Solving the Convection Diffusion Equation on Distributed Systems

N. Missirlis, F. Tzaferis, G. Karagiorgos,
A. Theodorakos, A. Kontarinis, A. Konsta

Department of Informatics
and Telecommunications
University of Athens

http://parallel.di.uoa.gr

This presentation

- Considers the local MSOR
- Determines the optimum parameters.
- Parallel Implementation.
- Considers the Load Balancing Problem
- Results

The problem

- Assume a $\sqrt{N} \times \sqrt{N}$ array of processors and assign a processor to each grid point.
- Jacobi is highly parallel but

  $$T = O(N) \quad \text{(local communication)}$$

- SOR is difficult to parallelize
  - Needs multicoloring
  - Adaptive use of ω needs global communication and results in

    $$T = O(N)$$

  same as Jacobi!

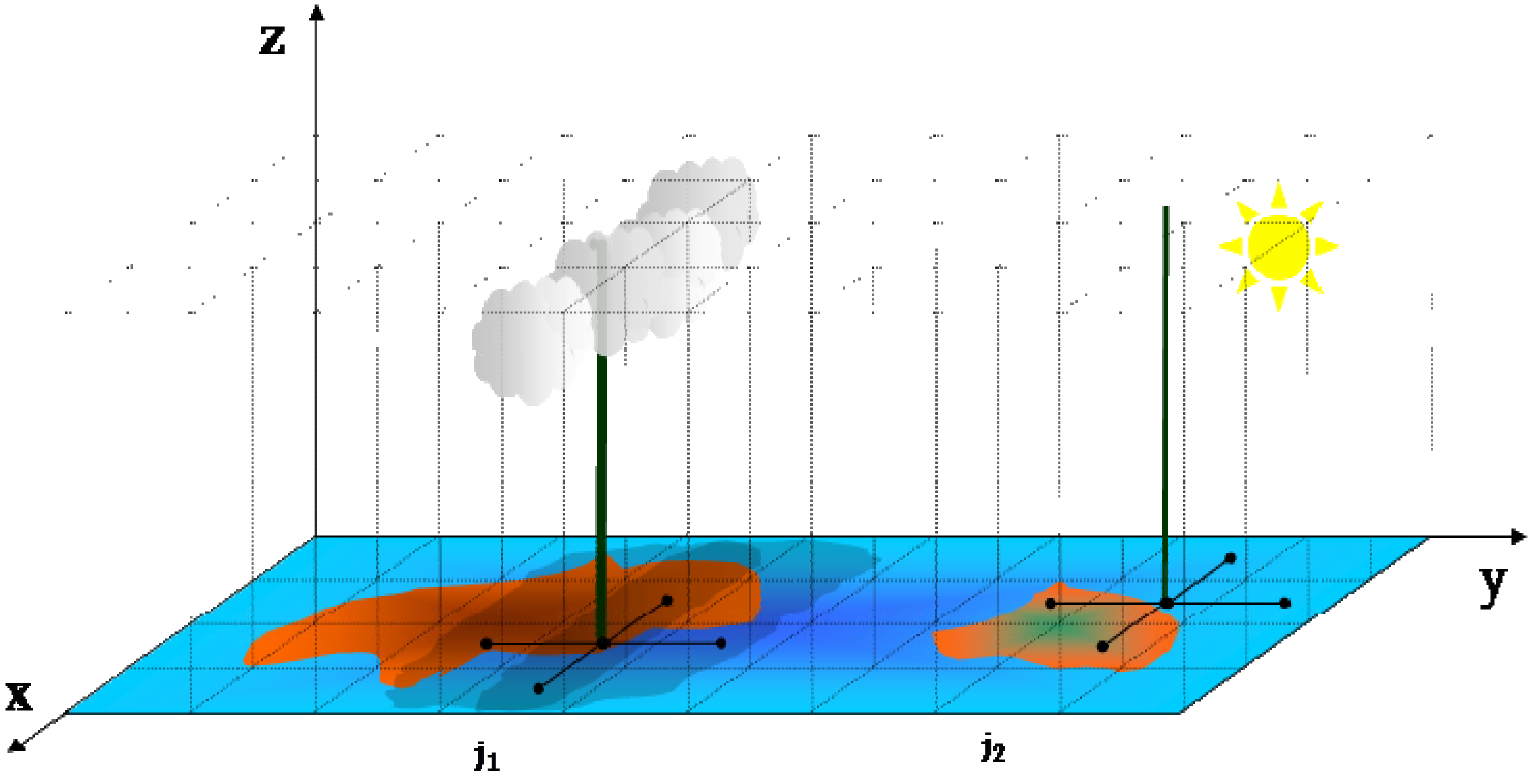
The solution

- Use local iterative methods (Ehrlich, Botta, Russel)
  - Use of ω_{ij} for each grid point.
  - Local communication
  - $T = O(\sqrt{N})$

Related research work

- Local SOR (Kuo et. Al. 1987)
- Local MSOR (Boukas and Missirlis 1997).
- Local SI-MEGS (Missirlis and Tzaferis 2002).

Introduction



Introduction (Convection Diffusion Equation)

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}-f(x,y)\frac{\partial u}{\partial x}-g(x,y)\frac{\partial u}{\partial y}=0 \qquad (1)$$

Discretization

$$\begin{aligned} u_{ij} &= \ell_{ij}u_{i-1,j}+r_{ij}u_{i+1,j}+t_{ij}u_{i,j+1}+b_{ij}u_{i,j-1} \\ i &= 1,2,\dots,N \qquad j=1,2,\dots,N. \end{aligned} \qquad (2)$$

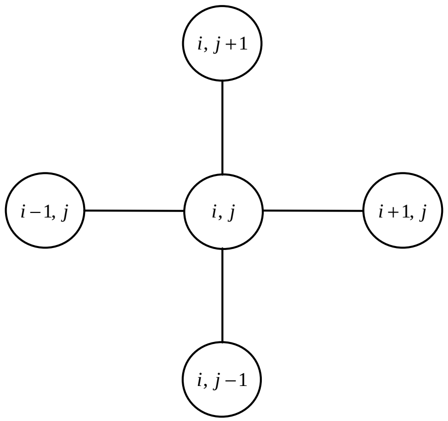
Introduction …

$$Ax = b$$

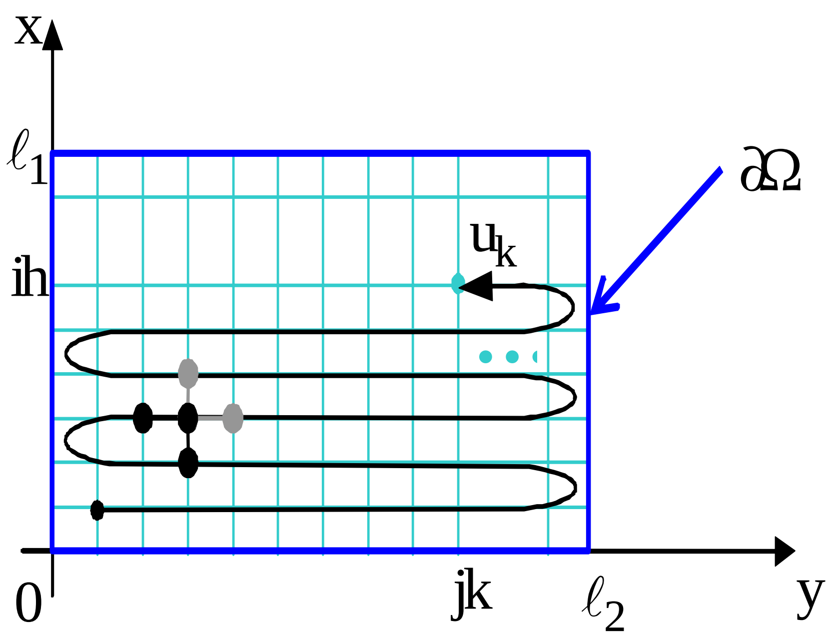
Jacobi

$$u_{ij}^{(n+1)} = \ell_{ij} u_{i-1,j}^{(n)} + r_{ij} u_{i+1,j}^{(n)} + t_{ij} u_{i,j+1}^{(n)} + b_{ij} u_{i,j-1}^{(n)}$$

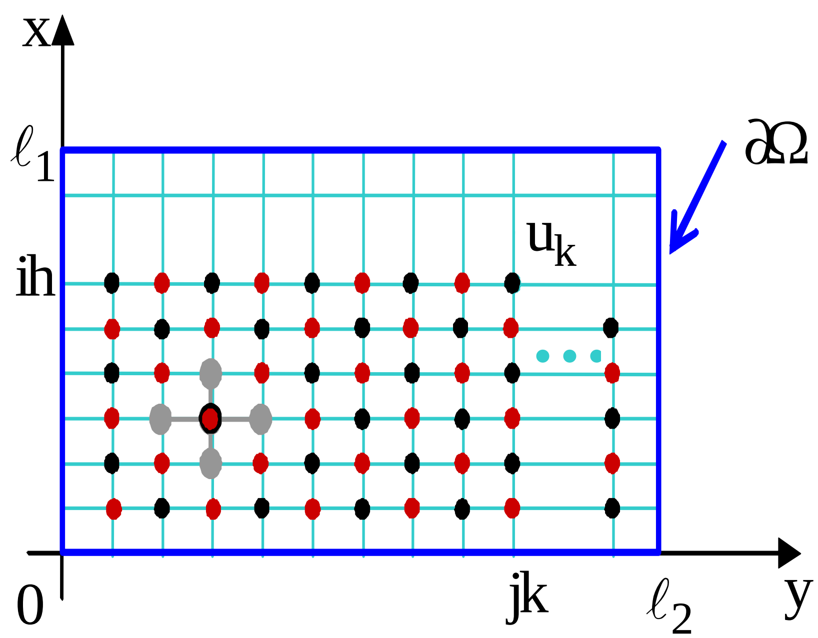
5-point stencil



Ordering



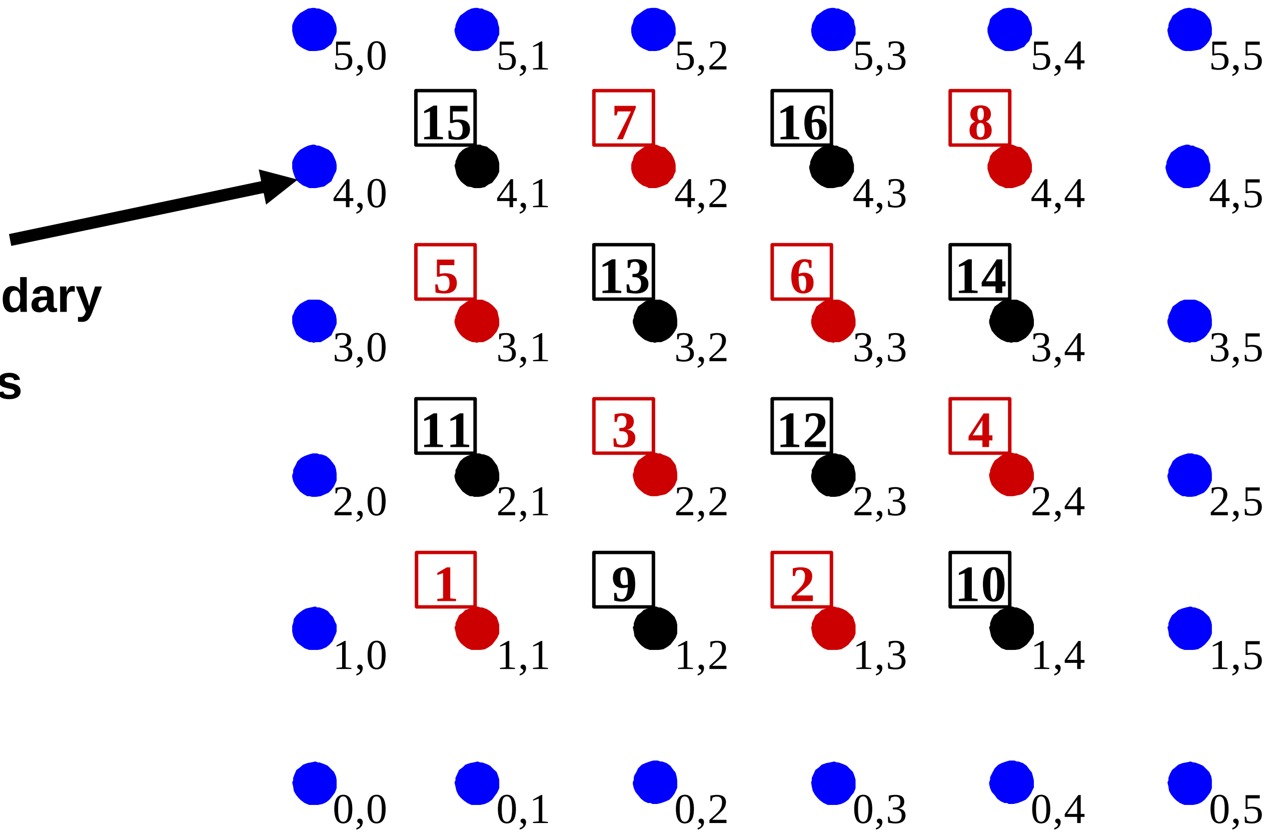
natural



red-black

R/B ordering

Boundary points



Local Modified ESOR

$$u_{ij}^{(n+1)} = (1-\tau_{ij})u_{ij}^{(n)} + \tau_{ij} J_{ij} u_{ij}^{(n)} \qquad \text{(i+j even)}$$

$$u_{ij}^{(n+1)} = (1-\tau'_{ij})u_{ij}^{(n)} + \omega'_{ij} J_{ij} u_{ij}^{(n+1)} + (\tau'_{ij} - \omega'_{ij}) J_{ij} u_{ij}^{(n)} \qquad \text{(i+j odd)}$$

where

$$J_{ij} u_{ij}^{(n)} = \ell_{ij} u_{i-1,j}^{(n)} + r_{ij} u_{i+1,j}^{(n)} + t_{ij} u_{i,j+1}^{(n)} + b_{ij} u_{i,j-1}^{(n)}$$

Local Modified SOR

$$u_{ij}^{(n+1)} = (1-\omega_{ij})u_{ij}^{(n)} + \omega_{ij} J_{ij} u_{ij}^{(n)}$$ (i+j even)

$$u_{ij}^{(n+1)} = (1-\omega'_{ij})u_{ij}^{(n)} + \omega'_{ij} J_{ij} u_{ij}^{(n+1)}$$ (i+j odd)

Optimum Parameters

The Jacobi operator J_{ij} has real eigenvalues

$$\omega_{ij} = \widehat{\omega}_{1,i,j} := \frac{2}{1 - \overline{\mu}_{ij}\underline{\mu}_{ij} + \sqrt{(1 - \overline{\mu}_{ij}^2)(1 - \underline{\mu}_{ij}^2)}}$$

$$\omega'_{ij} = \widehat{\omega}_{2,i,j} := \frac{2}{1 + \overline{\mu}_{ij}\underline{\mu}_{ij} + \sqrt{(1 - \overline{\mu}_{ij}^2)(1 - \underline{\mu}_{ij}^2)}}$$

Optimum Parameters

The Jacobi operator J_{ij} has imaginary eigenvalues

$$\omega_{ij} = \widehat{\omega}_{1,i,j} := \frac{2}{1 - \overline{\mu}_{ij}\underline{\mu}_{ij} + \sqrt{(1+\overline{\mu}_{ij}^2)(1+\underline{\mu}_{ij}^2)}}$$

$$\omega'_{ij} = \widehat{\omega}_{2,i,j} := \frac{2}{1 + \overline{\mu}_{ij}\underline{\mu}_{ij} + \sqrt{(1+\overline{\mu}_{ij}^2)(1+\underline{\mu}_{ij}^2)}}$$

Optimum values for complex eigenvalues ?

Open Problem

Optimum Parameters

If $\underline{\mu}_{ij} = 0$ then Local MSOR = Local SOR

$$\omega_{ij} = \frac{2}{1 + \sqrt{1 - \overline{\mu}_{ij}^2}}$$

real

$$\omega_{ij} = \frac{2}{1 + \sqrt{1 + \overline{\mu}_{ij}^2}}$$

complex

Local Modified SOR

Real

$$S\left(\Im_{\widehat{\omega}_{1,i,j},\widehat{\omega}_{2,i,j}}\right)=\sqrt{(\widehat{\omega}_{1,i,j}-1)(\widehat{\omega}_{2,i,j}-1)}=\frac{\sqrt{1-\overline{\mu}_{ij}^{2}}-\sqrt{1-\underline{\mu}_{ij}^{2}}}{\sqrt{1-\overline{\mu}_{ij}^{2}}+\sqrt{1-\underline{\mu}_{ij}^{2}}}$$

Imaginary

$$S\left(\Im_{\widehat{\omega}_{1,i,j},\widehat{\omega}_{2,i,j}}\right)=\sqrt{(1-\widehat{\omega}_{1,i,j})(1-\widehat{\omega}_{2,i,j})}=\frac{\sqrt{1+\overline{\mu}_{ij}^{2}}-\sqrt{1+\underline{\mu}_{ij}^{2}}}{\sqrt{1+\overline{\mu}_{ij}^{2}}+\sqrt{1+\underline{\mu}_{ij}^{2}}}$$

Local Modified SOR

Conclusion

Local MSOR will attain at least the convergence rate of local SOR, whereas its convergence rate will increase as $\underline{\mu}_{ij}$ increases.

Numerical Results

The coefficients used in each problem are:

1. $f(x,y) = \mathrm{Re}(2x-10)^3$, $g(x,y) = \mathrm{Re}(2y-10)^3$
2. $f(x,y) = \mathrm{Re}(2x-10)$, $g(x,y) = \mathrm{Re}(2y-10)^3$

(i) $\underline{\mu}_{\max} < 0.1$

(ii) $\left(\underline{\mu}_{\min}, \underline{\mu}_{\max}\right) \subset [0.1, 1.2]$

(iii) $\underline{\mu}_{\min} \geq 1$ LMSOR outperforms all methods

Numerical Results

Comparison of local iterative methods for h=1/81, * indicates no convergence after $5 \cdot 10^4$ iterations.

#	Method	Re = 1	Re = 10	Re = 10^2	Re = 10^3	Re = 10^4
1	(R , I)	(0,6400)	(0,6400)	(0,6400)	(0,6400)	(0,6400)
	$\underline{\mu}_{\min}$ $\underline{\mu}_{\max}$	0.058 0.118	0.613 1.2	6.13 12	61.3 120	613 1200
	LSOR Nat	167	321	2566	25394	*
	LSOR R/B	89	264	2501	24289	*
	LMSOR	89	278	452	519	733

Numerical Results

#	Method	Re = 1	Re = 10	Re = 10^2	Re = 10^3	Re = 10^4
2	(R , I)	(6400,0)	(6400,0)	(0,6400)	(0,6400)	(0,6400)
	$\underline{\mu}_{\min}$ $\underline{\mu}_{\max}$	0.019 0.019	0.015 0.016	0.093 0.119	0.958 1.2	9.58 12
	LSOR Nat	183	161	170	329	3074
	LSOR R/B	132	82	91	307	3145
	LMSOR	132	82	98	317	459

Local Modified EGS

The scheme for the Local Modified EGS is:

$$u_{ij}^{(n+1)} = (1-\tau_{ij})u_{ij}^{(n)} + \tau_{ij}(\ell_{ij}u_{i-1,j}^{(n)} + r_{ij}u_{i+1,j}^{(n)} + t_{ij}u_{i,j+1}^{(n)} + b_{ij}u_{i,j-1}^{(n)})$$

red points

$$u_{ij}^{(n+1)} = (1-\tau'_{ij})u_{ij}^{(n)} + (\ell_{ij}u_{i-1,j}^{(n+1)} + r_{ij}u_{i+1,j}^{(n+1)} + t_{ij}u_{i,j+1}^{(n+1)} + b_{ij}u_{i,j-1}^{(n+1)}) + (\tau'_{ij}-1)(\ell_{ij}u_{i-1,j}^{(n)} + r_{ij}u_{i+1,j}^{(n)} + t_{ij}u_{i,j+1}^{(n)} + b_{ij}u_{i,j-1}^{(n)})$$

black points

LMEGS – Optimum values

Case		Optimum τ_{ij}	Optimum τ'_{ij}	$S\left(\Im_{\tau_{ij},\tau'_{ij}}\right)$
R1	$\underline{\mu} < \overline{\mu} < 1$	$\frac{2}{2-\underline{\mu}^2-\overline{\mu}^2}$	$\frac{2(1-\overline{\mu}^2)}{2-\underline{\mu}^2-\overline{\mu}^2}$	$\frac{\overline{\mu}^2-\underline{\mu}^2}{2-\underline{\mu}^2-\overline{\mu}^2}$
R2	$\overline{\mu} > \underline{\mu} > 1$	$\frac{2}{2-\underline{\mu}^2-\overline{\mu}^2}$	$\frac{2(1-\underline{\mu}^2)}{2-\underline{\mu}^2-\overline{\mu}^2}$	$\frac{\overline{\mu}^2-\underline{\mu}^2}{\underline{\mu}^2+\overline{\mu}^2-2}$

LMEGS – Optimum values

Case		Optimum τ_{ij}	Optimum τ'_{ij}	$S\left(\Im_{\tau_{ij},\tau'_{ij}}\right)$
I1	$\underline{\mu} \neq \overline{\mu}$	$\dfrac{2}{2+\underline{\mu}^2+\overline{\mu}^2}$	$\dfrac{2(\overline{\mu}^2+1)}{2+\underline{\mu}^2+\overline{\mu}^2}$	$\dfrac{\overline{\mu}^2-\underline{\mu}^2}{2+\underline{\mu}^2+\overline{\mu}^2}$

Semi-Iterative LMEGS

Let $\sigma_{ij} = S\left(\Im_{\tau_{ij},\tau'_{ij}}\right)$ and the sequence

$$\delta_{ij}^{(1)} = 1 \qquad \delta_{ij}^{(2)} := \left(1 - 0.5\sigma_{ij}^{2}\right)^{-1} \qquad \delta_{ij}^{(k+1)} := \left(1 - 0.25\sigma_{ij}^{2}\delta_{ij}^{(k)}\right)^{-1}$$

Improvement: $u_{ij}^{(k+1)} := \delta_{ij}^{(k)} u_{ij}^{(k+1)} + (1 - \delta_{ij}^{(k)}) u_{ij}^{(k-1)}$

Numerical Results

Comparison of LMSOR and SI-LMEGS:

$$f(x,y) = g(x,y) = \mathrm{Re}\cdot x^2$$

N = 40	Re = 1	Re = 10	Re = 10^2	Re = 10^3	Re = 10^4
(R , I)	R	R	I	I	I
$\underline{\mu}_{min}$	0.0383	0.0380	0.0045	0.0116	0.0226
$\overline{\mu}_{max}$	0.9971	0.9971	0.9971	11.5304	115.73
SI-LMEGS	87	120	79	88	1058
LMSOR	97	125	83	76	932

Numerical Results

N = 120	Re = 1	Re = 10	Re = 10^2	Re = 10^3	Re = 10^4
(R , I)	R	R	R	I	I
$\underline{\mu}_{min}$	0.0130	0.0130	0.0119	0.0023	0.0025
$\overline{\mu}_{max}$	0.9997	0.9997	0.9997	3.9379	40.6161
SI-LMEGS	255	353	226	163	295
LMSOR	284	369	240	161	250

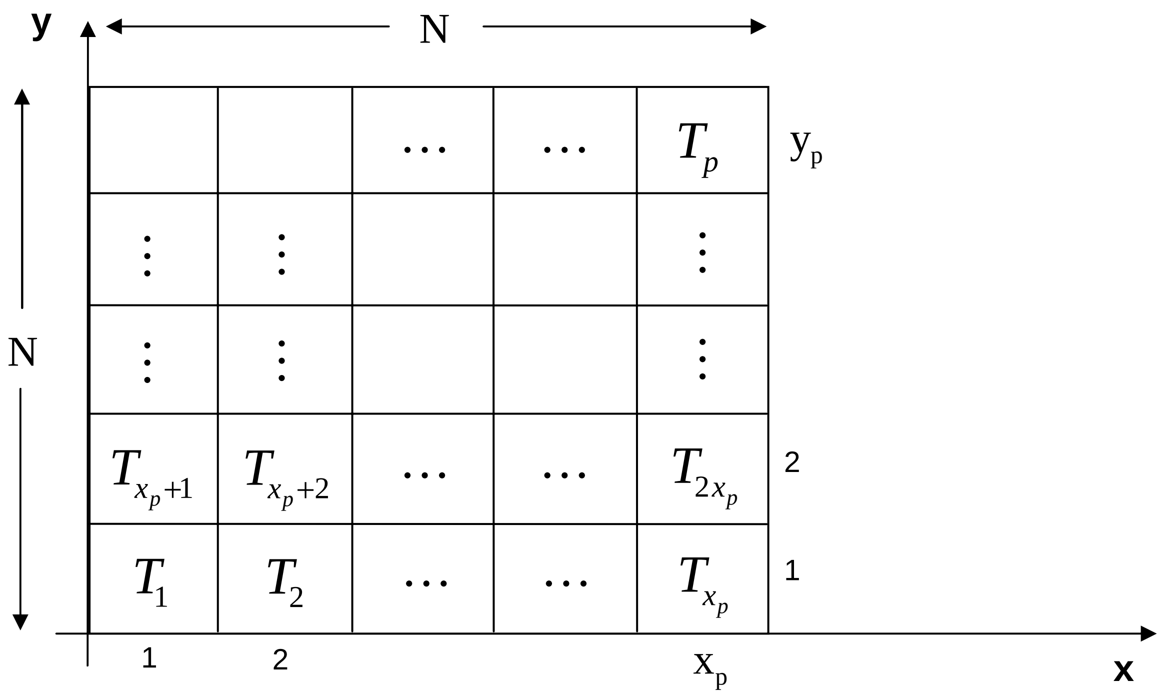
Numerical Results

Conclusions:

- If $\bar{\mu}_{\max} < 3$, SI-LMEGS is better than LMSOR

Parallel Implementation

We divide the N^2 –points mesh into p equal rectangles of size $N/x_p \times N/y_p$.



Parallel Implementation

Two phases in each step:

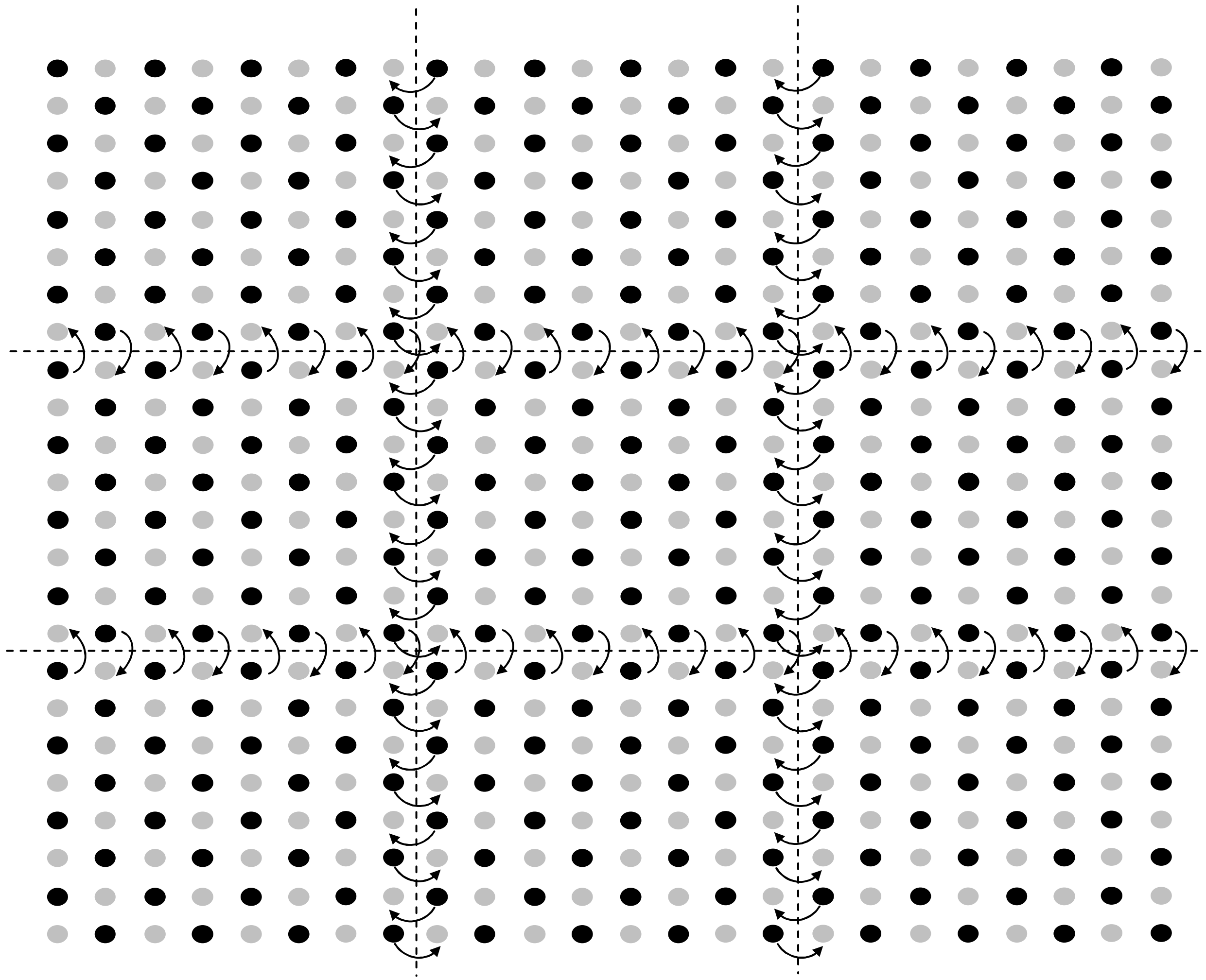
Phase 1: Communication of black boundary points.

Phase 2: Communication of red boundary points.

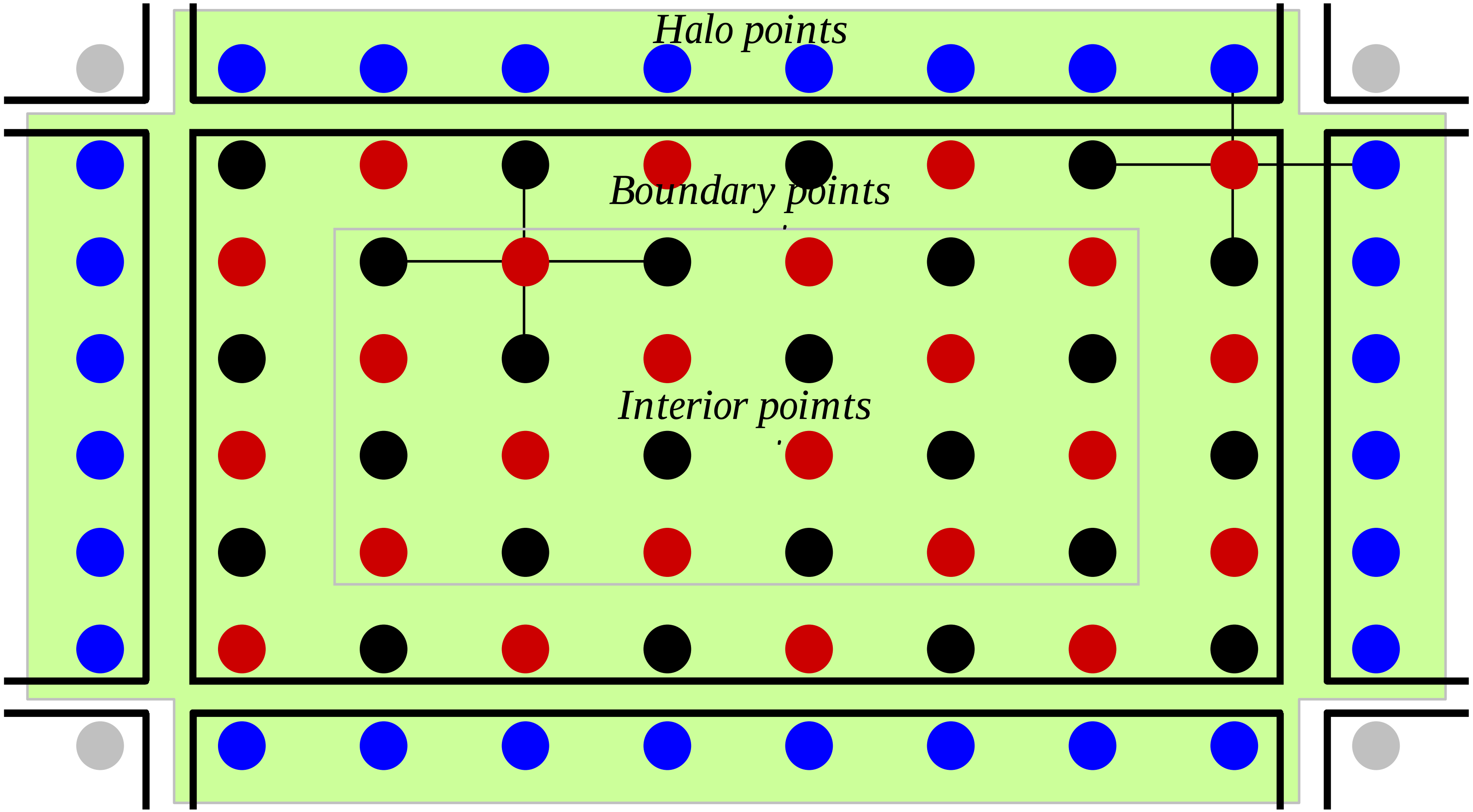
Computation and communication time for every process:

$$t_{comp}(T_i) = O(N^2/p) \qquad t_{comm}(T_i) = 2\left(\frac{N}{x_p} + \frac{N}{y_p}\right)$$

Parallel Implementation



Communication



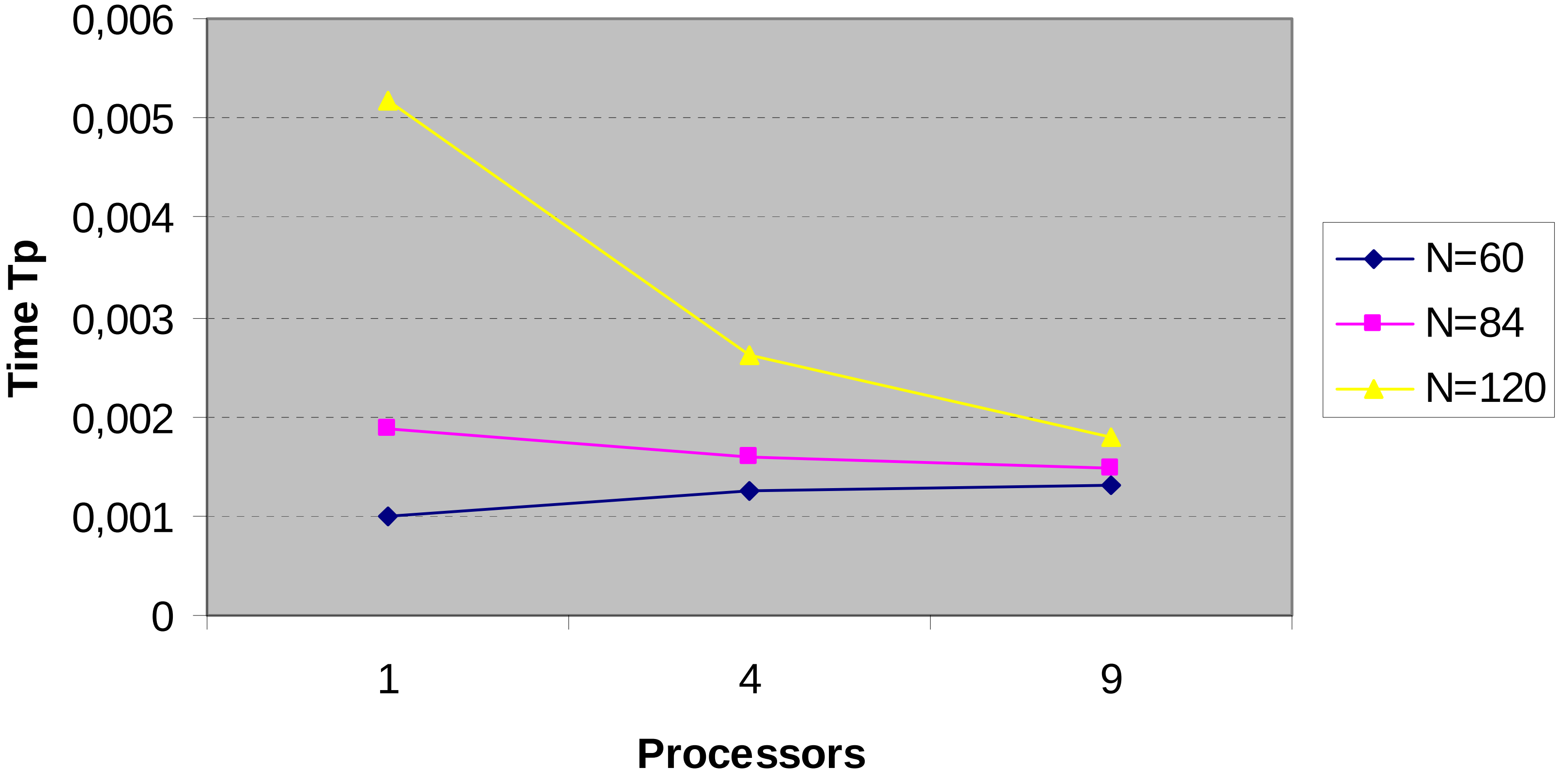
Parallel Implementation

Speed up: $$S_p = \frac{T_1}{T_p} = O\left(\frac{p}{1 + \left(y_p + x_p \right) / N} \right)$$

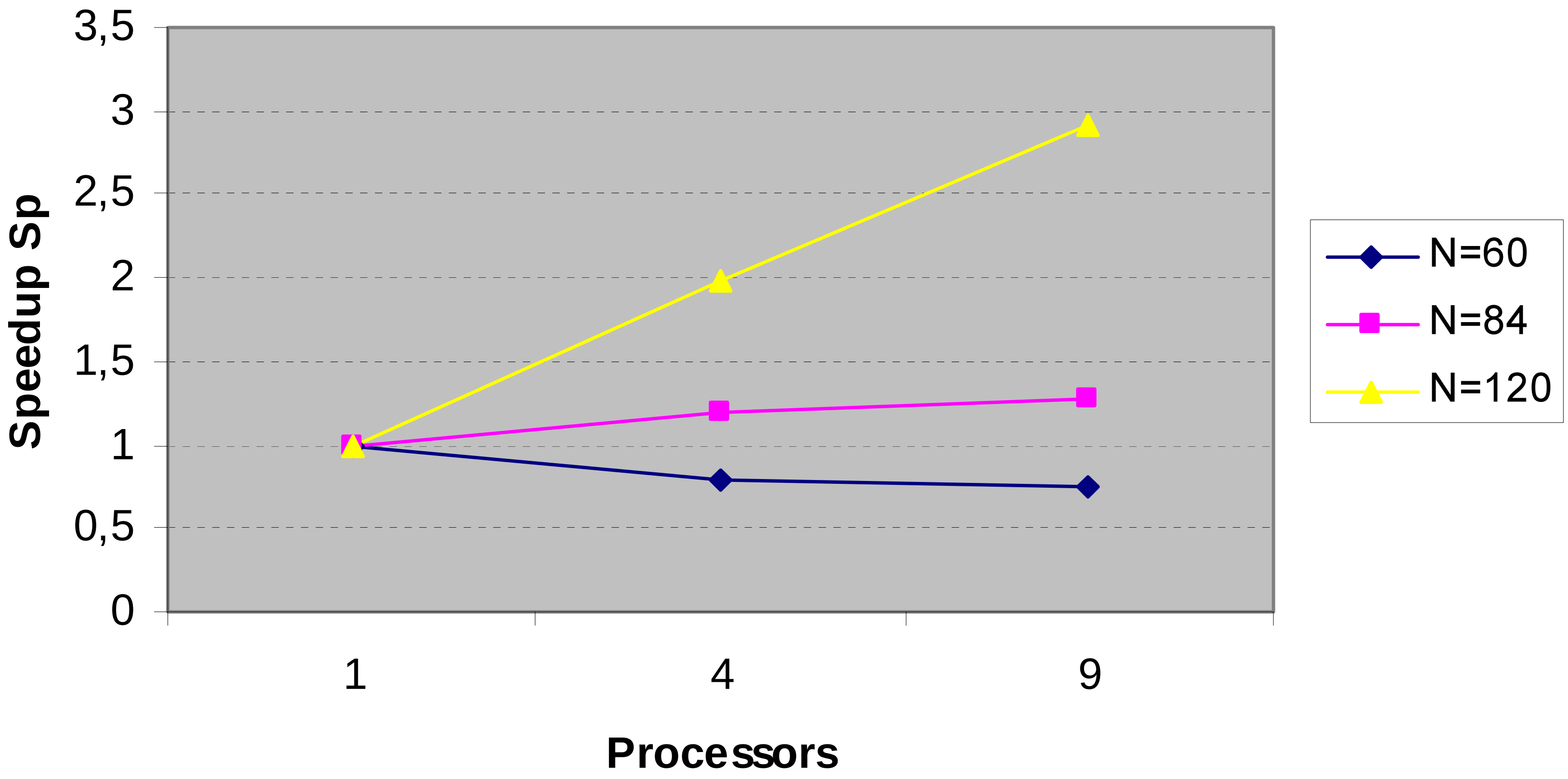
Efficiency: $$E_p = \frac{S_p}{p} = O\left(\frac{1}{1 + \left(y_p + x_p \right) / N} \right)$$

$S_p \to O(p)$ and $E_p \to 1$ while $\dfrac{p}{N} \to 0.$

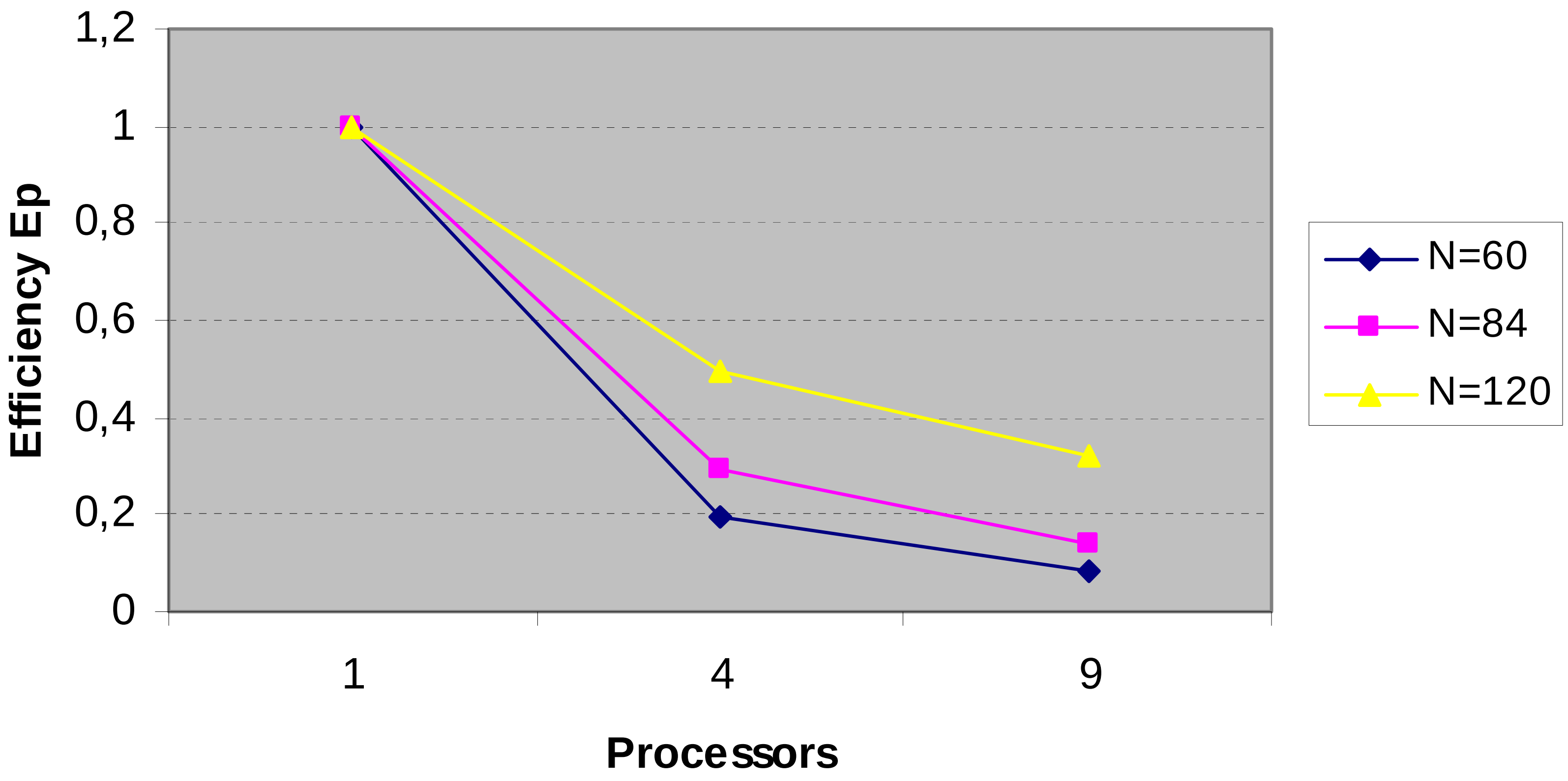
Parallel Implementation



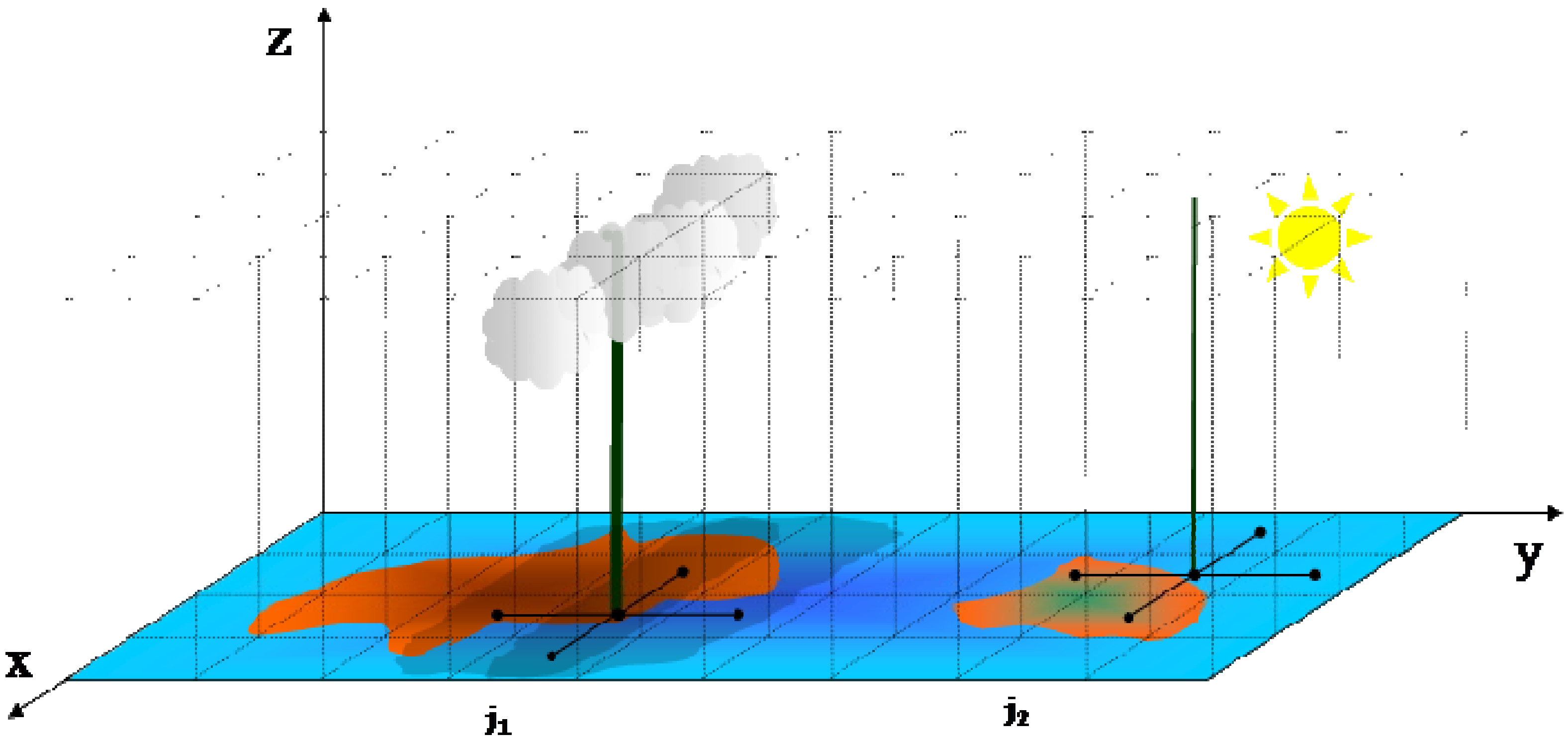
Parallel Implementation



Parallel Implementation



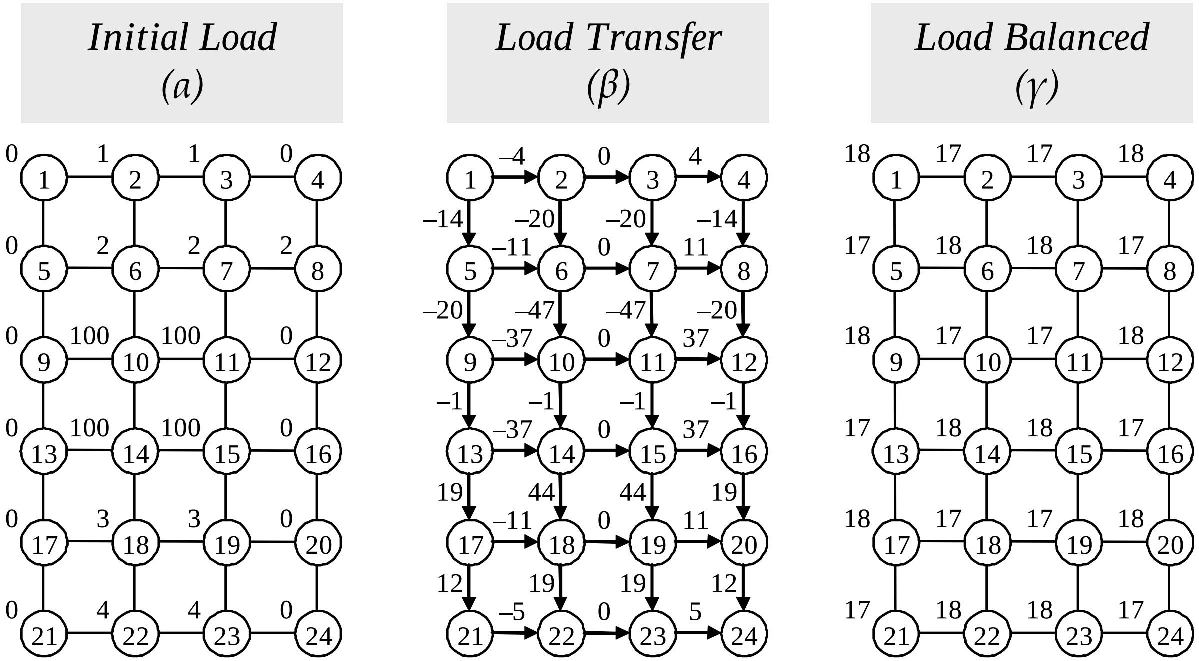
The Load Balancing Problem



The Diffusion Method

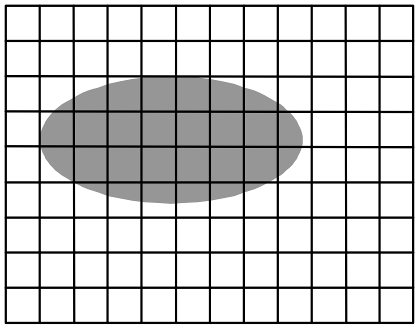
$$u_i^{(n+1)} = u_i^{(n)} + \sum_{j \in A(i)} \tau_{ij} \left(u_j^{(n)} - u_i^{(n)} \right)$$

[ECMWF Conf 00]

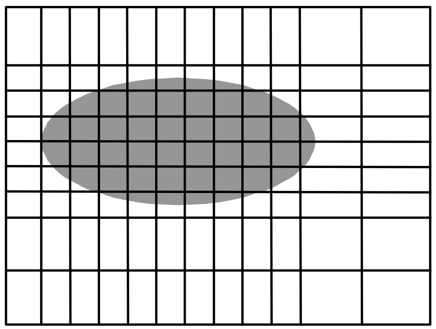


Domain Decomposition

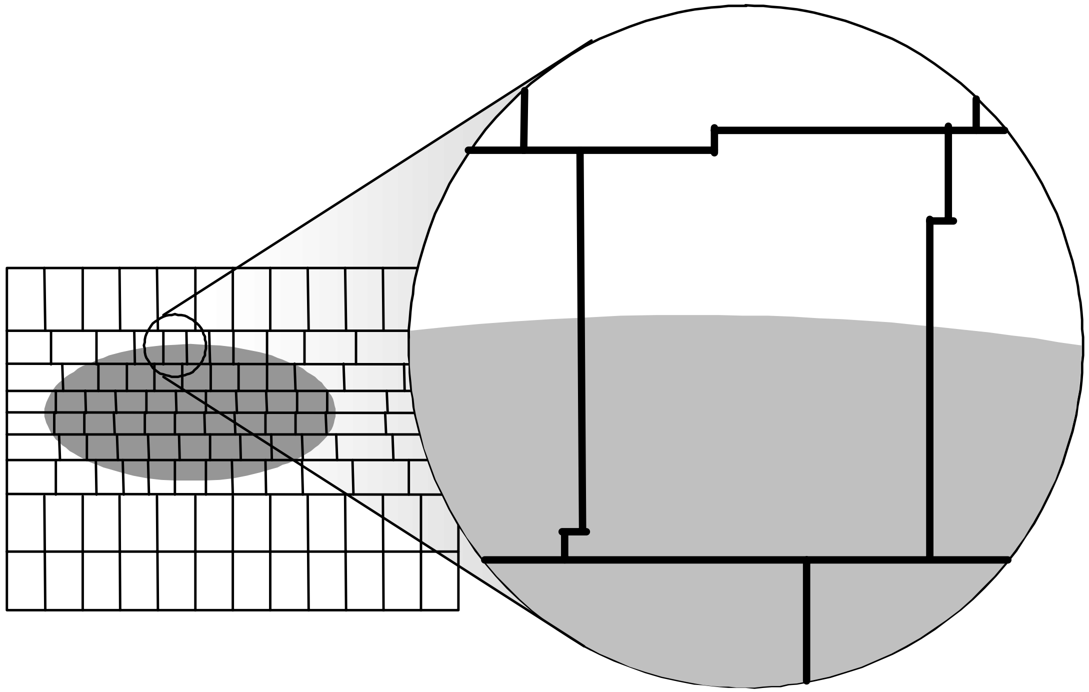
Initial Decomposition



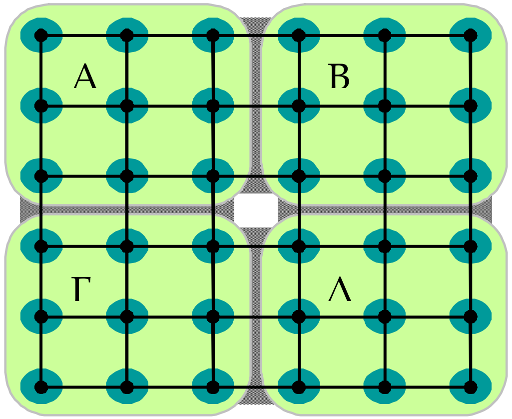
Nesting

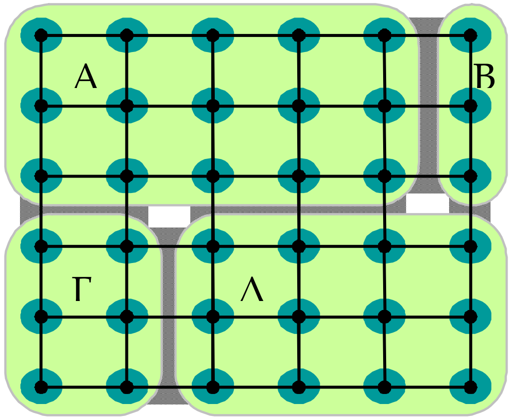


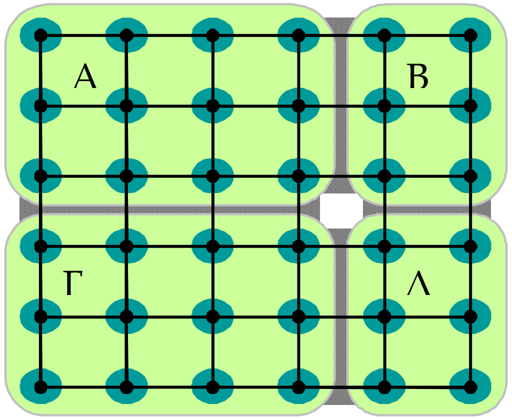
Adjusted Nesting



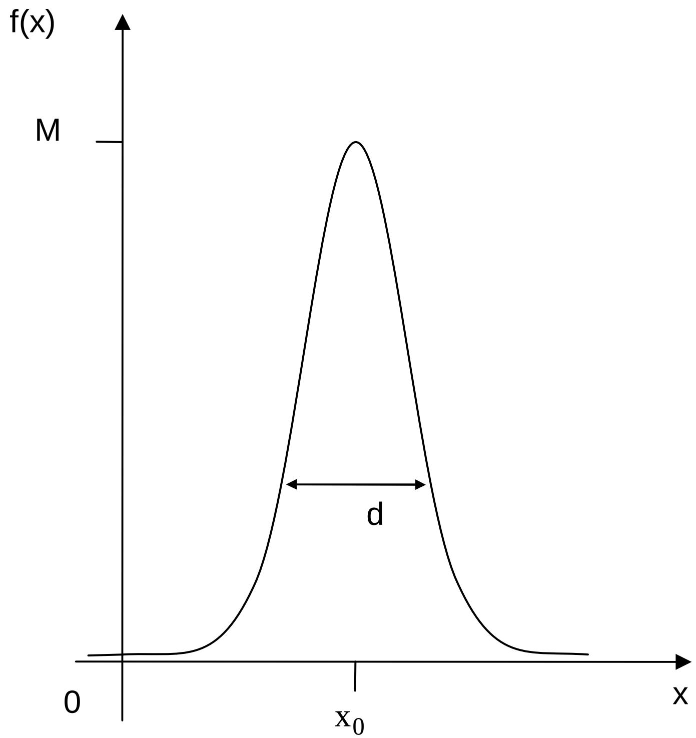
Load Transfer Policy







The Load



$$f(x) = M \cdot e^{-\left(\frac{x-x_0}{d/2}\right)^2}$$

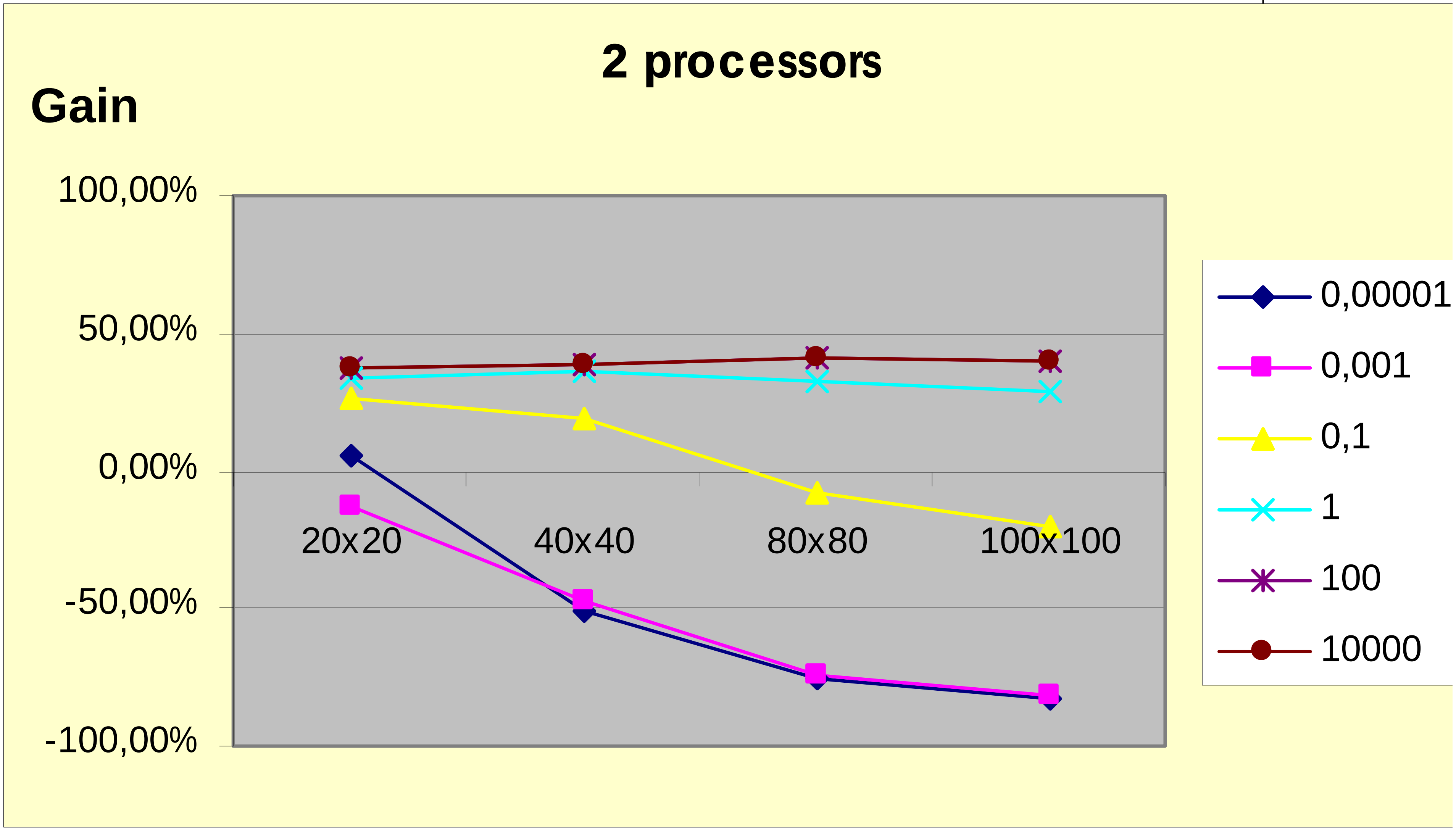
Numerical Results

size	6 processors						
	Without LB			With LB			
	Comm.	*Comp.*	*Time*	*Comm.*	*Comp.*	*Time*	*Gain %*
Scale Factor(M) = 0,00001							
20x20	1,321	14,710	16,031	2,371	14,186	16,557	-3,28%
40x40	1,620	14,729	16,349	15,250	6,541	21,791	-33,29%
80x80	8,819	38,036	46,855	107,029	9,813	116,842	-149,37%
100x100	13,097	65,018	78,115	197,644	13,188	210,832	-169,9 %
Scale Factor(M) = 0,001							
20x20	1,332	17,256	18,588	2,371	15,277	17,648	5,06%
40x40	1,623	17,896	19,519	15,267	5,832	21,099	-8,09%
80x80	8,745	38,927	47,672	107,098	10,237	117,335	-146,13%
100x100	12,586	66,280	78,866	197,027	13,420	210,447	-166,84%
Scale Factor(M) = 0, 1							
20x20	1,298	21,451	22,749	2,369	14,331	16,7	26,59%
40x40	1,601	48,174	49,775	15,380	27,548	42,928	13,76%
80x80	8,505	126,738	135,243	107,143	54,423	161,566	-19,46%
100x100	12,804	174,196	187	196,895	70,574	267,469	-43,03%

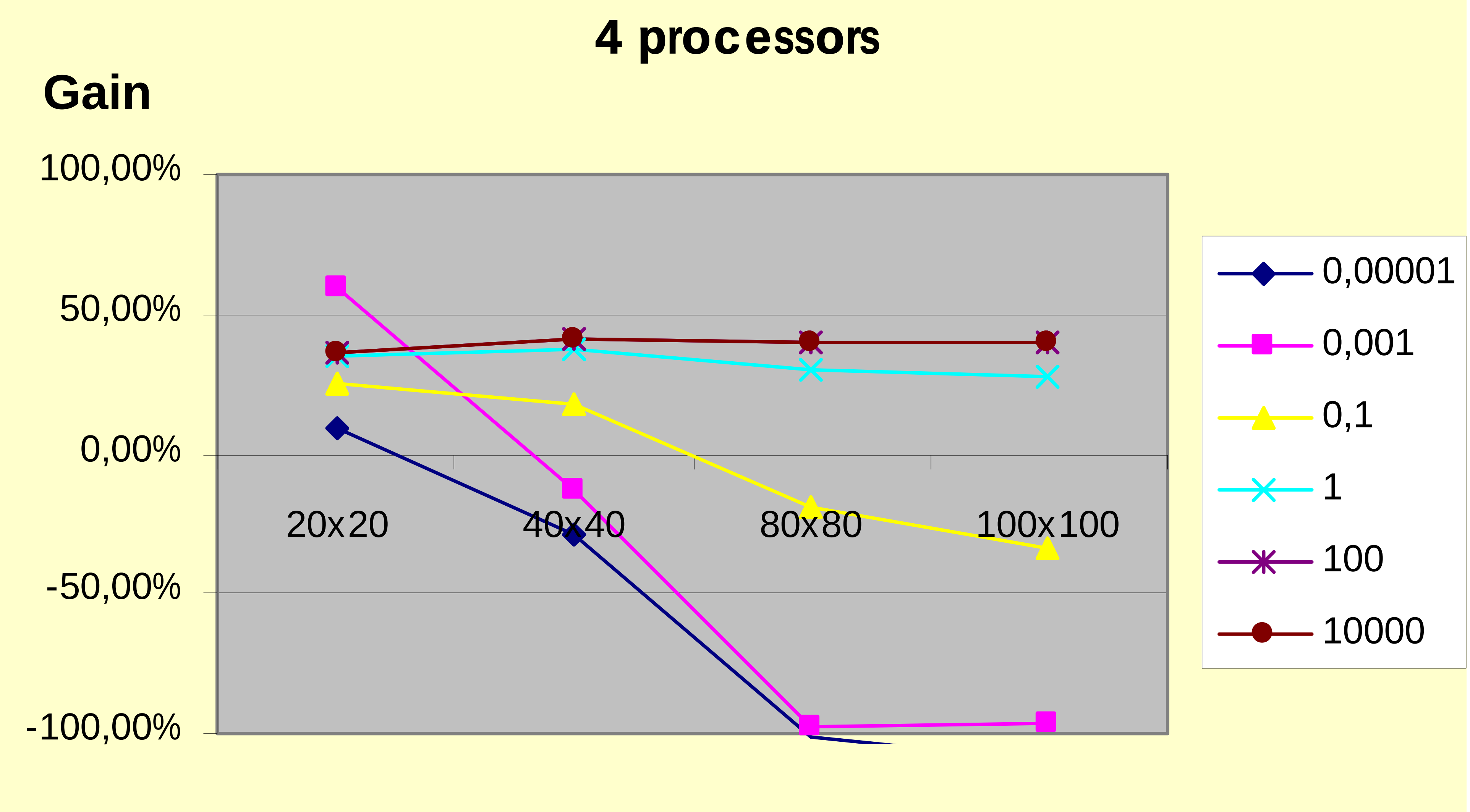
Numerical Results

Size	6 processors						
	Without LB			With LB			
	Comm.	*Comp.*	*Time*	*Comm*	*Comp.*	*Time*	*Gain%*
Scale Factor(M) = 1							
20x20	1,292	196,913	198,205	2,390	123,584	125,974	36,44%
40x40	1,616	405,537	407,153	15,382	235,292	250,674	38,43%
80x80	8,727	923,553	932,28	107,148	455,723	562,871	39,62%
100x100	12,615	1156,905	1169,52	197,564	590,688	788,252	32,6 %
Scale Factor(M) = 100							
20x20	1,295	19497,157	19498,452	2,371	12141,368	12143,739	37,72%
40x40	1,590	39715,174	39716,764	15,303	23087,958	23103,261	41,83%
80x80	8,802	88575,737	88584,539	107,160	44607,763	44714,923	49,52%
100x100	13,183	109198,314	109211,497	197,090	57804,577	58001,667	46,89%
Scale Factor(M) = 10000							
20x20	1,303	1949521,885	1949523,188	2,367	1213920,807	1213923,174	37,73%
40x40	1,641	3970679,057	3970680,698	15,344	2308354,573	2308369,917	41,86%
80x80	8,620	8853812,312	8853820,932	107,554	4459812,674	4459920,228	49,63%
100x100	12,834	10913401,265	10913414,099	197,285	5779198,351	5779395,636	47,04%

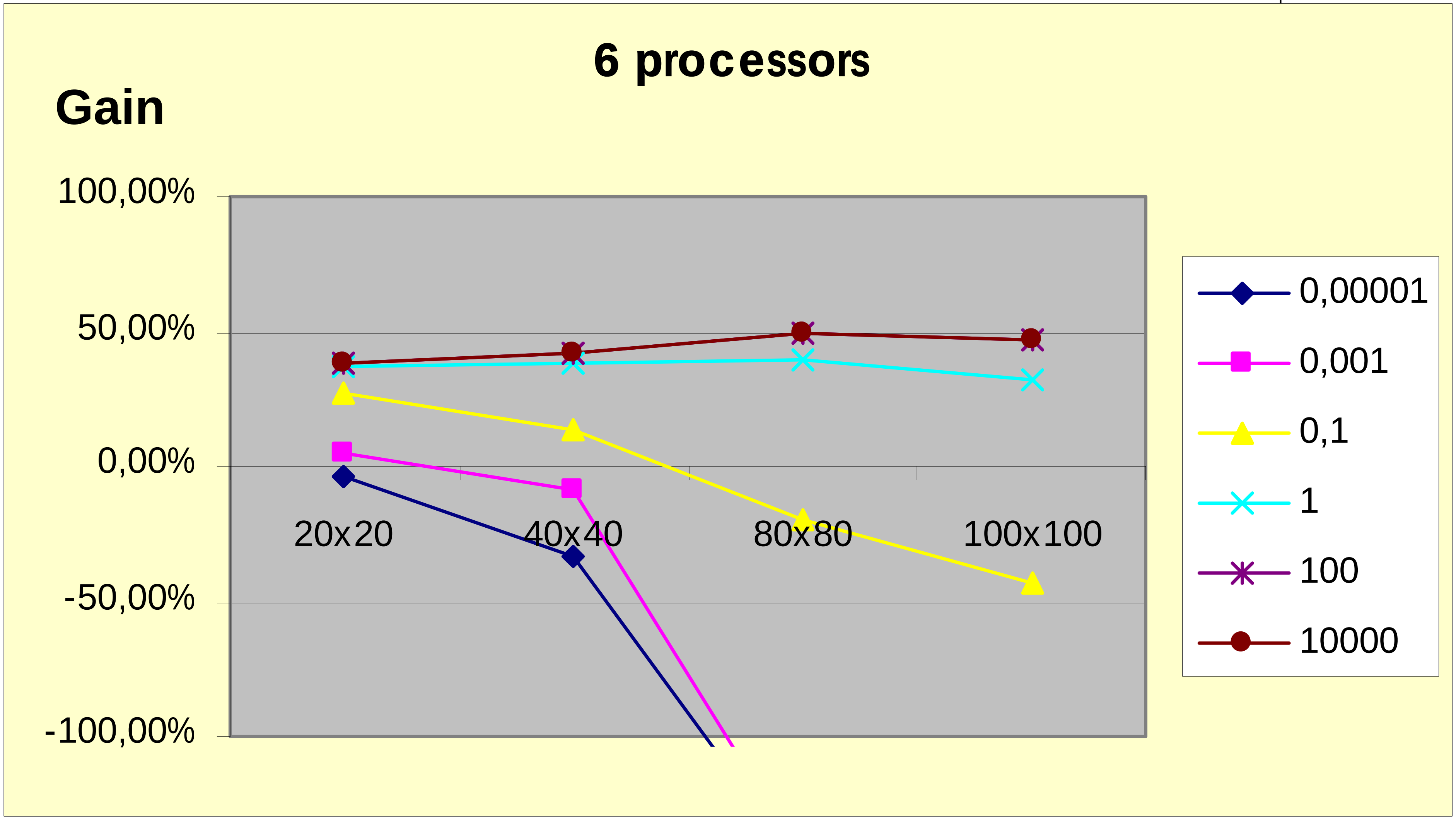
Numerical Results



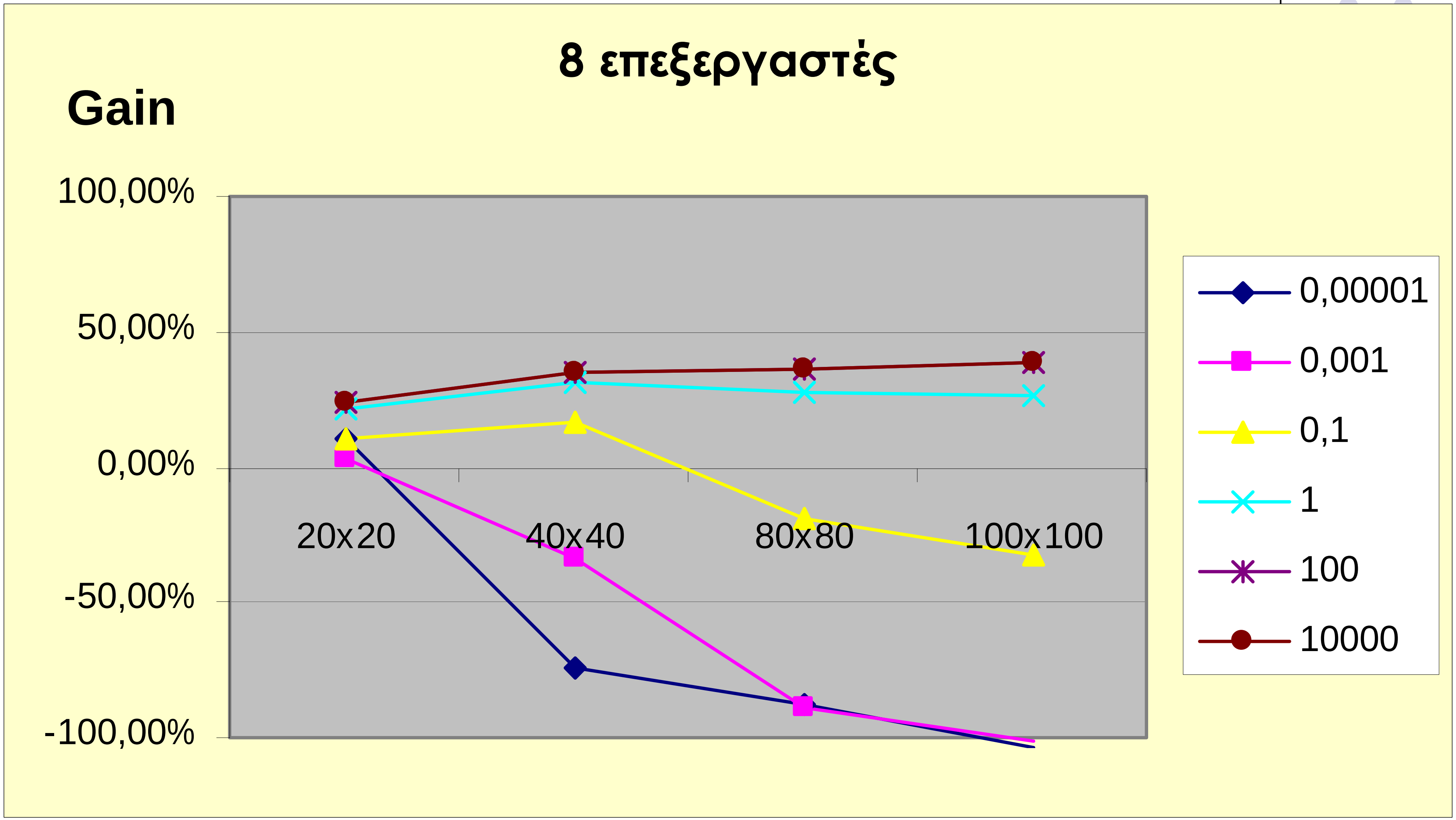
Numerical Results



Numerical Results



Numerical Results



CONCLUSIONS

- Local MSOR is easy to parallelize
- Efficient

$$S_p \rightarrow O(p) \quad \text{and} \quad E_p \rightarrow 1 \qquad \text{while} \quad \frac{p}{N} \rightarrow 0.$$

- It does worth to LB !

Future Work

- Determination of optimum values for the Local MESOR
- Dynamic LB for Solving the Adjusted Nesting Problem
- Heterogeneous, Asynchronous Load Balancing

Thank you for your attention

Any Questions ?