Recent results with \textit{elsA} on multi-cores

Michel Gazaix (ONERA)
Steeve Champagneux (AIRBUS)

October 15th, 2009
Outline

- Short introduction to elsA
- elsA benchmark on HPC platforms
  - Detailed performance evaluation
    - IBM Power5, AMD Opteron, INTEL Nehalem
    - Scalability: towards massively parallelism
- elsA installation on HP 10 000 cores machine
- First experience with GPU
  - Wall distance computation
- Which HPC strategy for elsA?
**elsA**: multiblock structured compressible flow solver

Developed by ONERA and CERFACS since 1997

- Multi-block, structured
  - "Unstructured" block coupling topology
    - 'match', 'nearmatch', 'nomatch' block interface, Chimera
- Cell-center
- External and internal flow simulations
- Multi-Physics
  - Aero-elasticity, Aero-acoustics, Aero-thermics
- Many turbulence models available
  - (U)RANS, DES, LES
- Object-Oriented architecture
  - Implementation language: C++ / Fortran
  - User interface: Python
Parallelism inside elsA

- Parallelism MPI
- SPMD programming model
- Coarse-grain parallelism
  - Each block is allocated to a specific processor
  - Several blocks can be allocated on the same processor
- MPI communications:
  - Exchange of data between adjacent blocks (interface coupling)
  - Treatment of global algorithmic features
    ⇒ global time step, non local boundary conditions
    (ex: prescribed flow rate)
- No support for OpenMP thread parallelism
Recent results with *elsA* on multi-cores

*elsA* benchmark on HPC platforms
Towards massively parallel (≥ 1000 processors)

Scalability

- The goal is to keep good performances for an application for an increasing suite of problem size and associated computers
- In HPC, two common notions of scalability: strong and weak scaling

**Strong scaling**
- How the solution time varies with NPROC for a **fixed total problem size**
- Parallelism used to reduce restitution time (Ex: Weather forecast)
- Good candidate in CFD: long unsteady computations
  - Complex physical model: DES, LES, DNS
  - Complex geometry: Multistage turbomachinery, Helicopters,...

**Weak scaling**
- How the solution time varies with NPROC for a **fixed problem size per processor**
- Parallelism used to increase problem size (Ex: Climate modelling)
  - Very common situation
Recent results with *elsA* on multi-cores

Towards massively parallel ($\geq 1000$ processors)

Strong Scalability is difficult in CFD

- **In practice, strong scaling very difficult to achieve for a Finite-Volume solver**
  - Mean block size decreases when NPROC increases
    - *Ratio (communication / computation) increases with NPROC*

- **Specific problem of structured grid load balancing**
  - Achieving good load balancing far from trivial
    - *Respect of multigrid constraints*

- **Computed solution should be independent of the number of processors**
  - Not strictly true if block splitting used
    - *Convergence of block-based implicit algorithm can decrease*

  *In practice, does not seem to be a significant problem*

  *However, time needed to get the "asymptotic regime" in URANS may depend on block splitting*
Recent results with *elsA* on multi-cores

Many platforms have been benchmarked

Time Consuming...

- NEC SX8+ (ONERA), NEC SX-9 (METEO FRANCE)
- IBM PowerPC (BSC MareNostrum)
- IBM BlueGene/L (CERFACS) and BlueGene/P (EDF)
- SUN, AMD Opteron (C²A²S²E cluster)
- BULL, Intel Itanium 2 (ONERA)
- BULL, Intel Nehalem (CCRT)
- HP, Intel Nehalem (AIRBUS)
- CRAY XT5, processor Istanbul (12 cores / node)
Recent results with *elsA* on multi-cores

**elsA : study of (weak) Scalability**

**Toy problem**

- One identical block allocated to each processor
- No physical boundary, only matching joins
- Each processor has exactly the same computing task
- Each processor has exactly the same amount of communications
  - Identical number and size of MPI messages
- "Ideal" load balancing

M. Gazaiχ, S. Champagneux, DLR Symposium "CFD on Future Architectures...", Oct 14th-15th, 2009
Recent results with *elsA* on multi-cores

**Scalability *elsA* on BlueGene/L**

IBM BlueGene (weak) scalability

CPU time (normalized with Nproc=8)

Nproc (8 <-> 2048)
Recent results with *elsA* on multi-cores

Scalability *elsA* on $C^2A^2S^2E$ cluster

*elsA* scalability: $C^2A^2S^2E$ cluster and IBM BlueGene/L
(Nproc=6, 16, 32, 64, 128, 256, 512, 1024, 2048)
Recent results with *elsA* on multi-cores

*elsA* installation on HP 10 000 cores machine

M. Gazaix, S. Champagneux, DLR Symposium "CFD on Future Architectures...", Oct 14th-15th, 2009
Recent results with *elsA* on multi-cores

**Scalability** *elsA* on HP 10 000 Nehalem cores machine
Lessons learned on HP 10 000 Nehalem cores machine

- **Maximum number of cores**: 8192
  - Poor reproducibility for NPROC larger than 4096
- **Maximum number of blocks**: 17 000
- **Maximum mesh size**: \(12 \times 10^9\) mesh points
- **Disturbing computing time variability**
  - Often observed for large NPROC
  - Make difficult scientific process to improve performance
    - *Ex*: tuning splitting block algorithm
- **Not very stable**
  - *ex*: `MPI_Sendrecv_replace`: Internal MPI error
- **HP Performance analyse tools**: not very easy to use
  - Specially for large NPROC
Recent results with *elsA* on multi-cores

**Scalability**

Strong scaling : difficult for a CFD structured code!

- **Splitting of existing blocks:**
  - Increase number of ghost cells (additional computations)
  - Increase of data transfer through connecting network
  - Implicit algorithm degradation

- **Global communication often more and more costly** ($\log(Np)$ stages)
  - Logarithmic scaling

- **Concrete example : DPW4 (Civil aircraft)**
Drag Prediction Workshop (DPW4)
Collaboration Airbus / Onera / C-S

- Wing - Fuselage - HTP
- Fine mesh : 47 millions cells
- Extra-Fine mesh :
  - Airbus : 103 millions cells
- "Huge" mesh
  - $1.6 \times 10^9$ cells
Recent results with *elsA* on multi-cores

DPW4 Huge mesh ($1.7 \times 10^9$ mesh points)
Recent results with *elsA* on multi-cores

**DPW4 Huge mesh** *(1.7 \times 10^9 mesh points)*
Recent results with *elsA* on multi-cores

**DPW4 Huge mesh** *(1.7 \times 10^9 mesh points)*
DPW4 : Strong scalability (Fine mesh, IBM BlueGene/L)
Recent results with *elsA* on multi-cores

**DPW4 : Strong scalability ("Huge" mesh, 1 point over2)**

![Graph showing strong scalability](image)
Recent results with *elsA* on multi-cores

**DPW4 : Strong scalability ("Huge" mesh)**

![Graph showing strong scalability with DPW4](image)
Recent results with elsA on multi-cores

**MPI elsA Efficiency**

- **Correct in most cases if NPROC ≤ 1024 for matching joins**
  - Requires a minimum number of cells per processor (approx. $50 \times 10^3$)

- **Work in progress for further improvements:**
  - Overlapping computations / (non blocking) communications
  - Message concatenation
    - Useful when latency is important
  - Load balancing algorithm adaptation
    - Currently: only number of cell is taken into account
    - Add an option to include also communication costs;
      requires setting of hardware (CPU, network) and software (MPI implementation) parameters

- **Work done (C-S) to improve scalability for non-matching joins**
  - Supported by AIRBUS, to be integrated in main elsA version

- **Scalability evaluation for Chimera in progress**

---

M. Gazeix, S. Champagneux, DLR Symposium "CFD on Future Architectures...", Oct 14th-15th, 2009
Recent results with *elsA* on multi-cores

First experience with GPU

M. Gazaix, S. Champagneux, DLR Symposium "CFD on Future Architectures...", Oct 14th-15th, 2009
First experience with GPU : Collaboration with C-S
Wall distance computation (1)

- Needed for most turbulence models
- Usually done before entering iterative loop
  - Steady case : can be saved on disk for future runs with identical geometry
  - Mesh deformation : re-computation necessary
- Can be costly
  - DPW4 Huge mesh (1.7 Billion points) : 8 hours on 2048 procs
- Algorithm relatively easy to translate to GPU
First experience with GPU : Collaboration with C-S

Wall distance computation (2)

- **GPU Nvidia Tesla C1060**
  - Graphical processor Nvidia T10 (extension of GT200 architecture)
  - Graphical memory : 4GB
- **CUDA : Use of Fortran API**
- **Performance**
  - Gain between 3 and 10 (depending of problem size)
Recent results with elsA on multi-cores

elsA strategy for massively parallel computations

Increase of HPC impact in CFD (1)
More computations, bigger, more accurate

- **Increase of simulation size**
  - Geometric details more and more taken into account
    ⇒ "technological effects"
  - Needs of increase accuracy
    ⇒ "Current meshes are probably not fine enough"

  *Conclusion DPW III*

  (http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop3)

- **High-order numerical methods**
  - Discontinuous Galerkin (DG), Spectral Volume...

- **Increase of physical modelling complexity**
  - DES or LES replacing RANS...
Increase of HPC impact in CFD (2)

- Increase of fully unsteady approach
  - Capturing relevant frequencies often very costly
    - Unsteady simulation of a multi-stage compressor
  - Coupling with flight mechanics: "Flying by the equations"
    - Estimation: 25 times more costly than a single steady computation

- Multi-physics: Aerodynamics / Structure / Acoustics / Thermics

- Production of data base ("Flying through the database")
  - Very large number of computations required
    - AIAA 2008-712 (Rossow / Kroll): \(10^5\) simulations

- Larger physical time simulation (better statistics)
- Uncertainties quantification (Polynomial chaos, Monte Carlo...)
- Increase of simulation versus experiment
- Optimisation loop
Increase of HPC in science
No more reserved to specialists

- Simulation capability: key for competitiveness
- End of CPU frequency increase
  - Scientific software must change
Recent results with *elsA* on multi-cores

**Paradigm change: 2005**

**Need for Parallelism**

![Graph showing the need for parallelism over time](image)
Which strategy for next five years?

Some certainties . . .

- End of CPU clock frequency increase
  - Heat dissipation, electric consumption

- Flop/Watt ratio increasingly important ("green" computing)

- "Memory wall problem"
  - Memory access and network will remain much too slow
    (compared with CPU)

- Multi-core architecture unavoidable
  - Memory bandwidth probably not sufficient for structured CFD code
Recent results with *elsA* on multi-cores

Which strategy for the next five years

Adaptation specific to multi-core

- Memory bandwidth will not grow as fast as the number of cores
- Multi-core will force specific *elsA* optimizations

  ▶ Increase spatial locality

    ⇒ For ex: internal loop using the biggest dimension (IJK)
    ⇒ Spatial locality inherently good in a structured code, not easy to increase significantly

  ▶ Increase temporal locality

    ⇒ Re-use data as much as possible

    not easy for a multi-application software such as *elsA*, with many options implemented in a modular way (to reduce maintenance costs)
Recent results with *elsA* on multi-cores

**Which strategy for the next five years**

Use of specialized processors

- Massive effort needed to port *elsA*
- Technology rapidly evolving, much faster than a scientific code...
  - *elsA*: born in 1997...
- **GPU**: Graphical processor (NVIDIA, ...)
  - Work will be done mostly on simple CFD kernels
  - *If successful, adapted to* *elsA*
- **IBM Cell**
  - *elsA*: Limited experience with Cell processor (2008)
  - *at that time, programming environment was much too unstable*
Recent results with * elsA * on multi-cores

**Strategy for the next five years**

- Improve scaling
- **Challenge of load balancing**
  - Composite mesh (HyperFlex)
    - \( \Rightarrow \) Structured / Unstructured, Cartesian, Chimera
  - Multi-physics coupled calculations
- **Multi-core architectures**
  - Hybrid OpenMP - MPI
    - \( \Rightarrow \) OpenMP inside SMP nodes, MPI between nodes
      - may decrease load balancing complexity
- I/O parallelisation
- **Experiment with GPU**
  - May require restructuring computing kernel
  - New language to master
- **Investigate "parallel-friendly" algorithms**
  - Ex : Time Spectral Method instead of URANS \( \longrightarrow \) better (strong) scaling

M. Gazaix, S. Champagneux, DLR Symposium "CFD on Future Architectures...", Oct 14th-15th, 2009
Recent results with *elsA* on multi-cores

**Conclusion**

- **We must not miss transition to massively parallelism**
- **elsA** is a relatively large software project
  - Size imposes effort coordination
- **Collective work:**
  - Expertise in CFD
  - Expertise in computing science (parallel algorithms, parallel architectures)
  - Collaboration with specialists
  - ⇒ *Multi-core, GPU*
- **Additional information:**
  - N. Gourdain,
  - L. Gicquel, M. Montagnac, O. Vermorel, M. Gazaix,
  - G. Staffelbach, M. Garcia, J.F. Boussuge, T. Poinsot:
  - High Performance Computing of Industrial Flows: Application to Aeronautic and Propulsion Challenges
  - von Karman Institute for Fluid Dynamics, Lecture Series 2009-05