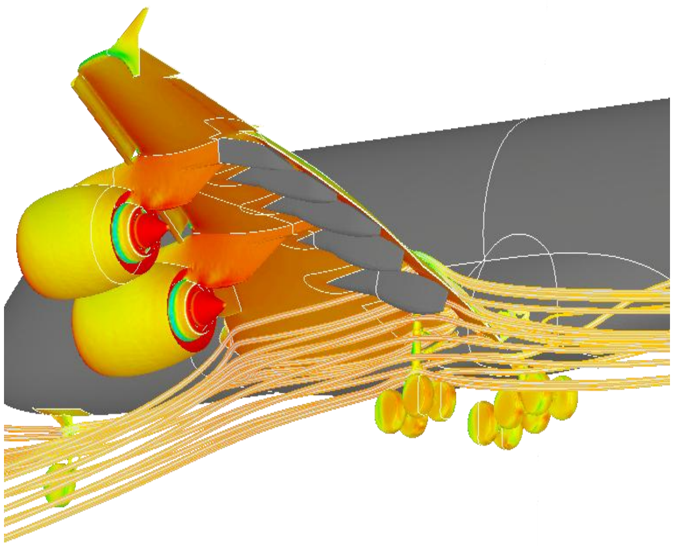
Petascale in CFD
Cetraro, 29th of June, 2011

Christian Simmendinger, T-Systems Solutions for Research

Overview

The TAU Solver



Scalability

- Core (Instruction Level)
  - SIMD – SRC-SRC
- Thread (Task Level)
  - Beyond OpenMP
- System Level
  - GASPI / PGAS API

•Maintainability of the Code
•Programmability
•Performance Portability

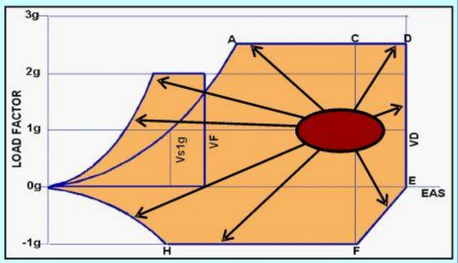
DLR

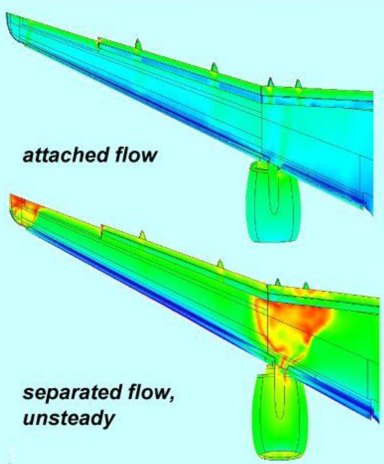
T · · Systems

The TAU Solver
Vision: Digital Aircraft

TAU: 3D Finite Volume Reynolds Averaged Navier Stokes (RANS) Solver

Full flight envelope coverage: *CFD mostly done near cruise point*





configurations:

clean

airbrakes deployed

high lift

50	flight points
100	mass cases
10	a/c configurations
5	maneuvers
20	gusts (gradient lengths)
4	control laws

~ 20,000,000 simulations

Engineering experience for **current** configurations and technologies

~ 100,000 simulations

Programmability , Maintainability and Performance Portability: Instruction Level

SSE2/SSE4.2/AVX/MIC: From 128 bit SIMD to 512 bit SIMD.
16 DP Flop/Cycle – or just one.

- Statistical Fact:
  Compiler performance doubles every 18 years. Do not expect compilers to resolve the SIMD issue for you. Especially C/C++ takes a severe performance hit due to aliasing.

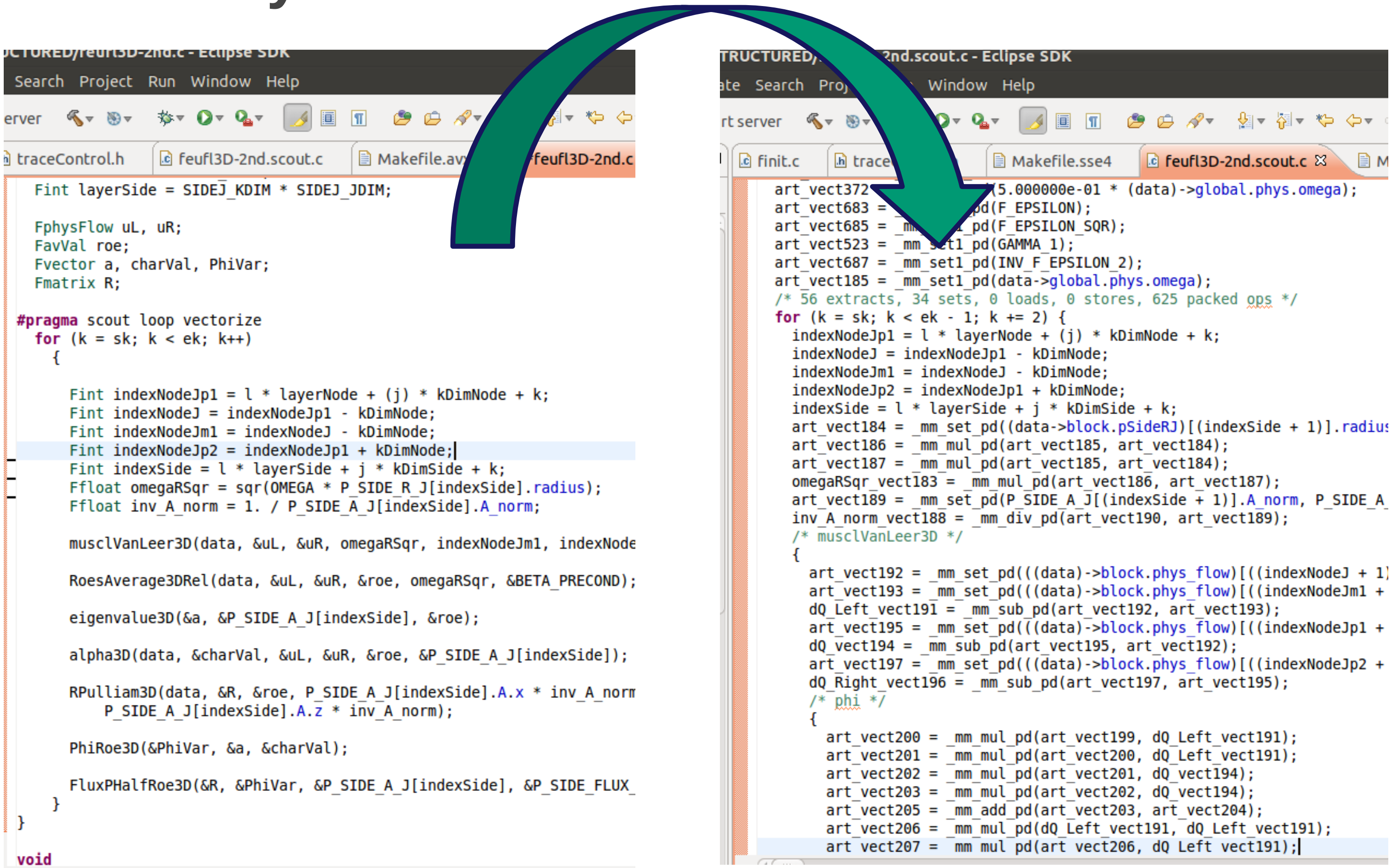
SIMD Intrinsics

- Maintainability
- Programmability
- Performance-Portability

Scalability

- Core (Instruction Level)
  - SIMD – SRC-SRC
- Thread (Task Level)
  - Beyond OpenMP
- System Level
  - GASPI / PGAS API

Programmability , Maintainability and Performance Portability: Instruction Level



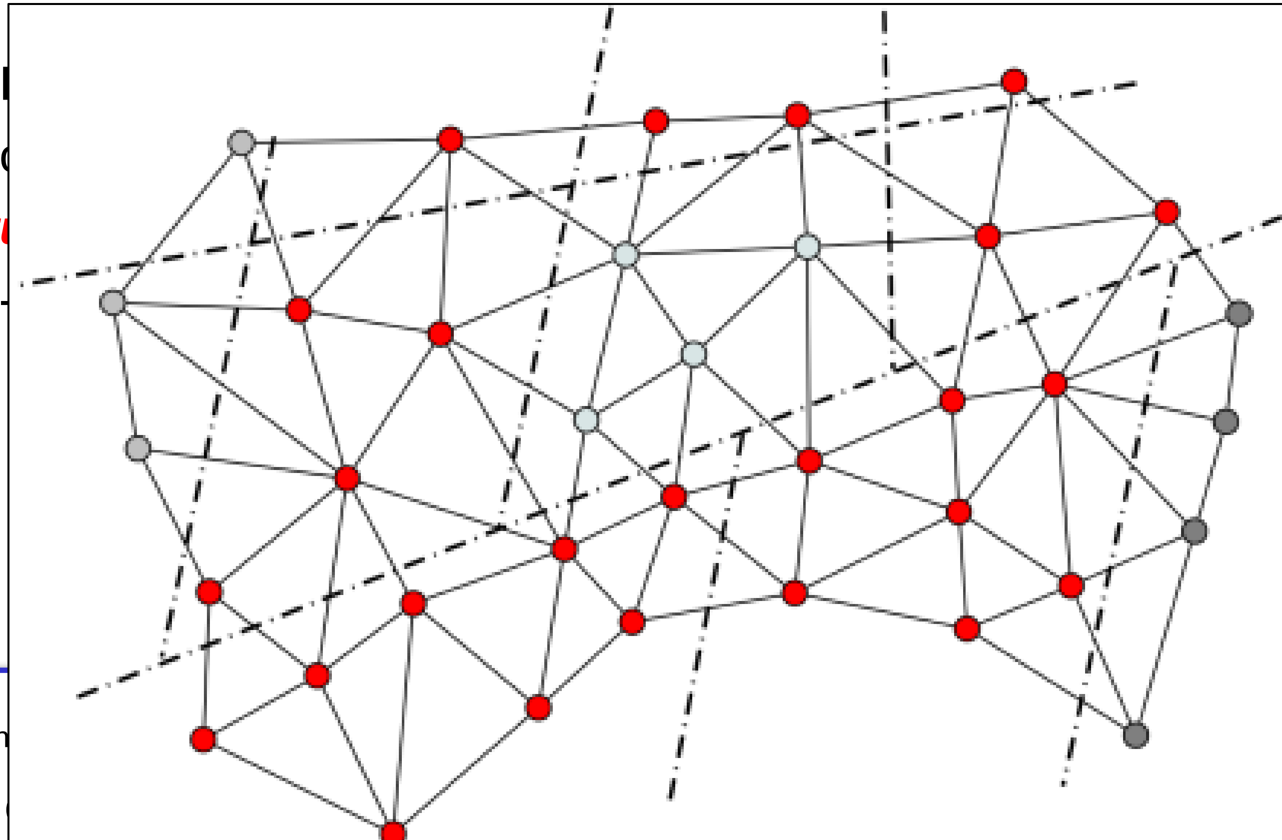
Programmability , Maintainability and Performance Portability: Thread Level

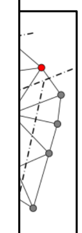
From MI

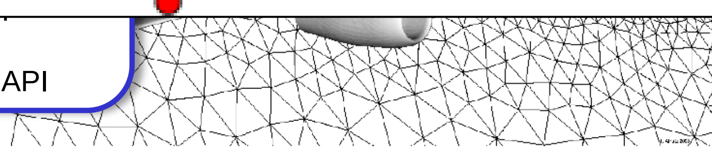
- Resc

  Mutu

- Sub-







Scalability

- Core (In
  -
- Thread (
  - Beyond OpenMP
- System Level
  - GASPI / PGAS API

Programmability , Maintainability and Performance Portability: Thread Level

- *Mutual Exclusion*

```
for (color = fcolor; color != NULL; color = color->succ)
   for (face = color->start; face < color->stop; face++)
```

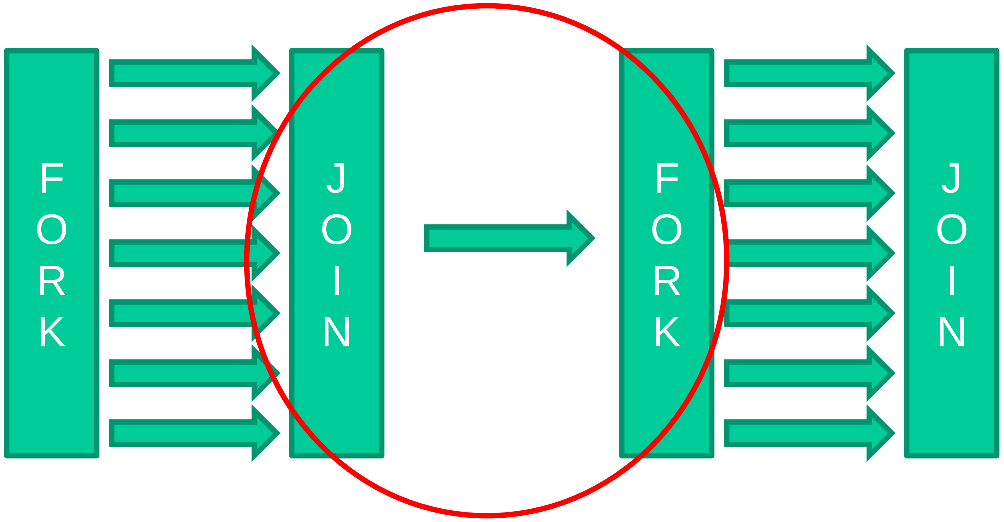
```
for(color=get_color();color != NULL; color=get_next_color(color))
   for (face = color->start; face < color->stop; face++)
```

- Maintainability
- Programmability
- Performance Portability

Programmability , Maintainability and Performance Portability: Thread Level

OpenMP – The Fork Join Model

- Incremental Parallelisation
- Amdahl Trap (Barrier and Load Imbalance)



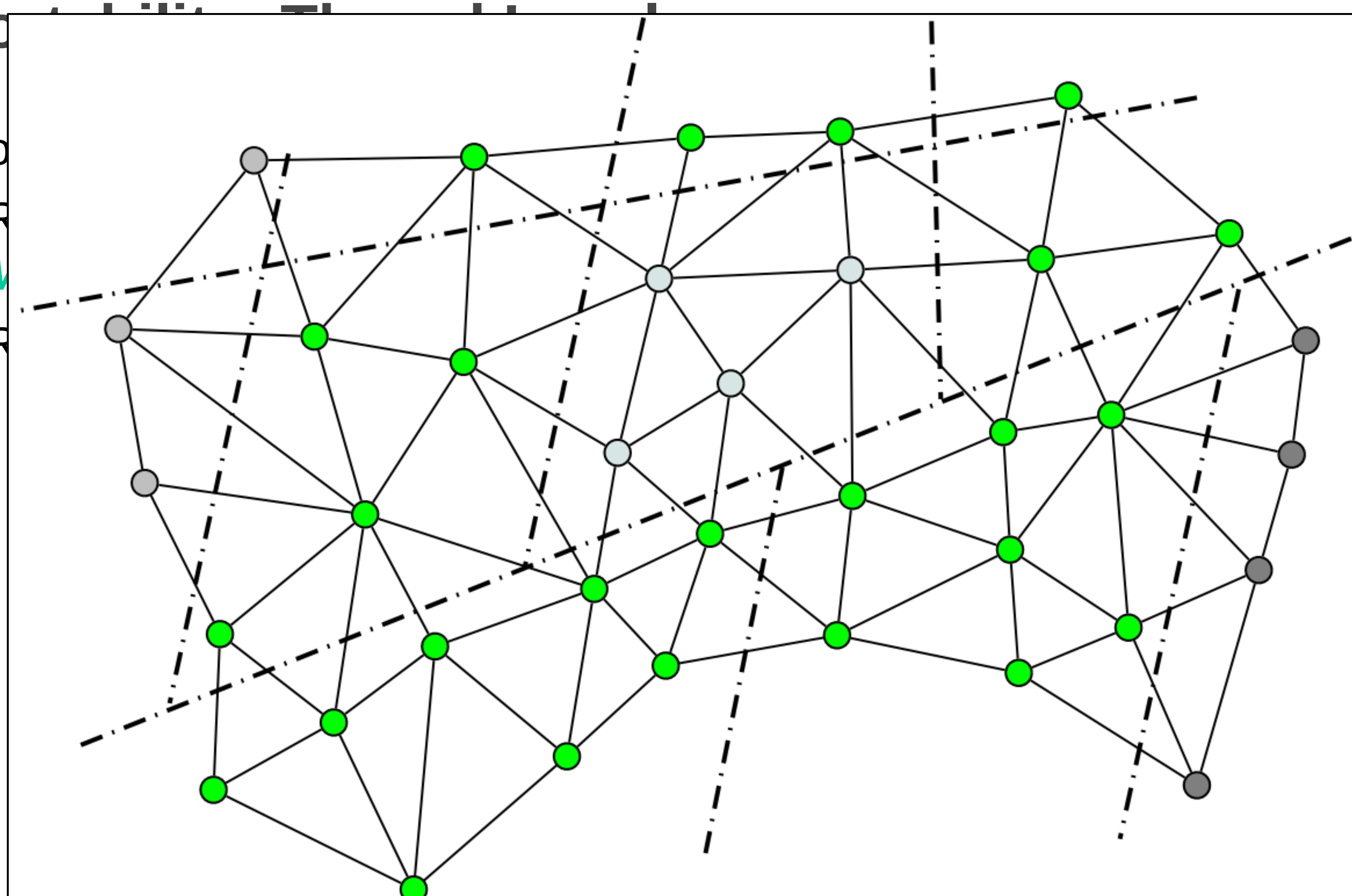
Programmability , Maintainability and Performance Po

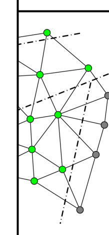
Beyo

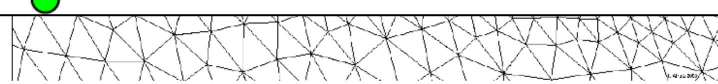
- R

  M

- R







Programmability , Maintainability and Performance Portability: Thread Level

- *Mutual Completion*

```
for(color=get_color();color != NULL; color=get_next_color(color))
   for (face = color->start; face < color->stop; face++)
BARRIER;
for(pnt = 0; pnt < nown; pnt++)
```

```
for(color=get_color();color != NULL; color=get_next_color(color))
   for (face = color->start; face < color->stop; face++)
while(get_pnt_range(&pstart, &pstop))
  for (pnt = pstart; pnt < pstop; pnt++)…
```

- Maintainability
- Programmability
- Performance Portability

Programmability , Maintainability and Performance Portability: Thread Level

This Programming Model is ***Generic.***

```
for(k = 0; k < KDIM; k++)
   for (j = 0; j < JDIM; j++)
      for (i = 0; i < IDIM; i++) {};
BARRIER;
for(k = 0; k < KDIM; k++) …
```

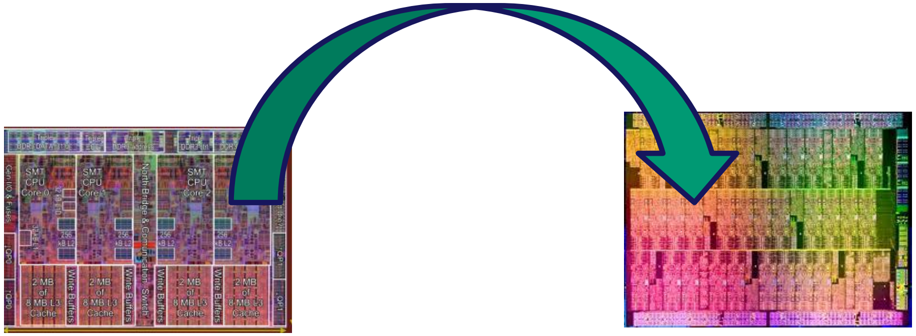
```
for(k = get_k_index(); k != -1; k = get_next_k_index(k))
   for (j = 0; j < JDIM; j++)
      for (i = 0; i < IDIM; i++) {};
 for(k = get_k_index(); k != -1; k = get_next_k_index(k)) …
```

J.Jägersküpper (DLR), C.Simmendinger (T-Systems SfR). A Novel Shared-Memory Thread-Pool Implementation for Hybrid Parallel CFD Solvers, EuroPar 2011, (to appear).

Programmability , Maintainability and Performance Portability: Thread Level

This extension of the OpenMP Fork-Join Programming Model is

- fully asynchronous
- data flow driven
- implicitly load balanced
- exclusively uses local locks instead of global barriers
- expected to scale to O(10^2) cores



Programmability , Maintainability and Performance Portability: System Level

In a Partitioned Global Address Space every thread can read/write the entire global memory of the application.

- Opportunities for improved programmability
  (PGAS languages like UPC, CAF, Chapel and others)
- Opportunities for improved ***Scalability***
  (GPI (Fraunhofer ITWM), GASPI)

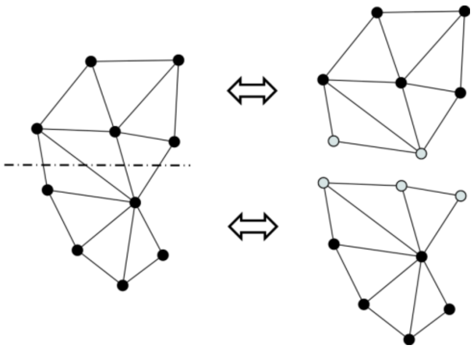
Global Address Space Programming Interface (GASPI)

A PGAS API which aims at replacing bulk-synchronous communication with asynchronous, 1-sided, RDMA driven communication patterns.

Scalability

- Core (Instruction Level)
  - SIMD – SRC-SRC
- Thread (Task Level)
  - Beyond OpenMP
- System Level
  - GASPI / PGAS API

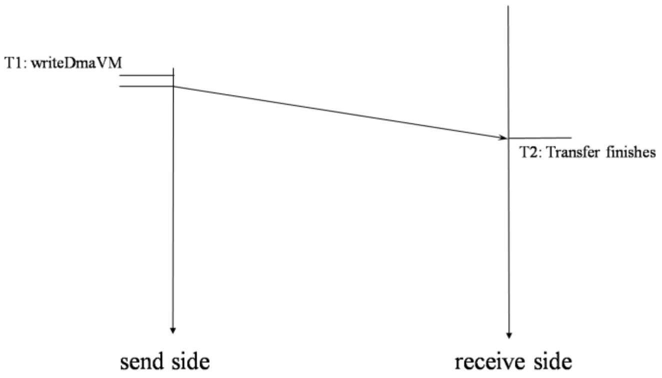
Programmability , Maintainability and Performance Portability: System Level



Overlapping Communication and Computation

```
if (this_thread_exchanges_pnt_data(g)) /* lock halo */
{
        exchange_pointdata();
        post_exch_pnt_data();  /* unlock halo*/
}
```

Programmability , Maintainability and Performance Portability: System



Multithreaded send/recv

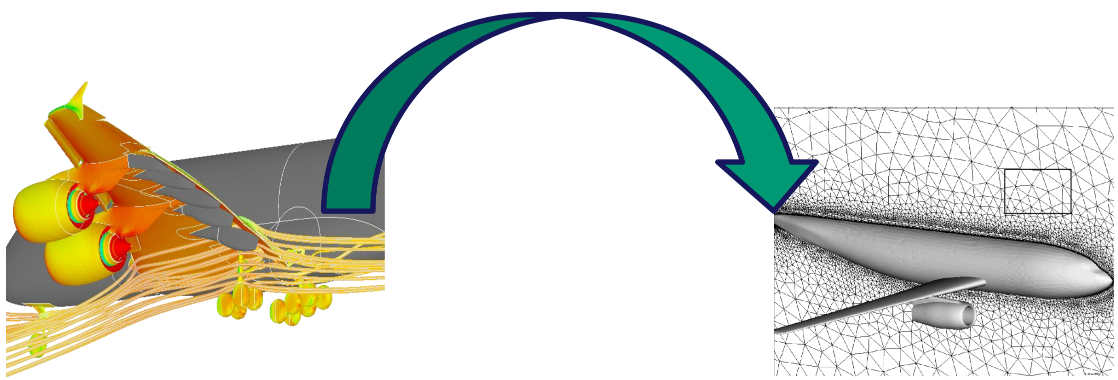
```
if ((sid = get_seq_num_and_lock(g)) < num_send_threads(g))
        send_dbl_threaded(comap, comap->ncommdomains, comap->commpartner,
        data, dim2, key);


If ((sid = thread_seq_num()) <  num_recv_threads(g))
        recv_dbl_threaded(g, num_recv_threads, sid,
        comap, comap->ncommdomains, comap->commpartner, data, dim2, key);
```

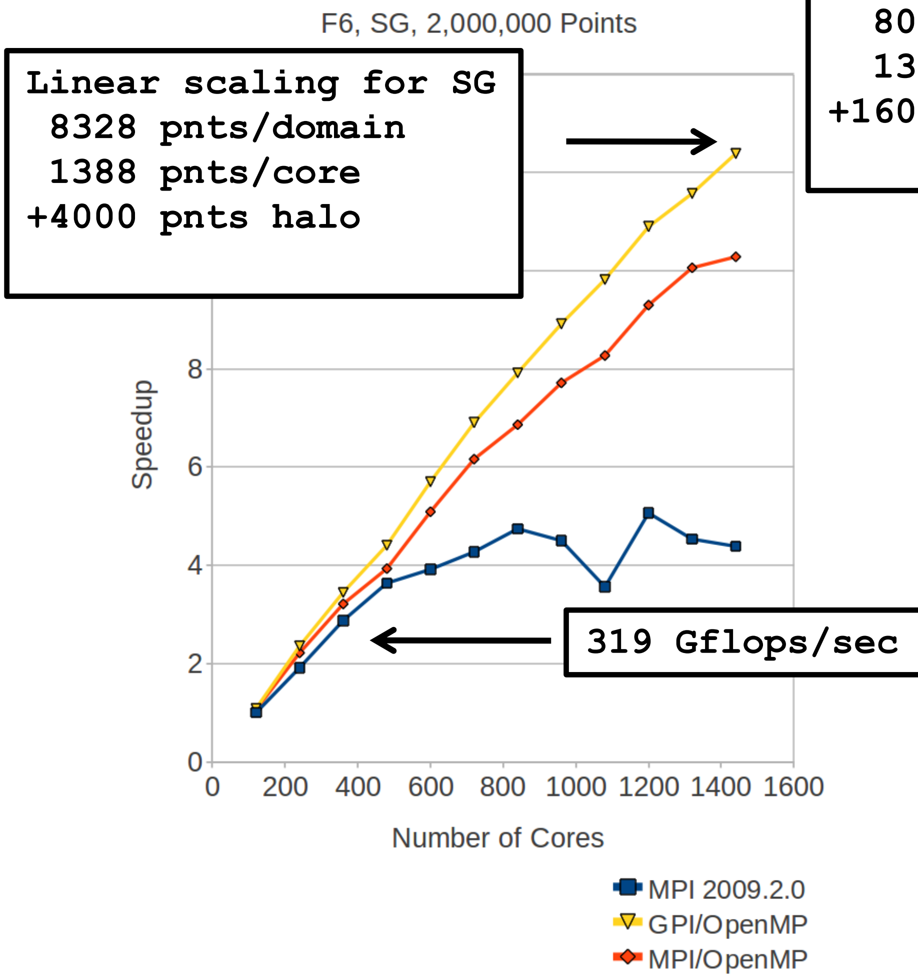
Programmability , Maintainability and Performance Portability: System Level

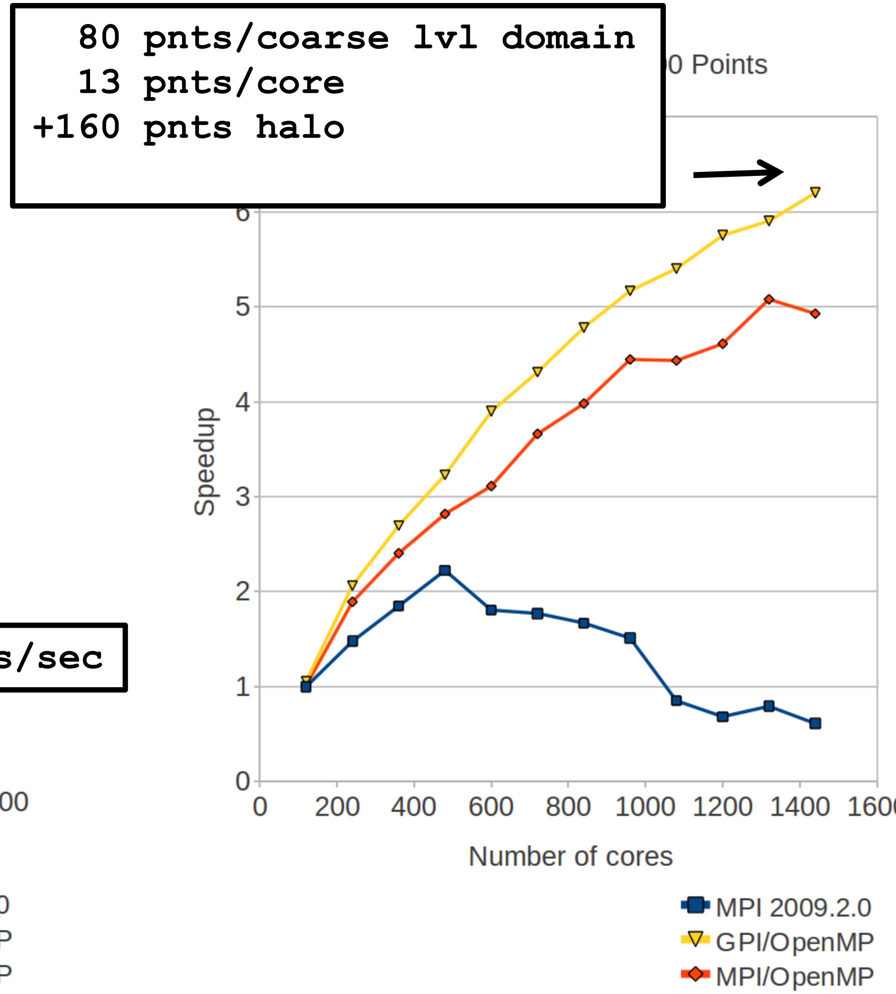
Strong Scaling Use Case: **How far can we scale ?**

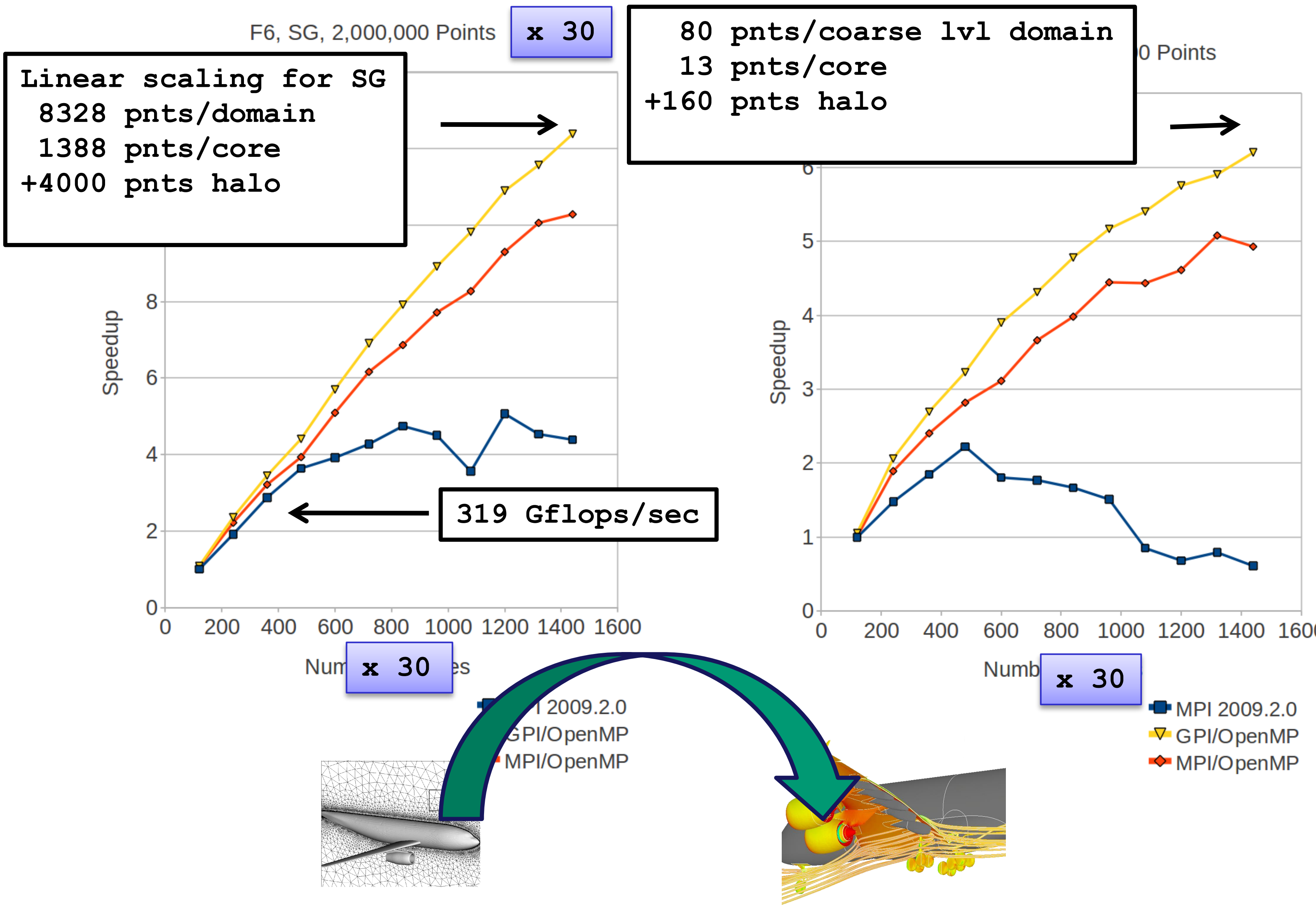
4W Multigrid, 1 Iteration Step.

- 1 Allreduce
- 140 Next Neighbour Halo Exchanges



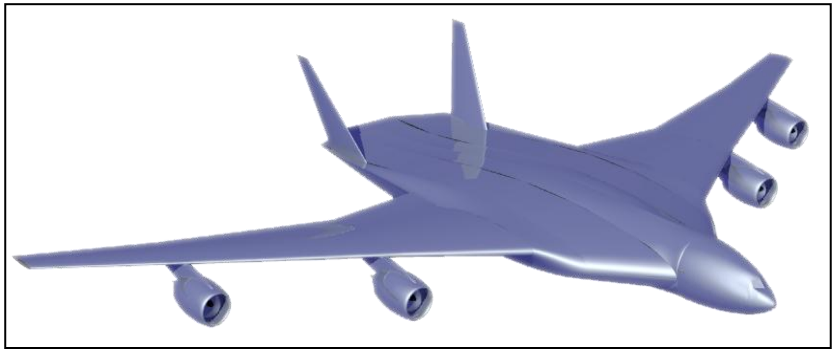

F6, SG, 2,000,000 Points
Linear scaling for SG
8328 pnts/domain
1388 pnts/core
+4000 pnts halo
319 Gflops/sec
Speedup
0
2
4
6
8
0 200 400 600 800 1000 1200 1400 1600
Number of Cores
MPI 2009.2.0
GPI/OpenMP
MPI/OpenMP

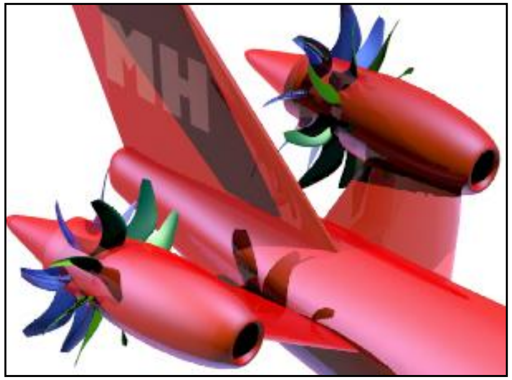

80 pnts/coarse lvl domain
13 pnts/core
+160 pnts halo
0 Points
Speedup
0
1
2
3
4
5
6
0 200 400 600 800 1000 1200 1400 1600
Number of cores
MPI 2009.2.0
GPI/OpenMP
MPI/OpenMP


F6, SG, 2,000,000 Points
x 30
Linear scaling for SG
8328 pnts/domain
1388 pnts/core
+4000 pnts halo
319 Gflops/sec
Speedup
0
2
4
6
8
0 200 400 600 800 1000 1200 1400 1600
x 30
80 pnts/coarse lvl domain
13 pnts/core
+160 pnts halo
0 Points
Speedup
0
1
2
3
4
5
6
0 200 400 600 800 1000 1200 1400 1600
Numb
x 30
MPI 2009.2.0
GPI/OpenMP
MPI/OpenMP

T - Systems

Summary





Scalability

- Core (Instruction Level)
  - SIMD – SRC-SRC
- Thread (Task Level)
  - Beyond OpenMP
- System Level
  - GASPI / PGAS API / GASPI

We are confidemt, that we can reach O(Petascale) within the constraints of a typical production run.