PaStiX

Mathieu Faverge Xavier Lacoste Pierre Ramet

13 février 2009

Plan

PaStiX

- Projet développé au sein de l'équipe ScAlApplix- *INRIA Bordeaux - Sud-Ouest*
  `http://www.labri.fr/projet/scalapplix/`
- Solveur linéaire pour de grand systèmes creux
- Factorisation directe et ILU(k)
- Utilise une méthode supernodale

PaStiX Users

PaStiX team

- M. Faverge (PhD student ANR NUMASIS)
- P. Henon (researcher INRIA)
- X. Lacoste (engineer INRIA)
- F. Pellegrini (assistant professor LaBRI/INRIA)
- P. Ramet (assistant professor LaBRI/INRIA)
- J. Roman (professor, leader of ScAlApplix INRIA project)

Main users

- Electomagnetism and structural mechanics codes at CEA-DAM CESTA
- MHD Plasma instabilities for ITER at CEA-Cadarache
- Fluid mechanics at IMB Bordeaux

PaStiX Functions (1/3)

- LLt, LDLt, LU factorization (symmetric pattern) with supernodal implementation
- Static pivoting (Max. Weight Matching)
  + Refinement : Iteratif / CG / GMRES
- 1D/2D block distribution + Full BLAS3
- Simple/Double precision + Float/Complexe operations

PaStiX Functions (2/3)

- MPI/Threads implementation
  (SMP node / Cluster / Multi-core / NUMA)
- Take care of NUMA effects
- Sequential or distributed interface
- Dynamic scheduling inside SMP nodes (static mapping)
- Support external ordering library ((PT-)Scotch/ MeTiS)

PaStiX Functions (3/3)

- Require only C + MPI + Posix Thread
- Multiple RHS (direct factorization)
- Incomplete factorization ILU(k) preconditionner
- Out-of Core implementation (in SMP mode only)

PaStiX Current Works

Current Works

- Parallel symbolic factorization
- Dynamic scheduling inside NUMA nodes (static mapping)
- Out-of Core implementation (for MPI and MPI/thread versions)
- Generic Finite Element Assembly (domaine decomposition associated to matrix distribution)

ANR

- NUMASIS (CIS05) : `http://numasis.gforge.inria.fr/`
- SOLSTICE (CIS06) : `http://solstice.gforge.inria.fr/`
- ASTER (CIS06) : `http://aster.gforge.inria.fr/`

PaStiX Release

- Latest version : "revolution" (24/10/2008 - svn 1789)
- Next version : "???" (Before the end of February)

Highlights

The direct solver PaStiX has been successfully used by CEA/CESTA to solve a huge symmetric complex sparse linear system arising from a 3D electromagnetism code on the TERA-10 CEA supercomputer.

- **45 millions unknowns** : required 1.4 Petaflops and was completed in half an hour on 2048 processors.
- **83 millions unknowns** : required 5 Petaflops and was completed in 5 hours on 768 processors.

To our knowledge a system of this size and this kind has never been solved by a direct solver.

PaStiX Changes for the next release

- Many bug fixes
- New interface : MURGE (common with HIPS)
- Suppress the Scotch dependency
- Allow to use more easily its own ordering
- Allow to switch the ordering method at runtime (IPARM_ORDERING)
- Make communication progress with some MPI implementation (David Goudin)
- Suppress the asynchronous send in solve step (due to precedent item)
- Correct the memory leak when refinement step is not called
- Add inertia in return parameters (IPARM_INERTIA)
- Add the distributed filling of internal matrix structure
- Add (PT-)Scotch/HIPS compatibility for integer size (`INTSIZE32` and `INTSIZE64`)

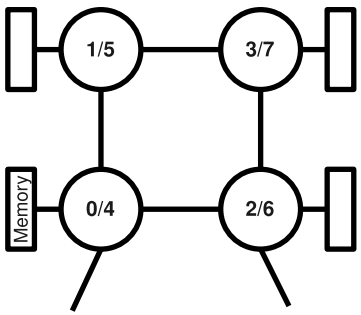
Plan

Bibliothèques nécessaires

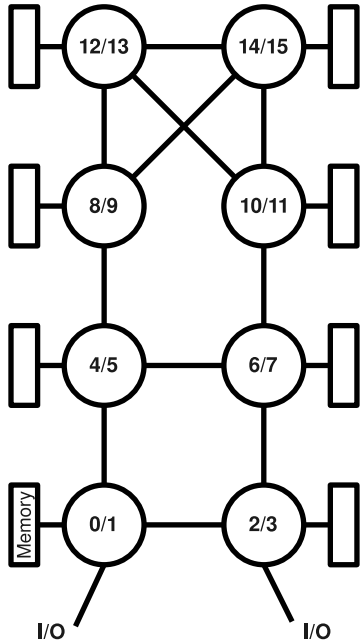
- Une bibliothèque de BLAS :
  GotoBLAS, ACML, MKL, ...
- Une bibliothèque de communication MPI :
  Mpich2, OpenMPI, MAD-MPI ...
- Une bibliothèque de threads :
  Thread POSIX ou MARCEL
- Une bibliothèque pour la renumérotation des inconnues :
  (PT-)SCOTCH et/ou METIS

Plan

Experimental Platforms (1/2)



(a) NUMA8



(b) NUMA16

- 4 and 8 Dual-Core AMD Opteron(tm) processors
- HyperTransport interconnection
- 4GB per core

FIG.: Architectures used for benchmarks

Experimental Platforms (2/2)

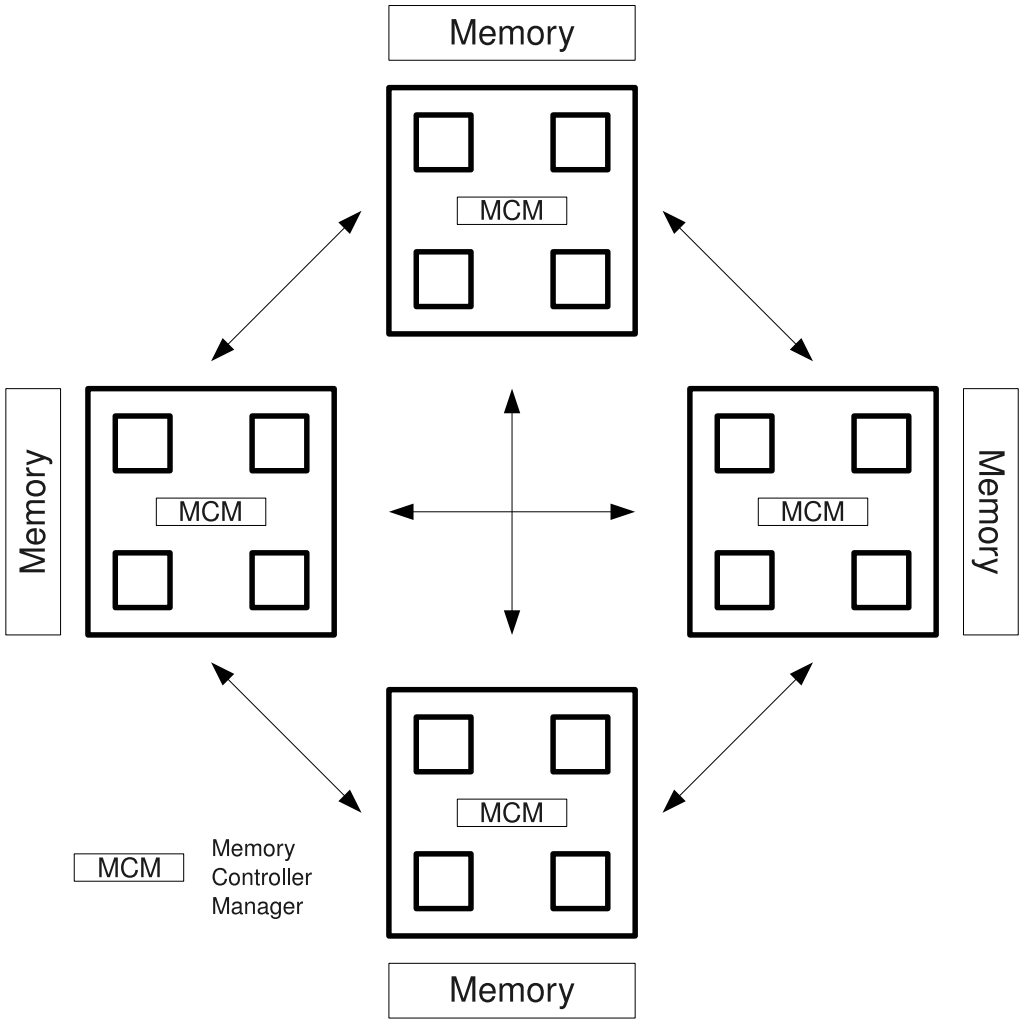


FIG.: SMP16

- 4 blades of 4 Power5
- 28GB per node

Matrices used for experiments

Name	Columns	NNZ_A	NNZ_L	OPC
MATR5	485 597	24 233 141	1 361 345 320	9.84422e+12
AUDI	943 695	39 297 771	1 144 414 764	5.25815e+12
NICE20	715 923	28 066 527	1 050 576 453	5.19123e+12
INLINE	503 712	18 660 027	158 830 261	1.41273e+11
NICE25	140 662	2 914 634	51 133 109	5.26701e+10
MCHLNF	49 800	4 136 484	45 708 190	4.79105e+10
THREAD	29 736	2 249 892	25 370 568	4.45729e+10
HALTERE	1 288 825	10 476 775	405 822 545	7.62074e+11

NNZ_A is the number of off-diagonal terms in the triangular part of the matrix A, NNZ_L is the number of off-diagonal terms in the factorized matrix L and OPC is the number of operations required for the factorization.

AUDI

Size : 943695

NZZA : 39297771

Fill : $1154647737 (\times 29,38)$

Coefficients : DOUBLE REAL

Symetric : yes

Factorization time (in seconds) :

MPI tasks number Threads number	2	4	8	16	32
2	378.00	193.00	103.00	57.90	37.90
4	195.00	100.00	56.00	34.40	—
8	102.00	56.70	31.8	22.00	—
16	60.10	33.30	24.50	—	—

AUDI

Solve time (in seconds) :

MPI tasks number Threads number	2	4	8	16	32
2	4.50	2.50	1.77	0.84	0.63
4	2.54	1.79	0.81	0.62	—
8	1.77	0.83	0.61	0.32	—
16	0.85	0.59	0.31	—	—

Memory per MPI process (in Go) :

MPI tasks number Threads number	2	4	8	16	32
2	7.16	4.71	2.97	2.27	1.78
4	7.26	4.95	3.20	2.06	—
8	7.54	5.33	3.30	2.14	—
16	7.98	5.47	—	—	—

MHD

Size : 485597

NZZA : 24233141

Fill : $1270258848 (\times 52, 41)$

Coefficients : DOUBLE REAL

Symetric : no

Factorization time (in seconds) :

MPI tasks number Threads number	2	4	8	16
2	597.00	329.00	MEM	MEM
4	299.00	162.00	113.00	66.60
8	160.00	98.40	58.40	51.60
16	114.00	59.70	49.00	—

MHD

Solve time (in seconds) :

MPI tasks number / Threads number	2	4	8	16
2	1.71	0.94	MEM	MEM
4	0.99	0.72	0.38	0.24
8	0.73	0.38	0.21	0.18
16	0.37	0.21	0.14	—

Memory per MPI process (in Go) :

MPI tasks number / Threads number	2	4	8	16
2	6.12	4.45	MEM	MEM
4	6.62	4.74	3.72	3.27
8	6.94	5.15	4.02	2.89
16	7.77	5.50	4.15	—

HALTERE

Size : 1288825

NZZA : 10476775

Fill : 404977313(×38.65)

Coefficients : DOUBLE COMPLEXE

Symetric : yes

Factorization time (in seconds) :

MPI tasks number Threads number	2	4	8	16
2	180.00	88.50	48.90	29.80
4	88.00	48.90	27.60	17.90
8	48.50	29.30	20.20	17.00
16	33.20	24.70	17.40	-

HALTERE

Solve time (in seconds) :

MPI tasks number Threads number	2	4	8	16
2	3.06	1.77	1.02	0.54
4	1.77	1.03	0.54	0.35
8	1.02	0.54	0.29	0.15
16	0.54	0.28	0.16	-

Memory per MPI process (in Go) :

MPI tasks number Threads number	2	4	8	16
2	5.08 Go	2.77 Go	1.67 Go	1.14 Go
4	5.04 Go	2.78 Go	1.70 Go	1.13 Go
8	5.07 Go	2.87 Go	1.80 Go	1.31 Go
16	5.12 Go	2.94 Go	2.12 Go	-

HALTERE : incomplete factorization with 16 threads/MPI

level 1 : fill=3.51 iter=250 (6e-7)

level 3 : fill=7.39 iter=12 (1e-7)

level 5 : fill=10.10 iter=5 (1e-7)

Factorization and Solve time (in seconds) :

MPI tasks number level of fill	1	2		1	2
1	10.30	9.73		1.05	0.61
3	24.50	13.90		1.04	0.62
5	34.30	20.30		0.96	0.53

AMANDE

Size : 6994683

NZZA : 58477383

Fill : $3150095338 (\times 53,87)$

Coefficients : DOUBLE COMPLEXE

Symetric : yes

MPI tasks number Threads number	4	8
8	—	257.00
16	262.00	—

(need 12.7Go per MPI process on 8x8 processors)

NICE20

Size : 715923

NZZA : 28066527

Fill : $1067798610 (\times 38,04)$

Coefficients : DOUBLE REAL

Symetric : yes

Factorization time (in seconds) :

MPI tasks number Threads number	2	4	8	16
2	360.00	179.00	102.00	58.10
4	181.00	97.60	54.20	34.90
8	104.00	53.20	33.00	22.90
16	60.30	36.00	—	—

NICE20

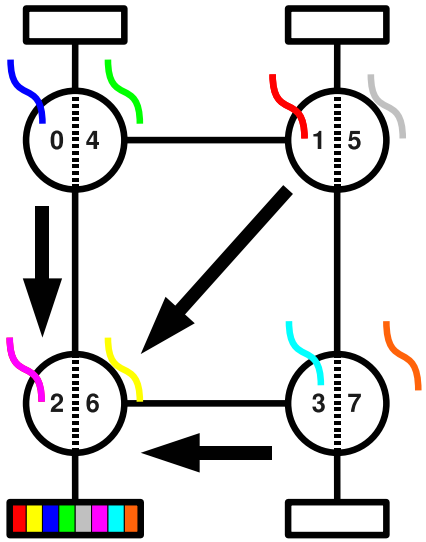
Solve time (in seconds) :

MPI tasks number Threads number	2	4	8	16
2	2.93	1.60	0.98	0.66
4	1.59	0.97	0.66	0.36
8	0.97	0.63	0.35	0.26
16	0.67	0.34	—	—

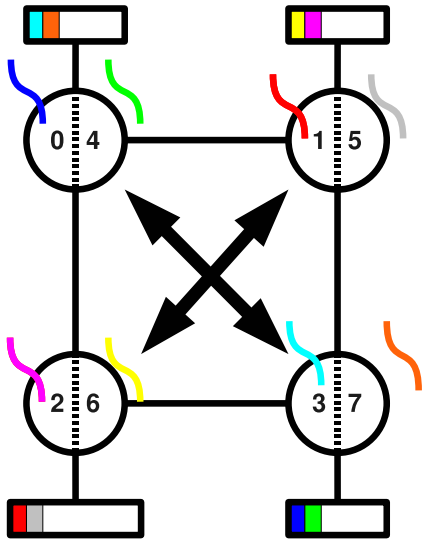
Memory per MPI process (in Go) :

MPI tasks number Threads number	2	4	8	16
2	5.36	3.28	2.40	1.69
4	5.51	3.50	2.35	1.75
8	5.76	3.62	2.47	1.72
16	6.12	4.01	—	—

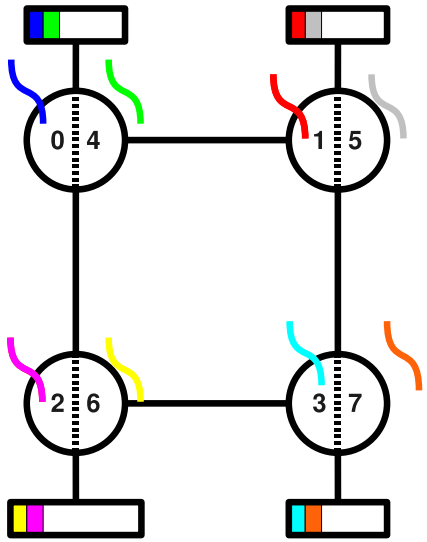
Memory Location



(a) On one core



(b) Worst Location



(c) Good Location

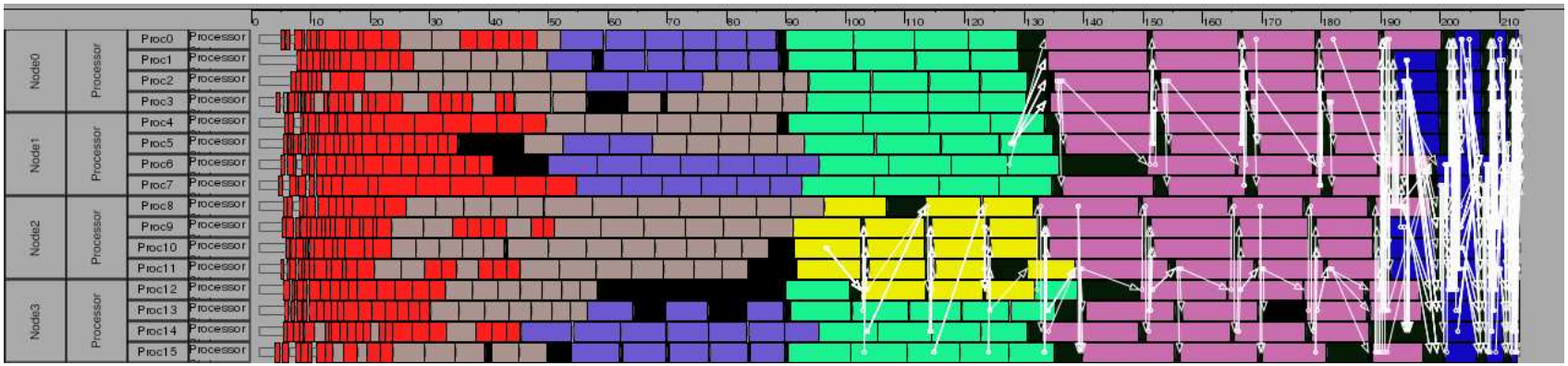
Memory location : Results

Matrix	NUMA8		NUMA16		SMP16	
	Global	Local	Global	Local	Global	Local
MATR5	437	410	527	341	162	161
AUDI	256	217	243	185	101	100
NICE20	227	204	204	168	91.40	91
INLINE	9.70	7.31	20.90	15.80	5.80	5.63
NICE25	3.28	2.62	6.28	4.99	2.07	1.97
MCHLNF	3.13	2.41	5.31	3.27	1.96	1.88
THREAD	2.48	2.16	4.38	2.17	1.18	1.15
HALTERE	134	136	103	93	48.40	47.90

TAB.: Influence of NUMA-aware allocation on numerical factorization of PASTIX solver (in seconds)

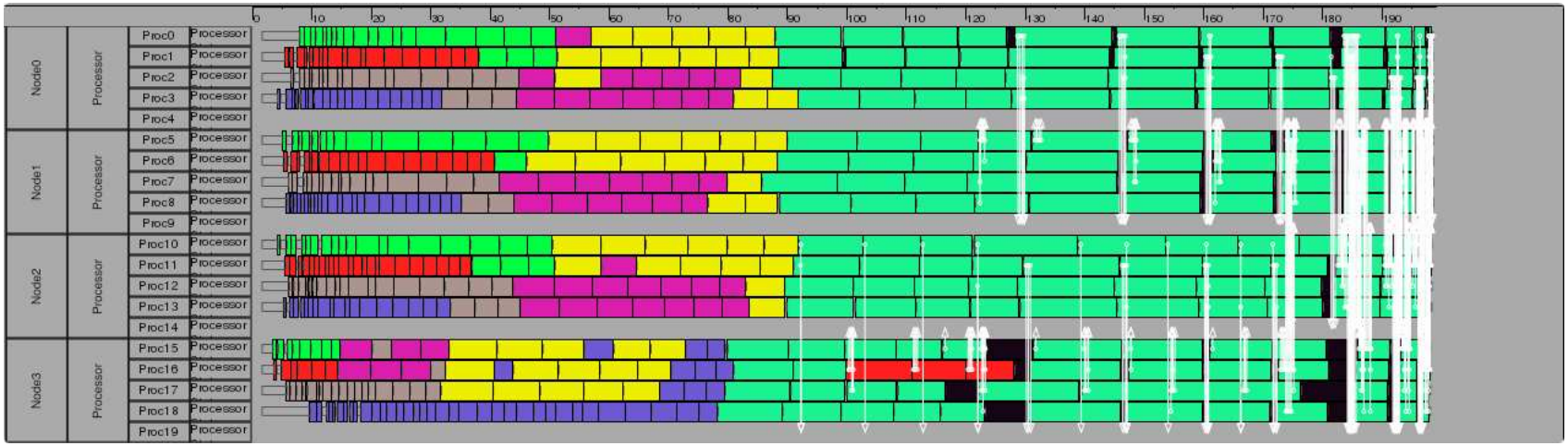
- *Global* : Initial allocation
- *Local* : New NUMA-aware allocation

Static Scheduling Gantt Diagram



- Each color gives the number of candidate processors for the task (level in the tree).
- *MATR5* test case on NUMA8 with 4 MPI process of 4 threads
- Black blocks are idle times
- White arrows are communications

Dynamic Scheduling Gantt Diagram



- Each color gives the number of candidate processors for the task (level in the tree).
- *MATR5* test case on NUMA8 with 4 MPI process of 4 threads
- Black blocks are idle times
- White arrows are communications

Results on threaded version

Matrix	NUMA8			NUMA16			SMP16		
	V0	V1	V2	V0	V1	V2	V0	V1	V2
MATR5	437	410	**389**	527	341	**321**	162	161	**150**
AUDI	256	217	**210**	243	185	**176**	101	100	**100**
NICE20	227	**204**	227	204	168	**162**	91.40	91	**90.30**
INLINE	9.70	**7.31**	7.32	20.90	15.80	**14.20**	5.80	**5.63**	5.87
NICE25	3.28	**2.62**	2.82	6.28	**4.99**	5.25	2.07	1.97	**1.90**
MCHLNF	3.13	**2.41**	2.42	5.31	3.27	**2.90**	1.96	1.88	**1.75**
THREAD	2.48	2.16	**2.05**	4.38	2.17	**2.03**	1.18	1.15	**1.06**
HALTERE	134	136	**129**	103	**93**	94.80	48.40	47.90	**47.40**

V0 : Static scheduling and no NUMA-aware allocation

V1 : Static scheduling and NUMA-aware allocation

V2 : Dynamic scheduling and NUMA-aware allocation

Results on MPI/threaded version

Nb. Node	NUMA8				SMP16			
	AUDI		MATR5		AUDI		MATR5	
	V0	V1	V0	V1	V0	V1	V0	V1
1	217	**210**	410	**389**	100	**100**	161	**150**
2	142	**111**	212	**200**	60.4	**56.8**	113	**87**
4	69	**57.7**	171	**114**	33.7	**32.6**	59.3	**54.6**
8	45.3	**35.6**	117	**78.8**	-	-	-	-

All with NUMA-aware allocation

V0 : Static scheduling

V1 : Dynamic scheduling

Plan

Infos

- HomePage of the project :
  `https://pastix.gforge.inria.fr/`
- Downloads :
  `http://gforge.inria.fr/frs/?group_id=186`
- Publications related to this project can be found on :
  `http://pastix.gforge.inria.fr/doc/publis/Keyword/SPARSE.html`
- Quick reference card :
  `http://gforge.inria.fr/docman/index.php?group_id=186`
- Contact :
  `(faverge,henon,lacoste,ramet)@labri.fr`