Hardware-oriented numerics and the FEAST project

Dipl.-Inform. Christian Becker

Christian Becker@mathematik.uni-dortmund.de

Institute of Applied Mathematics

University of Dortmund

LS3

Motivation

'Typical': 3D Poisson problem

- $100 \times 100 \times 100$ grid points $\longrightarrow$ problem size $N = 10^6$
- Complexity of GE: $N^{7/3} \approx\ 10^{14}$ FLOP
  100 sec on a 1 TFLOP/s computer
- Complexity of (opt.) multigrid: $1000N \approx\ 10^9$ FLOP
  100 sec on a 10 MFLOP/s computer

- $1000 \times 1000 \times 1000$ grid points $\longrightarrow$ problem size $N = 10^9$
- Complexity of GE: $N^{7/3} \approx\ 10^{21}$ FLOP
  1,000,000 sec on a 1 PFLOP/s computer
- Complexity of (opt.) multigrid: $1000N \approx\ 10^{12}$ FLOP
  1,000 sec on a 1 GFLOP/s computer

$\Rightarrow$ **Question:** Are 10 MFLOP/s realistic in modern simulation packages?

Sparse Matrix Vector Multiplication I

Standard sparse matrix vector algorithm: (DAXPY indexed)

```
          DO 10 IROW=1,N
          DO 10 ICOL=KLD(IROW),KLD(IROW+1)-1
10        Y(IROW)=DA(ICOL)*X(KCOL(ICOL))+Y(IROW)
```

Performance rates of the FEATFLOW code with different numbering schemes (Cuthill–McKee, TwoLevel, Stochastic) for matrix vector multiplication:

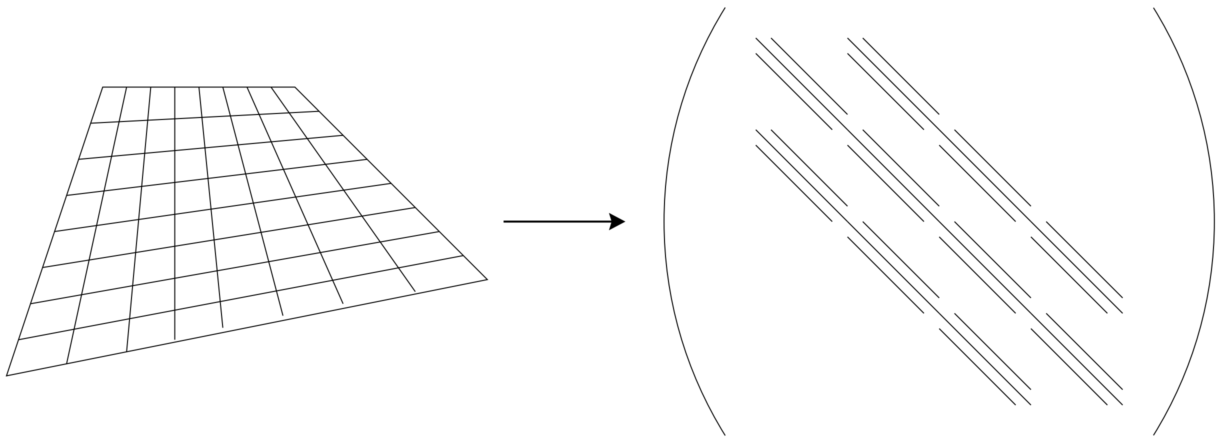
Computer	#Unknowns	**CM**	**TL**	**STO**
Alpha ES40 (667 Mhz)	8,320	147	136	**116**
	33,280	125	105	**100**
	133,120	81	71	**58**
	532,480	60	51	**21**
	2,129,920	58	47	**13**
	8,519,680	58	45	**10**

Sparse Matrix Vector Multiplication II

- sparse techniques basis for most of the recent software packages
- different numberings can lead to identical numerical results and work (w.r.t. arith.ops and data accesses) but to huge differences in CPU time
- sparse techniques 'slow' and depend on problem size and kind of data access

Alternative: Sparse Banded Techniques

Question: How to exploit more performance?
Line- or rowwise numbering:



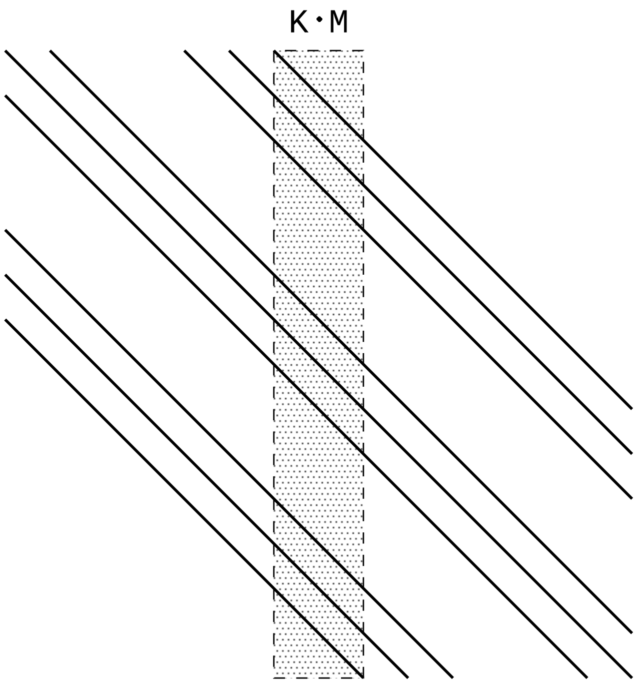
Sparse *banded* matrix vector multiplication:

- FD discretization leads to band structure on tensorproduct meshes
- storing of matrix elements in diagonals
- matrix vector multiplication 'bandwise'
- in equidistant case for certain operators diagonals are constant

LS3

Sparse Banded Techniques I

Bandwise windowed multiplication (variable, constant):




```
      DO 10 IM=1,M/K
       DO 100 I=1,K*M
100    Y(I)  =Y(I)  +DD(I)*X(I)+DL(I)*X(I-1)+DU(I)*X(I+1)

       DO 200 I =1,K*M
200    Y(I-M)=Y(I-M)+LD(I)*X(I)+LL(I)*X(I-1)+LU(I)*X(I+1)

       DO 300 I=1,K*M
300    Y(I+M)=Y(I+M)+UD(I)*X(I)+UL(I)*X(I-1)+UU(I)*X(I+1)
 10   CONTINUE
```

LS3

Measurement

Measurement for efficency: MFlop rate?

Problems:

- x86 Ops $\Rightarrow$ μOps?
- pipeline?
- speculative execution?

$\Longrightarrow$ *numerical* MFlop rate

'Ideal' algorithm for matrix vector multiplication for vectors with length N requires $18N$ operations.

$$\text{MFlop/s} = 18N/t$$

LS3

Sparse Banded Techniques II

2D case	N	DAXPY(I)	MV		MG-TriGS	
			V	C	V	C
Sun V40z	65^2	1521 (422)	1111	1605	943	1178
(1800 MHz)	257^2	264 (106)	380	1214	446	769
'Opteron'	1025^2	**197 (54)**	**362**	**1140**	**333**	**570**
NEC SX-6	65^2	1170 (422)	1204	1354	268	341
(500 MHz)	257^2	1100 (412)	1568	2509	316	459
'Vector'	1025^2	**1120 (420)**	**1597**	**3421**	**339**	**554**
IBM POWER4	65^2	1521 (845)	2064	3612	1386	1813
(1700 MHz)	257^2	1100 (227)	1140	3422	1048	1645
'Power'	1025^2	**390 (56)**	**550**	**2177**	**622**	**1138**

Question: How to use these techniques on complex domains? $\Rightarrow$ **ScaRC**

LS3

ScaRC (Scalable Recursive Clustering) I

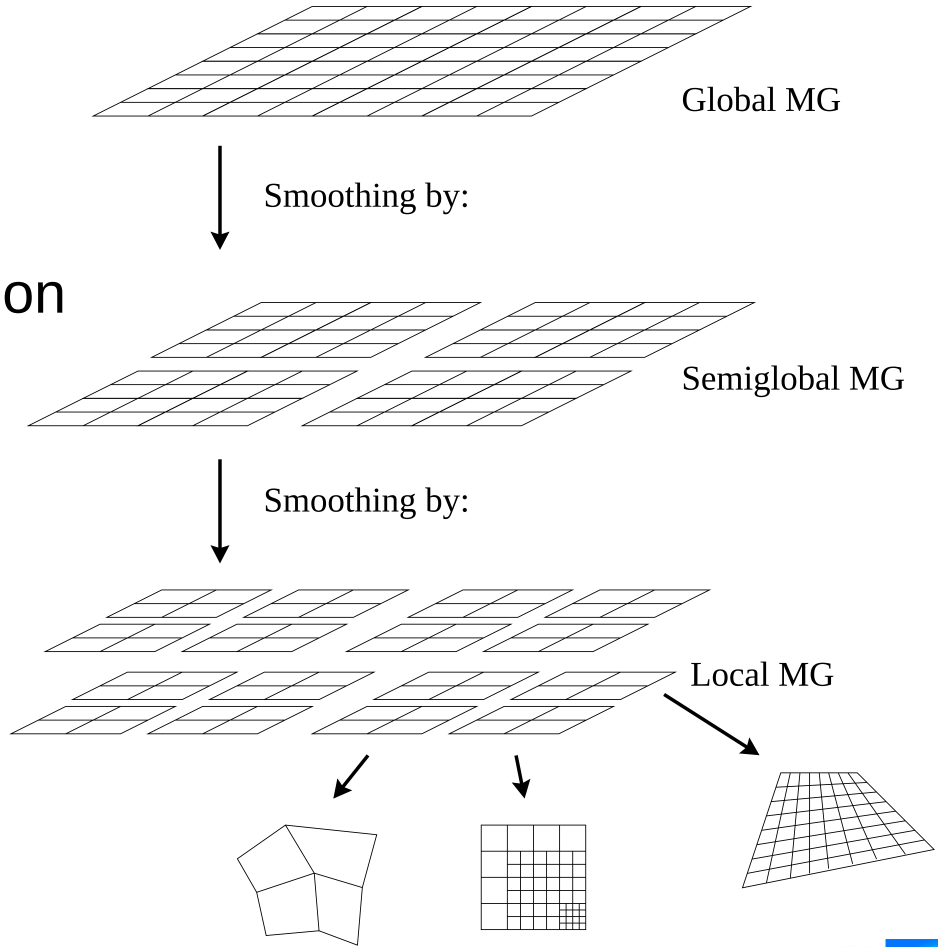
Standard multigrid with (recursively defined) block smoothers

plus

Standard Domain Decomposition with minimal overlap, sequence of coarse grid problems via multigrid

plus

Embedded into standard CG–method



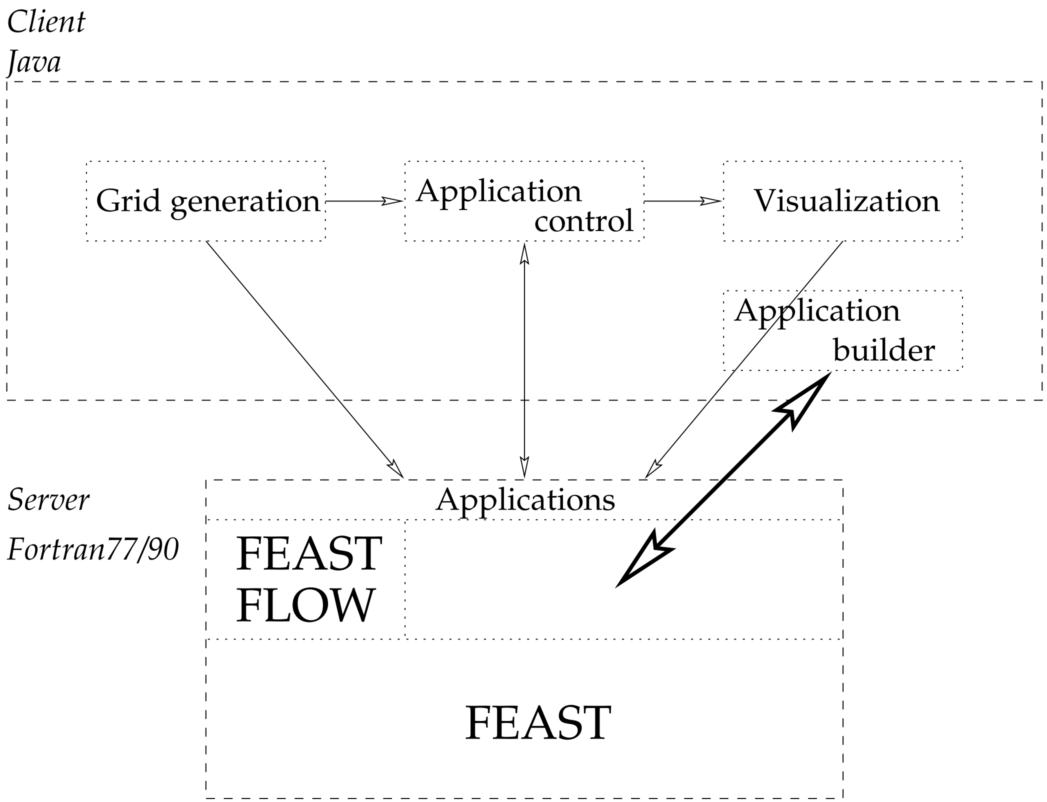
LS3

ScaRC (Scalable Recursive Clustering) II

ScaRC scheme description:

```
SOLVER=CG,16,REL:1.0E-6,HL_SD
  PREC=ALL,1,1.0,HL_SD,1,0
SOLVER=MG,1,REL:1E-6,V,1,1,HL_SD
  SMOOTHER=ALL,0,0.8,HL_SD,1,0
    SOLVER=MG,3,REL:1.0E-6,F,1,1,HL_PB
      SMOOTHER=ALL,0,0.8,HL_PB,1,0
        SOLVER=MG,2,REL:1E-6,F,2,2,HL_MB
          SMOOTHER=ALL,0,1.0,HL_MB,0,0
            LSMOOTHER=TRIGS
          COARSE=CG,256,REL:1.0E-9,HL_MB
            PREC=ALL,1,1.0,HL_MB,0,0
              LSMOOTHER=JACOBI
      COARSE=PGLOBAL
  COARSE=GLOBAL
```

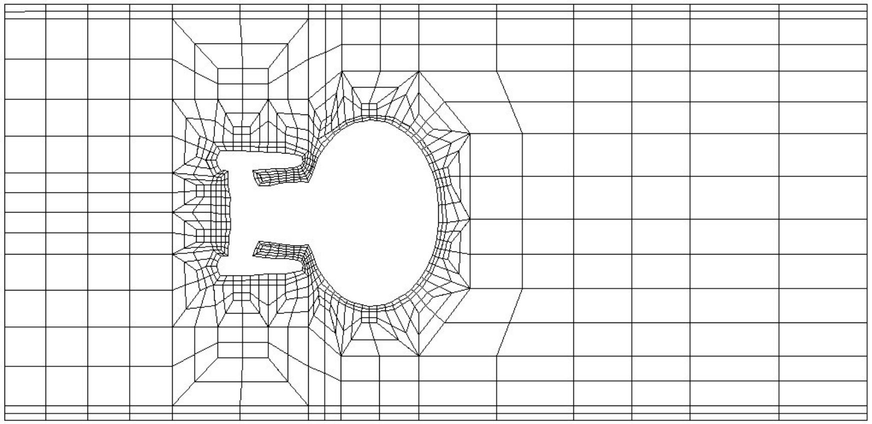
Finite Element Analysis & Solution Tools

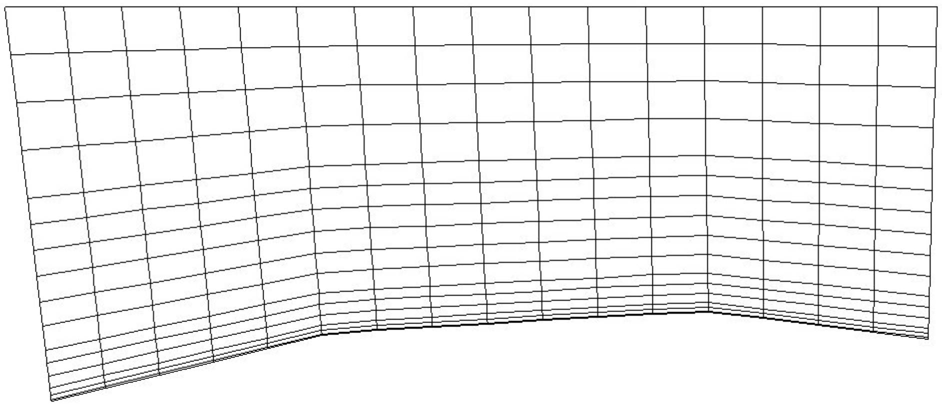


FEAST $=$ SBBLAS + ScaRC + domain decomposition + FEM discretisation

Example: Realization of ScaRC in FEAST I

2D decomposition and zoomed (macro) element (LEVEL 3) with locally anisotropic refinement towards the wall:





LS3

Example: Realization of ScaRC in FEAST II

ScaRC-CG solver (smoothing steps: 1 global ScaRC; 1 local 'MG-TriGS') for locally (an)isotropic refinement

Global (parallel) convergence rates

$\#NEQ$	Dirichlet 'Velocity'		Neumann 'Pressure'	
	$AR \approx 10$	$AR \approx 10^6$	$AR \approx 10$	$AR \approx 10^6$
$210,944$	0.17 (**8**)	0.18 (**8**)	0.21 (**9**)	0.15 (**8**)
$843,776$	0.17 (**8**)	0.17 (**8**)	0.20 (**9**)	0.17 (**8**)
$3,375,104$	0.18 (**9**)	0.19 (**9**)	0.22 (**10**)	0.22 (**10**)
$13,500,416$	0.19 (**9**)	0.18 (**9**)	0.23 (**10**)	0.23 (**10**)

LS3

Example: Total Efficiency

- Poisson problem, multigrid with TriGS smoother, NCC-1701D grid
- Sun V40z, four Opteron 844 CPUs with 1800Mhz, 16 GByte memory
- CPU times and numerical MFlop rates for different numbers of CPUs

N	1p	2p	3p	4p
843,776	11.04(191)	5.72(368)	3.85(547)	3.45 (611)
3,381,507	30.36(271)	15.62(526)	10.73(766)	8.42 (976)
13,513,219	98.64(328)	51.04(634)	34.79(931)	27.80(1165)
54,027,267	367.85(301)	198.35(559)	129.07(859)	107.70(1029)

Outlook

Further work has to be done for

- optimal SparseBandedBLAS for different architectures
- optimized multigrid and ScaRC driver
- dynamic loadbalancing