_{\mathrm{j}}^{\mathrm{I}}|$$

$$\mathrm{n}(\mathbf{r}) = \tilde{\mathrm{n}}(\mathbf{r}) + \sum_{\mathrm{I}} (\mathrm{n}^{1,\mathrm{I}}(\mathbf{r}) - \tilde{\mathrm{n}}^{1,\mathrm{I}}(\mathbf{r}))$$

Algorithm 1 LOBPCG

Require: $\mathbf{\Psi}^0 = \{\Psi_1^0, \ldots, \Psi_m^0\}$ close to the minimum and K a preconditioner; $\mathbf{P} = \{P_1^{(0)}, \ldots, P_m^0\}$ is initialized to 0.

1: **for** i=0,1,...,κ **do**
2: $\quad \Upsilon^{(i)} = \Upsilon(\mathbf{\Psi}^{(i)})$
3: $\quad \mathbf{R}^{(i)} = \mathcal{H}\mathbf{\Psi}^{(i)} - \Upsilon^{(i)}\mathcal{O}\mathbf{\Psi}^{(i)}$
4: $\quad \mathbf{W}^{(i)} = K\mathbf{R}^{(i)}$
5: $\quad$ The Rayleigh-Ritz method is applied within the subspace $\Xi = \left\{P_1^{(i)}, \ldots, P_m^{(i)}, \Psi_1^{(i)}, \ldots, \Psi_m^{(i)}, W_1^{(i)}, \ldots, W_m^{(i)}\right\}$
6: $\quad \mathbf{\Psi}^{(i+1)} = \Delta^{(i)}\mathbf{\Psi}^{(i)} + \Lambda^{(i)}\mathbf{W}^{(i)} + \Gamma^{(i)}\mathbf{P}^{(i)}$
7: $\quad \mathbf{P}^{(i+1)} = \Lambda^{(i)}\mathbf{W}^{(i)} + \Gamma^{(i)}\mathbf{P}^{(i)}$
8: **end for**

More efficient than CG in many cases [1]

[1] A. Knyazev *Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method*, SIAM Journal on Scientific Computing **23**, 517 (2001).

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \tilde{\mathcal{H}} | \tilde{\Psi}_n \rangle = \epsilon_n \langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \mathcal{O} | \tilde{\Psi}_n \rangle$$

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \tilde{\mathcal{H}} | \tilde{\Psi}_n \rangle = \epsilon_n \langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \mathcal{O} | \tilde{\Psi}_n \rangle$$

Time consuming parts

- The non-local terms (`nonlop`).

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

Time consuming parts

- The non-local terms (`nonlop`).
- The resolution of the KS equations (`lobpcg`).

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \tilde{\mathcal{H}} | \tilde{\Psi}_n \rangle = \epsilon_n \langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \mathcal{O} | \tilde{\Psi}_n \rangle$$

Time consuming parts

- The non-local terms (`nonlop`).
- The resolution of the KS equations (`lobpcg`).
- The diagonalisation within the sub-space (`subdiago`+`orthon`).

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \tilde{\mathcal{H}} | \tilde{\Psi}_n \rangle = \epsilon_n \langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \mathcal{O} | \tilde{\Psi}_n \rangle$$

Time consuming parts

- The non-local terms (`nonlop`).
- The resolution of the KS equations (`lobpcg`).
- The diagonalisation within the sub-space (`subdiago+orthon`).
- The calculation of the density and local potential (FFT).

SCF loop

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

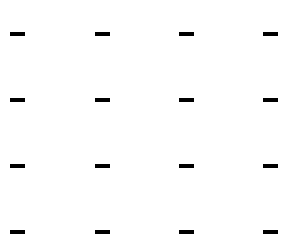
$$\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \tilde{\mathcal{H}} | \tilde{\Psi}_n \rangle = \epsilon_n \langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \mathcal{O} | \tilde{\Psi}_n \rangle$$

Famous parallelizations

- over the **k**-points.
- over the bands → require a block eigensolver.
- over the plane-waves → require a parallel 3dim-FFT algorithm.

Hypothesis – Requirements

- M bands and a P^3 FFT grid
- Two-dimensional m×p grid of processors with m "band-processors" and p "FFT-processors".



- m and p have to be commensurate with M and P, respectively
- For the 3dim-FFT one have to ensure: $p \leq \frac{P}{2}$

Standard implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \tilde{\mathcal{H}} | \tilde{\Psi}_n \rangle = \epsilon_n \langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \mathcal{O} | \tilde{\Psi}_n \rangle$$

All the coefficients are on the proc. 0

$$\begin{matrix} c_n(\mathbf{g}) & - & - & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{matrix}$$

Standard implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \tilde{\mathcal{H}} | \tilde{\Psi}_n \rangle = \epsilon_n \langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \mathcal{O} | \tilde{\Psi}_n \rangle$$

Distribution over the p "FFT-processors"

$c_n(g_1)$	$c_n(g_2)$	$\dots$	$c_n(g_p)$
–	–	–	–
–	–	–	–
–	–	–	–

Standard implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

Distribution over the m "band-processors"

$$\begin{matrix} c_n(g_{11}) & c_n(g_{12}) & \dots & c_n(g_{1p}) \\ c_n(g_{21}) & c_n(g_{22}) & \dots & c_n(g_{2p}) \\ \vdots & \vdots & \ddots & \vdots \\ c_n(g_{m1}) & \dots & \dots & c_n(g_{mp}) \end{matrix}$$

Standard implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

In `lobpcg`: blocks of size m

$$\begin{matrix} c_{1:m}(g_{11}) & c_{1:m}(g_{12}) & \dots & c_{1:m}(g_{1p}) \\ c_{1:m}(g_{21}) & c_{1:m}(g_{22}) & \dots & c_{1:m}(g_{2p}) \\ \vdots & \vdots & \ddots & \vdots \\ c_{1:m}(g_{m1}) & \dots & \dots & c_{1:m}(g_{mp}) \end{matrix}$$

Standard implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

Within each column...

$$\left| \begin{array}{c|ccc} c_{1:m}(g_{11}) & c_{1:m}(g_{12}) & \cdots & c_{1:m}(g_{1p}) \\ c_{1:m}(g_{21}) & c_{1:m}(g_{22}) & \cdots & c_{1:m}(g_{2p}) \\ \vdots & \vdots & \ddots & \vdots \\ c_{1:m}(g_{m1}) & \cdots & \cdots & c_{1:m}(g_{mp}) \end{array} \right.$$

Standard implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

... one have a m×m matrix...

$$\begin{vmatrix} c_1(g_{11}) & c_2(g_{11}) & \dots & c_m(g_{11}) \\ c_1(g_{21}) & c_2(g_{21}) & \dots & c_m(g_{21}) \\ \vdots & \vdots & \ddots & \vdots \\ c_1(g_{m1}) & \dots & \dots & c_m(g_{m1}) \end{vmatrix}$$

Standard implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

... which can be transposed...

$$\begin{vmatrix} c_1(g_{11}) & c_1(g_{21}) & \dots & c_1(g_{m1}) \\ c_2(g_{11}) & c_2(g_{21}) & \dots & c_2(g_{m1}) \\ \vdots & \vdots & \ddots & \vdots \\ c_m(g_{11}) & \dots & \dots & c_m(g_{m1}) \end{vmatrix}$$

Standard implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

... and gathered in one column...

$$\begin{vmatrix} c_1(g_1) \\ c_2(g_1) \\ \vdots \\ c_m(g_1) \end{vmatrix}$$

Standard implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

... to give the final distribution.

$$\left| \begin{array}{c|ccc} c_1(g_1) & c_1(g_{12}) & \dots & c_1(g_p) \\ c_2(g_1) & c_2(g_{22}) & \dots & c_2(g_p) \\ \vdots & \vdots & \ddots & \vdots \\ c_m(g_1) & \dots & \dots & c_m(g_p) \end{array} \right|$$

Standard implementation

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

We can apply a parallel 3dim-FFT on each line

$$\begin{matrix} c_1(g_1) & c_1(g_2) & \dots & c_1(g_p) \\ c_2(g_1) & c_2(g_2) & \dots & c_2(g_p) \\ \vdots & \vdots & \ddots & \vdots \\ c_m(g_1) & \dots & \dots & c_m(g_p) \end{matrix}$$

Benchmarks – Test cases

- Gold system with 108 atoms, 648 bands, 108^3 FFT grid, 1 **k**-point and E_{cut}=24 Ha.
- We stop the SCF for nstep=15
- A two-dimensional grid of processors with nproc=1, 4, 18, 36, 54, 108, 162 and 216.
- Example: for nproc=108, we can choose m×p=108×1, 54×2, 36×3, 27×4, 18×6, 12×9, 9×12, 6×18, 4×27 and 3×36
- Test are performed on 2 supercomputers:

Supercomputer	Node	Interconnection
Tantale (CCRT)	4-procs AMD OPTERON 2.4 GHz	Infiniband
TERA-10 (CEA/DIF)	Novascale 16-procs Intel Itanium	Quadrics

Beyond the standard implementation - Optimisations

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

The LAPACK and BLAS libraries

- In `lobpcg`: timing of `zgemm` for 54 processors

	`zgemm`/Total	Ratio
Standard libraries	80000/150000	53%
Optimized libraries	10000/100000	10%

Beyond the standard implementation - Optimisations

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow$$

$$\left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \tilde{\mathcal{H}} \middle| \tilde{\Psi}_n \right\rangle = \epsilon_n \left\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} \middle| \mathcal{O} \middle| \tilde{\Psi}_n \right\rangle$$

The SCALAPACK library

- In `vtowfk`: timing of `subdiago` for 216 processors.

	648 bands	1296 bands
Standard implementation	8%	23%
SCALAPACK library	3%	2%

Beyond the standard implementation - Optimisations

$$\tilde{\Psi}_n(\mathbf{r}) = \sum_{\mathbf{G}} c_n(\mathbf{G}) e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}}$$

$$\downarrow$$

$$[\tilde{n} + \hat{n}](\mathbf{r}) \quad \text{and} \quad \rho_{ij} \longleftarrow \{c_n(\mathbf{G}); \epsilon_n\}$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

$$v_{loc}(\mathbf{r}) \quad \text{and} \quad v_{nl}(\mathbf{r}) \qquad\qquad | \tilde{\mathcal{H}} - \epsilon_n \mathcal{O} | = 0$$

$$\downarrow \qquad\qquad\qquad\qquad \uparrow$$

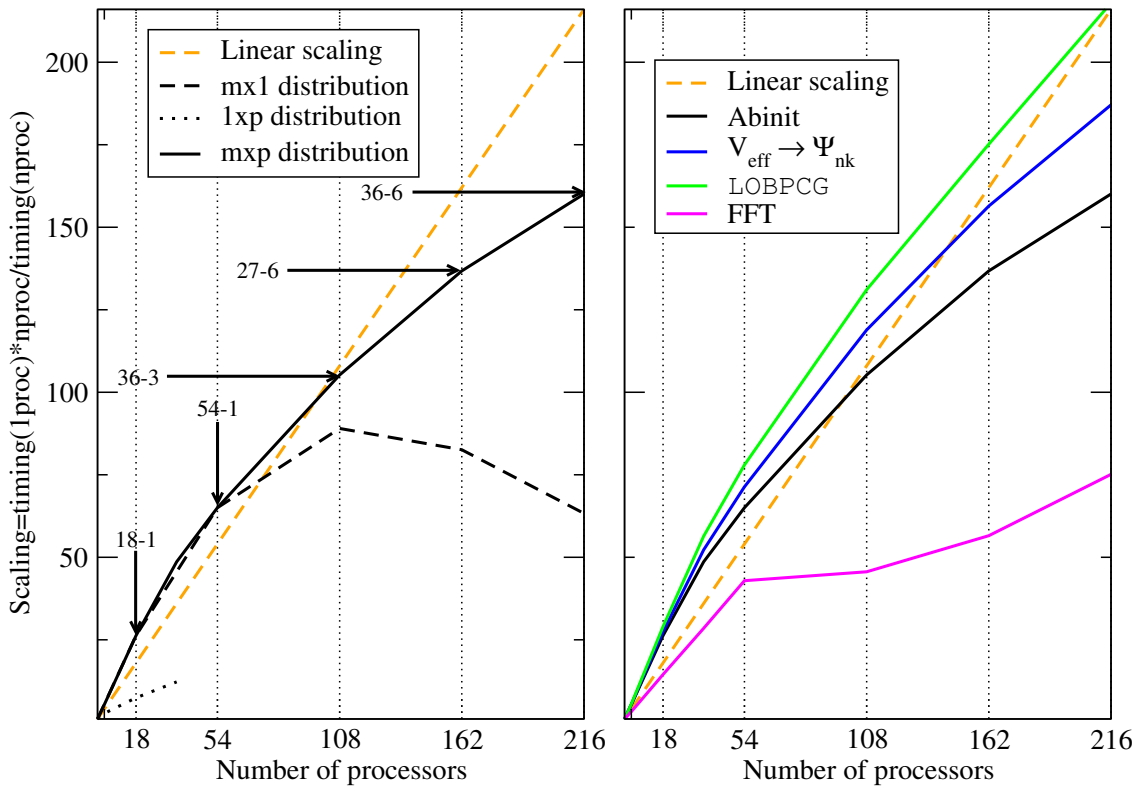
$$\langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \tilde{\mathcal{H}} | \tilde{\Psi}_n \rangle = \epsilon_n \langle e^{i(\mathbf{k}+\mathbf{G}).\mathbf{r}} | \mathcal{O} | \tilde{\Psi}_n \rangle$$

Generalisation of the transposition principle

- Before each call to nonlop we transpose the distribution and use a "FFT distribution".
- Minimize the number of collective communications.
- Permits to avoid bad load balancing.

Norm-conserving Results

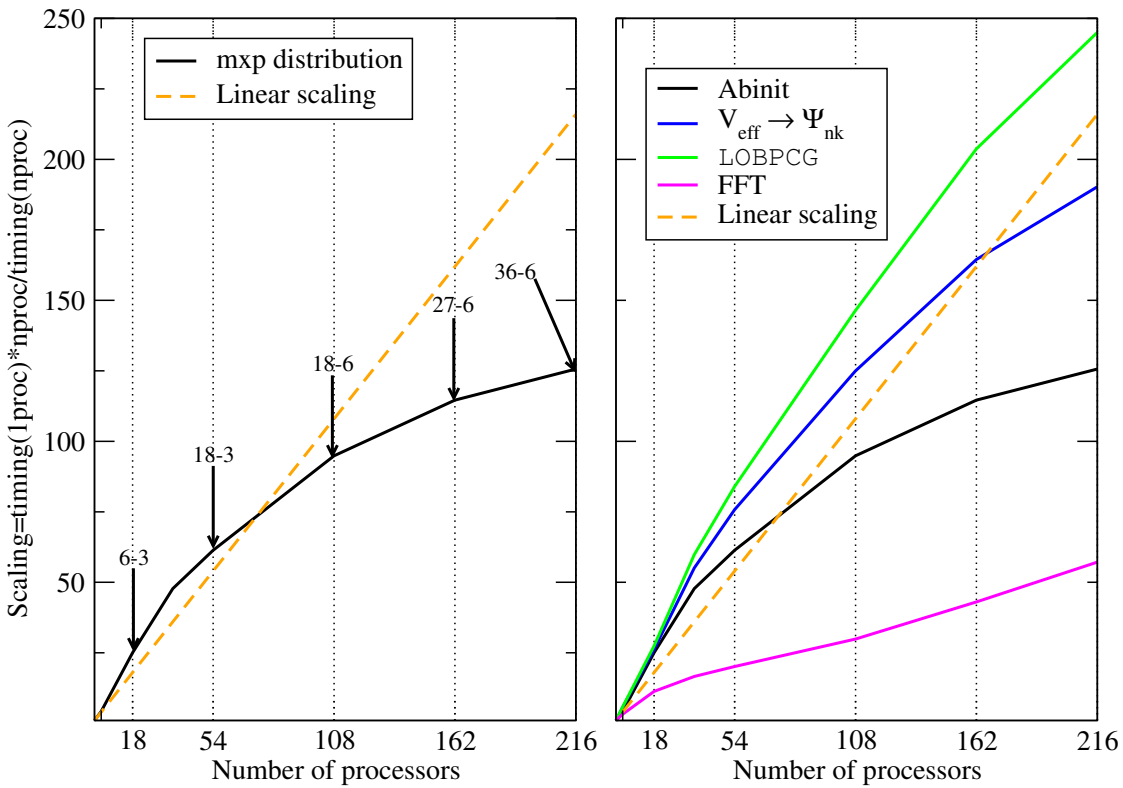
108 atoms, 648 bands, 108^3 FFT grid, 1 **k**-point and E_{cut}=24 Ha.



- In sequential: 90 000 sec. (with 90% in `lobpcg`).
- Linear scaling up to 100 for ABINIT and 200 for `lobpcg`.

PAW results (with 1 projector per angular momentum)

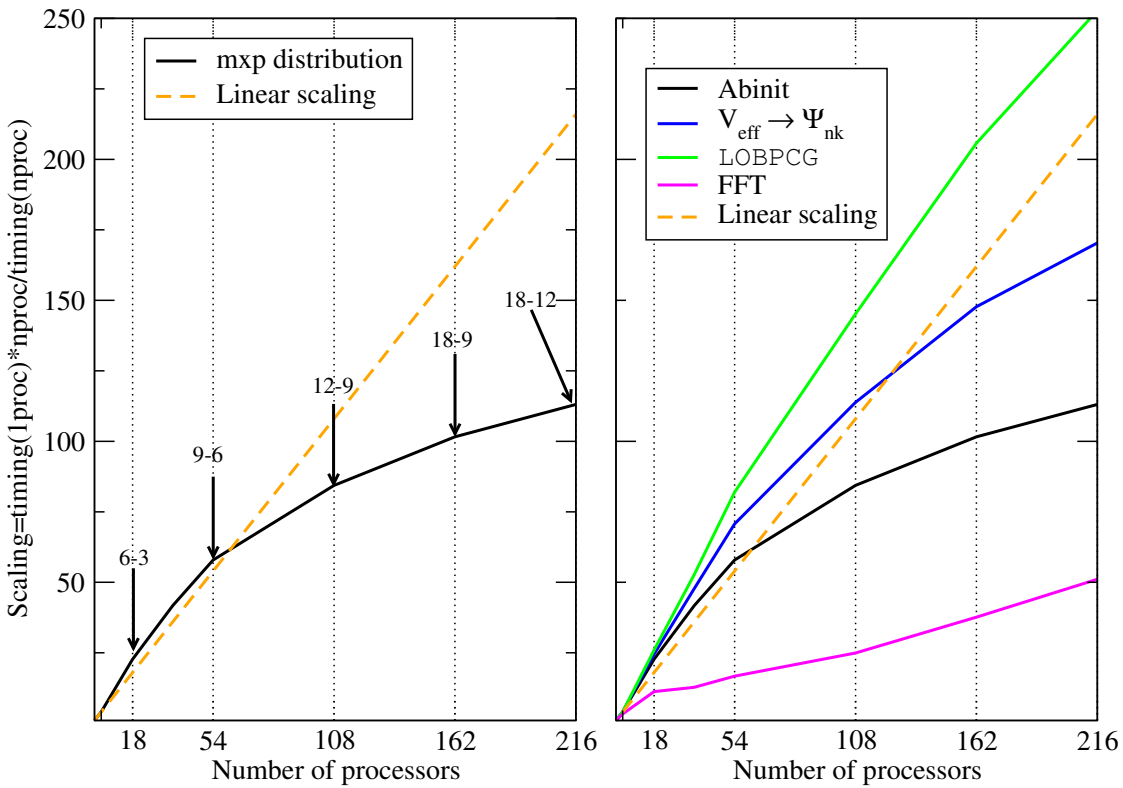
108 atoms, 648 bands, 72^3 FFT grid, 1 **k**-point and E_{cut}=12 Ha.



- In sequential: 60 000 sec. (with 90% in `lobpcg`).
- We loose efficiency between `lobpcg`, `vtowfk` and ABINIT.

PAW results (with 2 projectors per angular momentum)

108 atoms, 648 bands, 72^3 FFT grid, 1 **k**-point and E_{cut}=12 Ha.



- In sequential: 75 000 sec. (with 90% in `lobpcg`).
- Due to: `nonlop_ylm`, the double grid and spherical terms.

Compilation and input variables

- Compile with the -DMPI_FFT and -DSCALAPACK preprocessing flags.
- Link to the optimized librairies (LAPACK, BLAS and SCALAPACK) on your plateform.
- npband=m and npfft=p for the band- and FFT-processors.
- wfoptalg=4: the `lobpcg` 1 method
- fft_opt_lob=2: the generalisation of the transposition principle.
- fftalg=401: the 3dim parallel FFT of Goedecker *et al.* [2].

[2] S. Goedecker, M. Boulet and T. Deutsch *An efficient 3-dim FFT for plane wave electronic structure calculations on massively parallel machines composed of multiprocessor nodes*, Comput. Phys. Comm. **154**, 105 (2003).

Restrictions: ground state only calculations

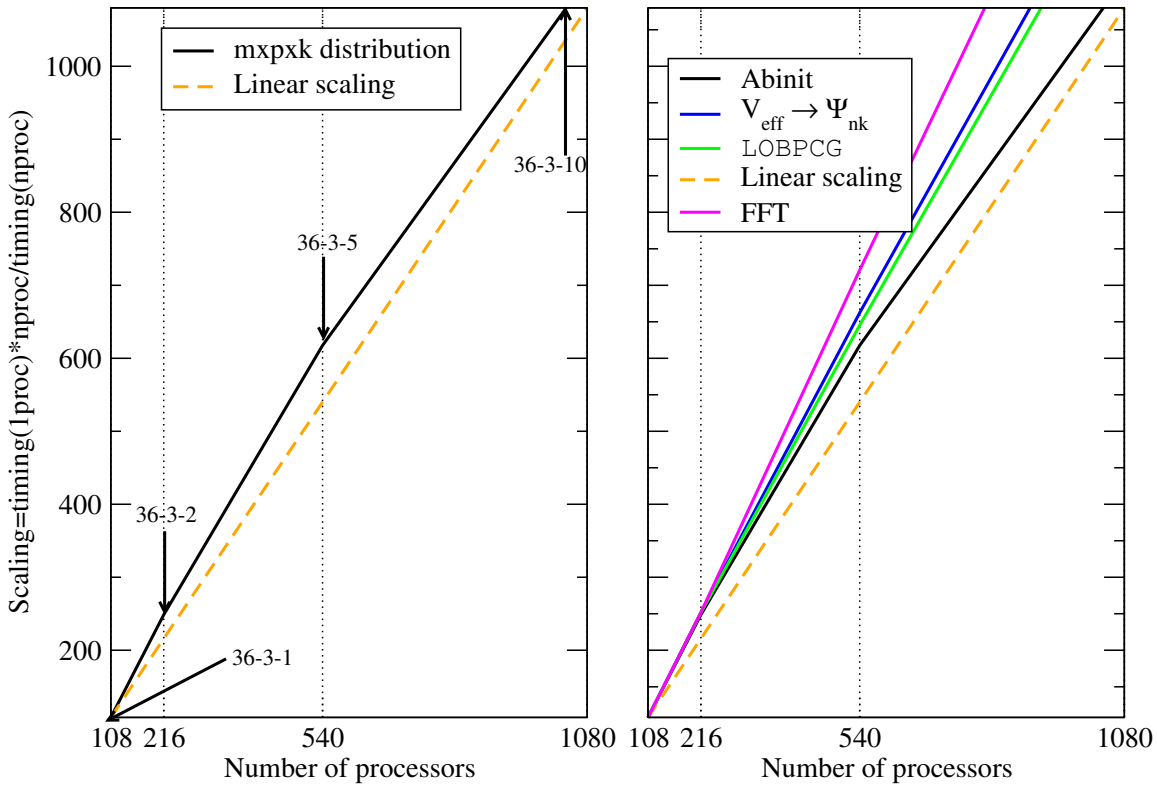
- iprcch=0. Other values are allowed for testing purposes but the parallelization is then very poor. For large systems the correction of forces (iprcch=1) and the prediction of the density at each MD step (iprcch=2) lead to memory overflow and swap.
- istwfk=nkpt*1. The time-reversal symetry is not yet compatible with the bandFFT parallelization.
- nsppol=nspden=1. No spin-polarized (or antiferromagnetic) calculations.
- useylm=0. No PAW calculations.
- iscf> 0. Non self-consistent calculations are not tested.
- I/O is not yet available.

Conclusion

- ABINIT scales lineary up to 100 processors in NC and PAW.
- `lobpcg` is responsible for the superlinear behaviour and scales perfectly up to 200 processors.
- The BandFFT parallelization have to be optimized in the framework of PAW calculations.
- TODO: Retrieve all the functionalities of ABINIT (spin polarization, NSCF calculations, I/O ...)

Perspectives: the triple nkG parallelization (in 5.4.x)

108 atoms, 648 bands, 108^3 FFT grid, 10 **k**-point and E_{cut}=24 Ha.



Linear scaling of ABINIT up to 1000 processors.