

Parallel space charge computation with Parallel Astra

**CANDLE – DESY – PSI
COLLABORATION WORKSHOP**

Sascha Meykopff, DESY
July 1st, 2010
CANDLE, Yerevan

And Now for Something Completely Different



- Astra overview
- Parallel work history
- Project restrictions
- Scaling issues
- Parallelization of the cylindrically space charge solver
- Test case
- Measure results
- Outlook
- Conclusion

Astra provides 3 particle-in-cell (PIC) algorithms.

Cylindrically symmetric grid algorithm is the most frequently used.

Single cells are addressed by a radial and a longitudinal bin number.

Astra iterates over all particles to determine the charge present in each cell.

Tracking loop iterates over all particles and take external fields and local space charge into account.

The forces exerted on each particle are integrated with a fourth-order Runge-Kutta algorithm.

The tracking code is difficult to parallelize because each tracking step depends on the last one.

We distribute n particles among m processes.

Every process calculates only a part of the particle distribution.

Heavy synchronization after each time step.

2005: Initial parallel branch by Lars Fröhlich

April 2007: Complete restart based on latest Astra source, svn r1

Mai 2007: First working version

July 2007: XML Aperture support

March 2008: Backport changes from Klaus

Dez. 2009: Backport updated 2D-space charge solver from Klaus

Jan. 2010: Speedup external field calculations, new parser

Feb. 2010: Parallelized 2D space charge solver, requested by
Igor

June 2010: Prepare 3D space charge parallelization

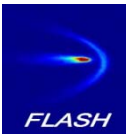
- The general program structure is fixed, otherwise backports 'll be much more difficult.
- The numerical calculations are fixed, or we can't call it Astra.
- The supported functions must be equal.
- More features 'll be unused.
- User friendly, static linked mpi library

Scaling problems detected



DESY, Zeuthen reported a scaling problem with Parallel Astra.
The problem was confirmed on our new ferrari3 cluster.
The old Ferrari2 consists of only 24 cores.

Parallelization of the cylindrically space charge solver



Precondition: particle data is synchronized on all cores if
setup field is called

determine all used subroutines, profiling code

most work is done inside `int_ring` function (integrates
greensfunction over ring from r_1 to r_2 and z_1 to z_2 using a
modified gauss quadrature), called $2 \cdot N_{\text{long}} \cdot N_{\text{rad}}^2$ times

remove all global and static variables from subroutines (
FORTRAN SAVE parameter, COMMON blocks, MODULE)

gotos are already removed

extract special cases

massive loop unrolling → concentrate `int_ring` calls in 2
functions, output depends only on input variables!

distribute precalculations of `int_ring`

synchronize result arrays afterwards (`mpi_allgather`)

Test case:

Astra input file from DESY, Zeuthen, PITZ 1.8

200k particles, tracking stop: 10m, cylindrically space charge,
Nrad: 40, Nlong: 80

2 cavities, 1 solenoid

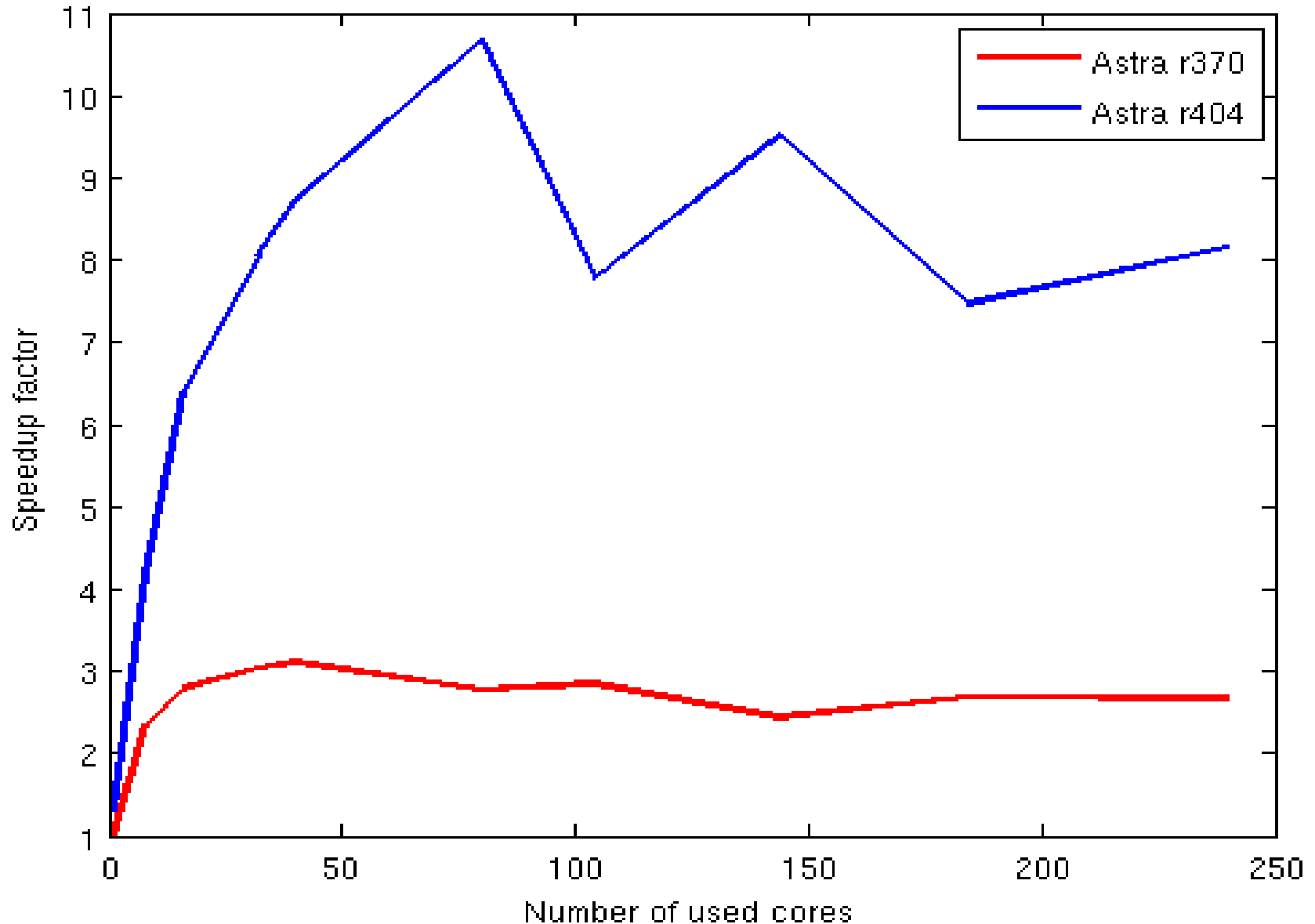
Test 1: start at 0m, 7 screens, 3000 emittance calculations

Test 7: start at 3m, reduced output

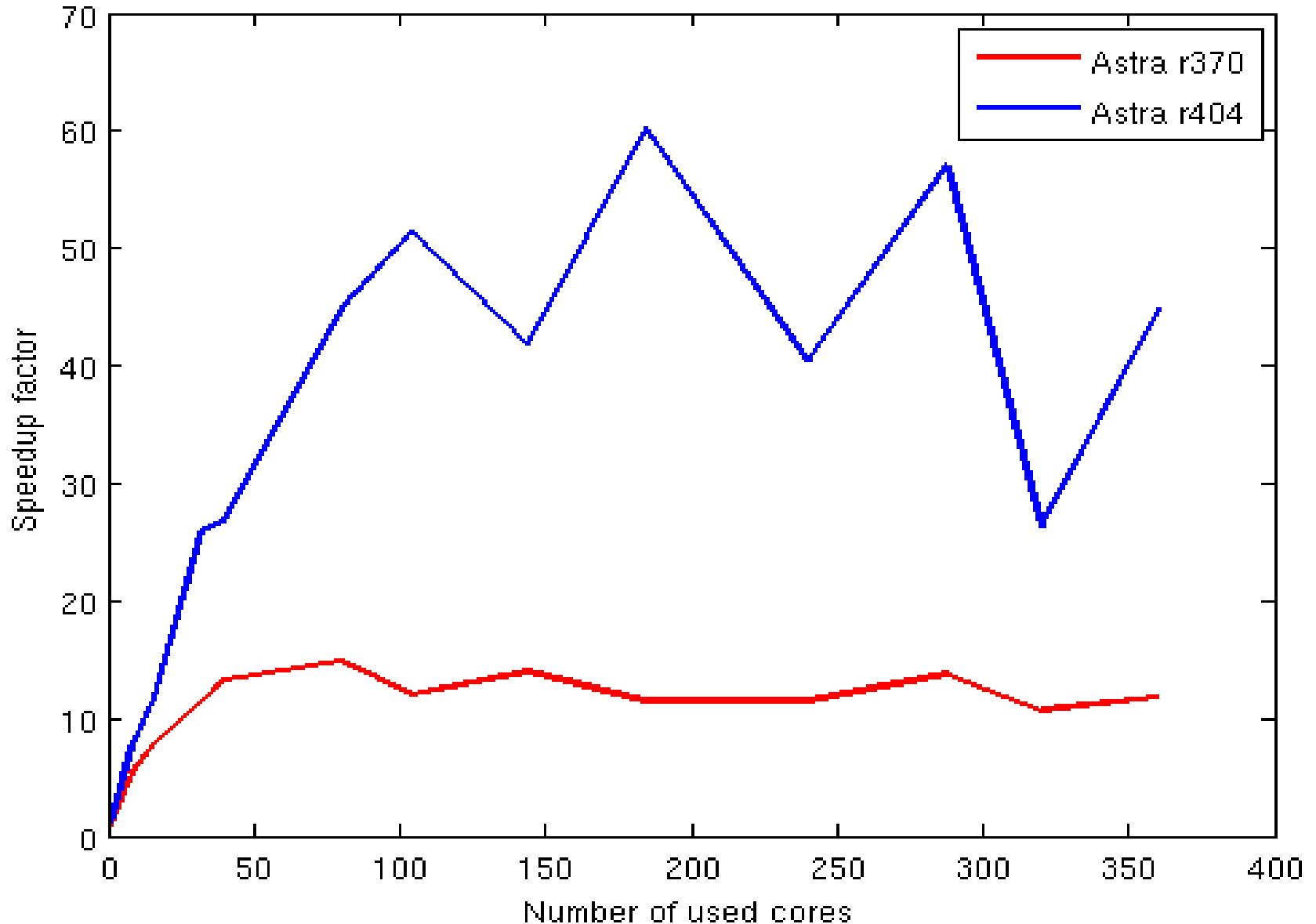
Test with parallel Astra revision 370 and revision 404 on ferrari3
cluster with 48*8 cores

Run time serial astra; test1: 4:40h, test 7: 3:20h

Speedup plot (PIZ 1.6 test1, 200k particles, 0m - 10m, cylindrical sc algorithm)

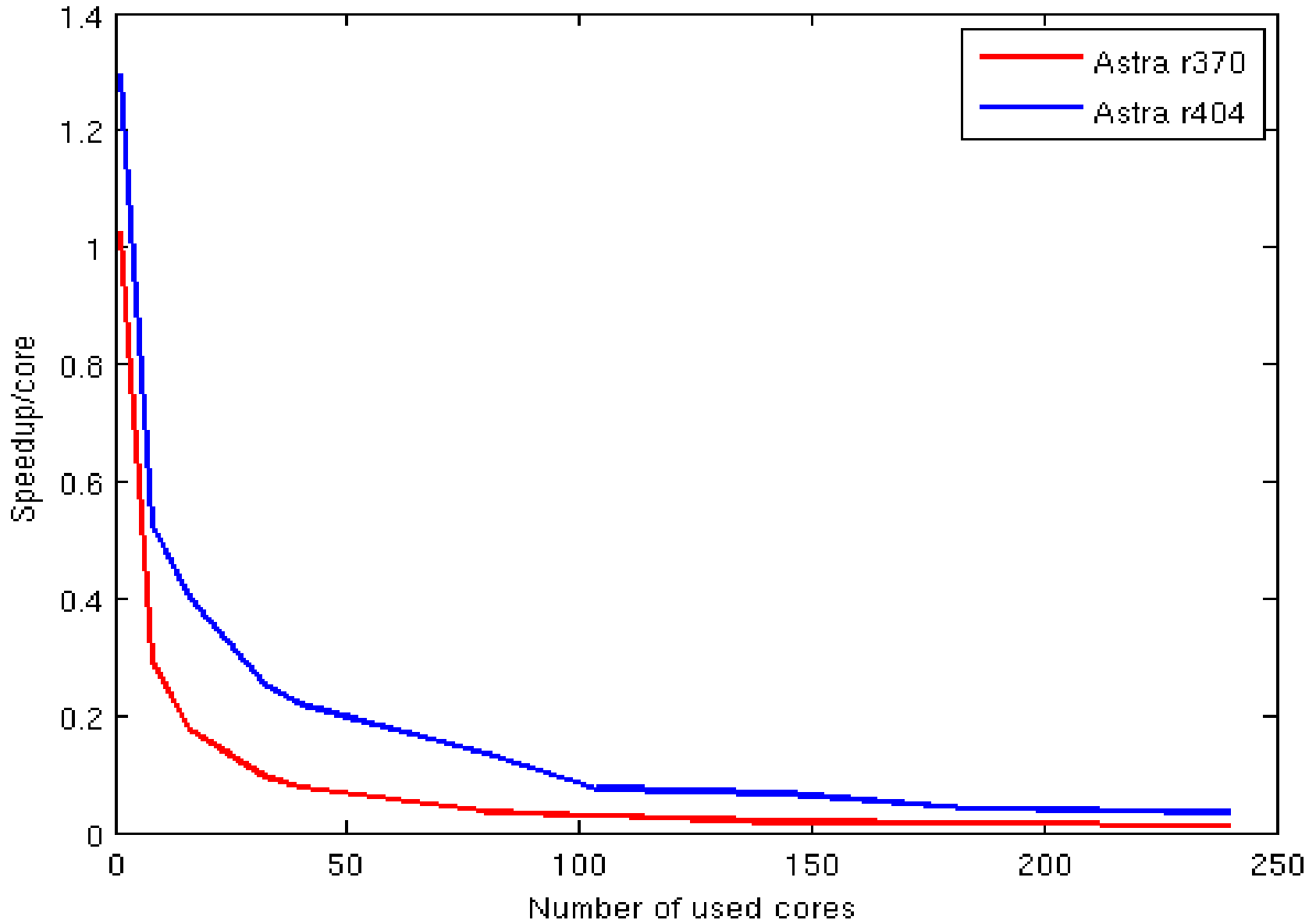


Speedup plot (PIZ 1.6 test7, 200k particles, 3m - 10m, cylindrical sc algorithm)

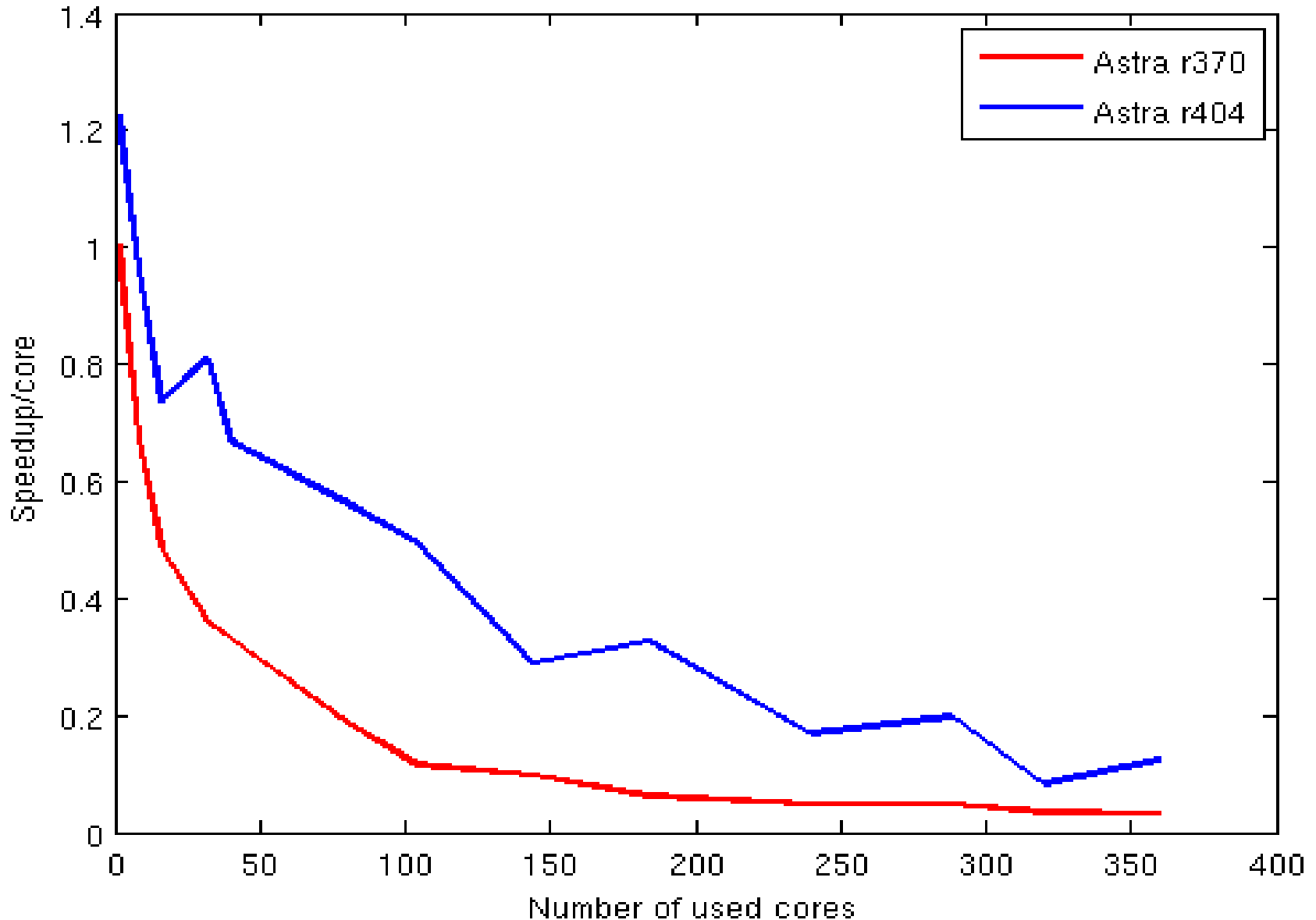


Speedup / core with emission

Speedup/core (PIZ 1.6 test1 , 200k particles, 0m - 10m, cylindrical sc algorithm)



Speedup/core (PIZ 1.6 test7, 200k particles, 3m - 10m, cylindrical sc algorithm)



- continue with parallelization of 3d space charge solver code
- modify the emittance calculation code to reduce particle synchronization
- speed up is-in-field algorithm

- Parallel Astra is still alive and in use
- More parallelization work in progress
 - Bug reports are welcome

Visita Parallel Astra:

<http://tesla.desy.de/~meykopff/>

Thank you for your attention