Scalability study of particle method with dynamic load balancing

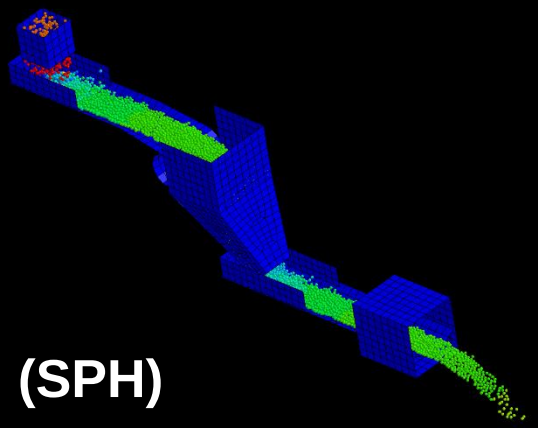
Hailong Teng

tenghl@lstc.com

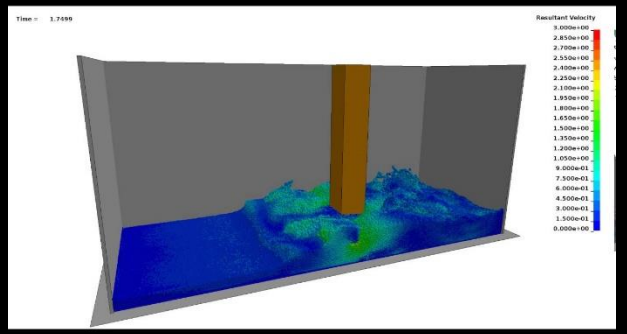
Livermore Software Technology Corporation

Deutsches LS-DYNA Forum
October 2018, Bamberg

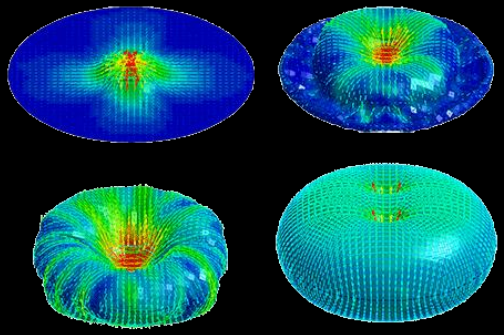
Motivation: Particle Method in LS-DYNA



- Discrete Element Method(DEM)
- Smoothed particle hydrodynamics (SPH)



- **Airbag Particle Method (CPM) & Particle Blast Method(PBM)**
- SPG,EFG….



Motivation: Benchmark study

Benchmark data

I'm using a following paper by Hailong and Jason as reference.
"Particle Blast Method for the Simulation of Blast Loading"
13th International LS-DYNA Users Conference

- Stand-off distance = 700mm
- TNT 15kg
- Case1:NPHE=2e5, NPAIR=3.8e6, IUNIT=1
  Case2:NPHE=2e6, NPAIR=2e6, IUNIT=1
  Total particle number = 4.0e+6
- Structures : Circular plate with clamped
  radius=0.5m, thickness=20mm
  SHELL elform=16, NIP=5
- MAT_PLASTIC_KINEMATIC
  RO=7838[kg/m^3], E=2.12e11[Pa],
  OR=0.28, SIGY=1.2e9[Pa],
  ETAN=6.5e9[Pa], SRC=300, SRP=5.0

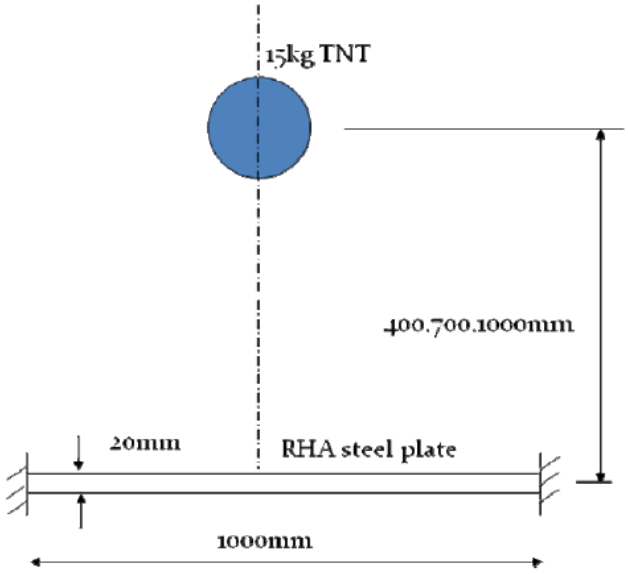
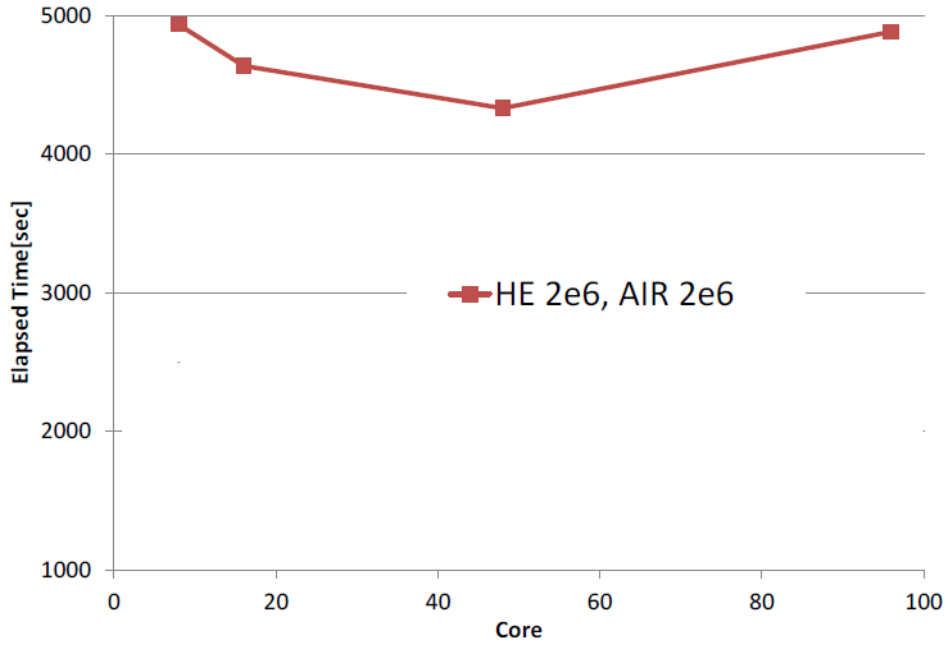


Fig.2 Test set-up

株式会社JSOL

Motivation: Benchmark study

Parallel Performance

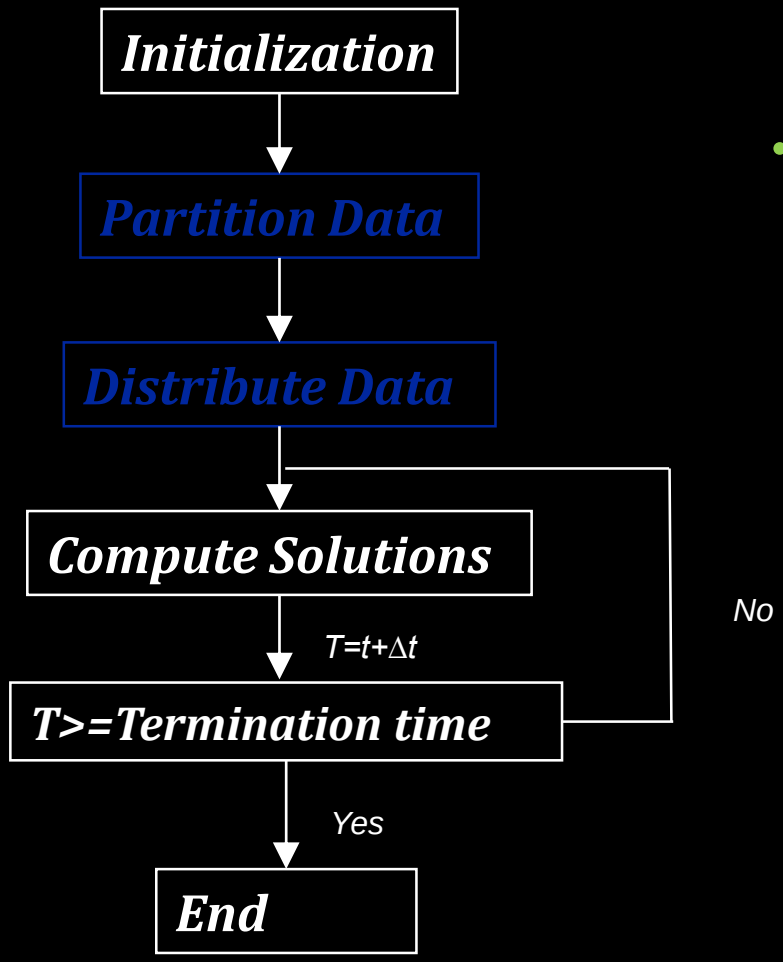


- From 8 cores to 48 cores, ~14% CPU time reduction
- Scale down when using more than 48 cores!

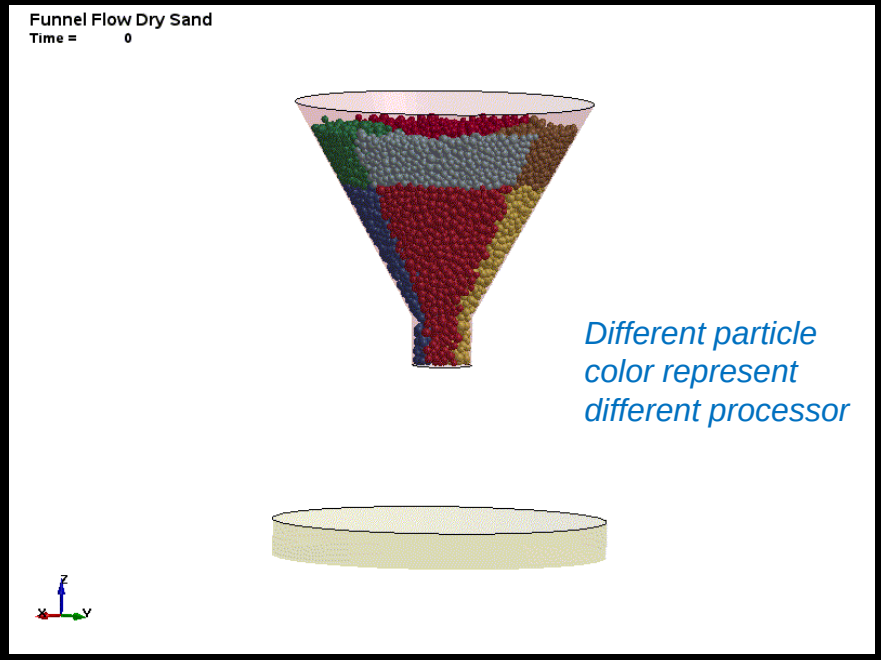
株式会社JSOL

Static Partitioning:

Partition is performed once and used throughout simulation



- *For Particle method, load imbalance continue to be a significant bottleneck in the simulation*
- *The MPP job might get good scalability at initial stage with perfect decomposition, later when particles undergo large motion, the scalability begin to deteriorate*

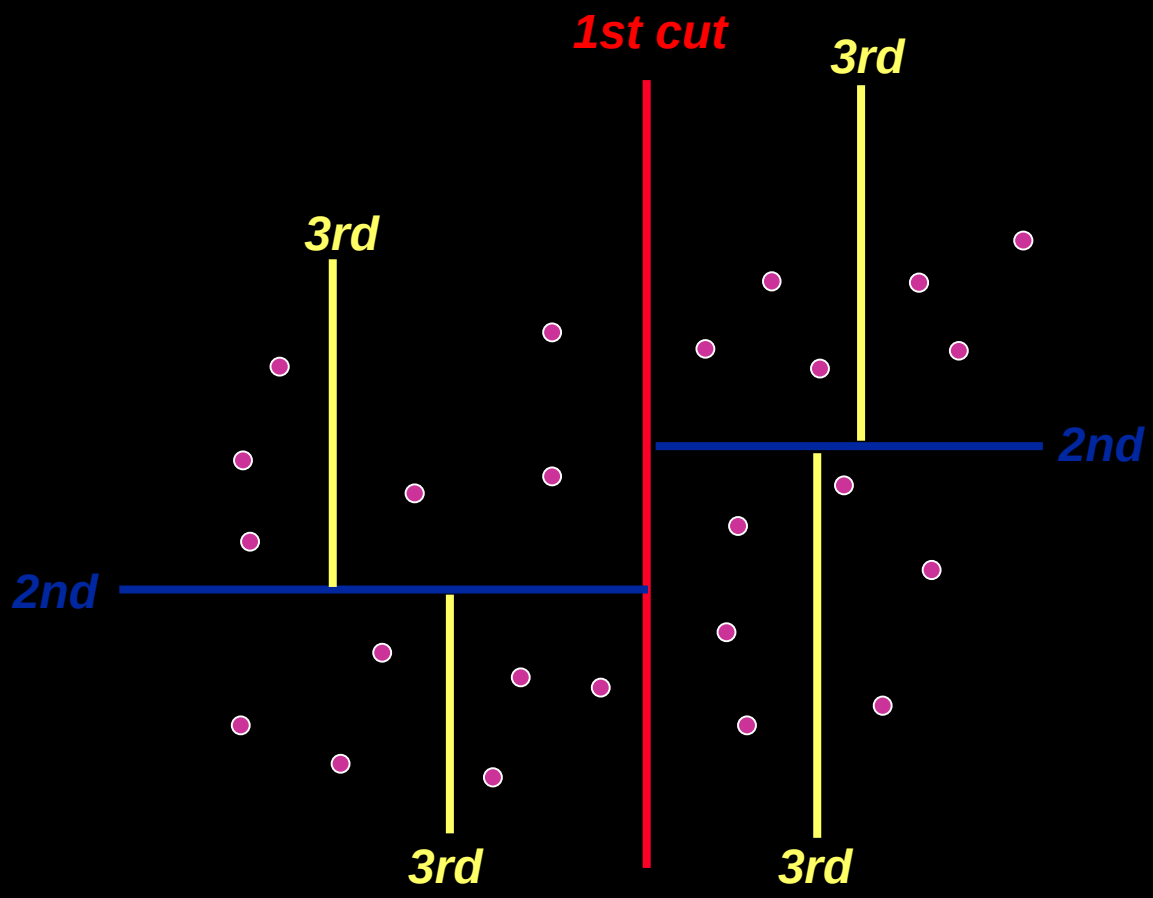


Motivation of Dynamic Rebalancing

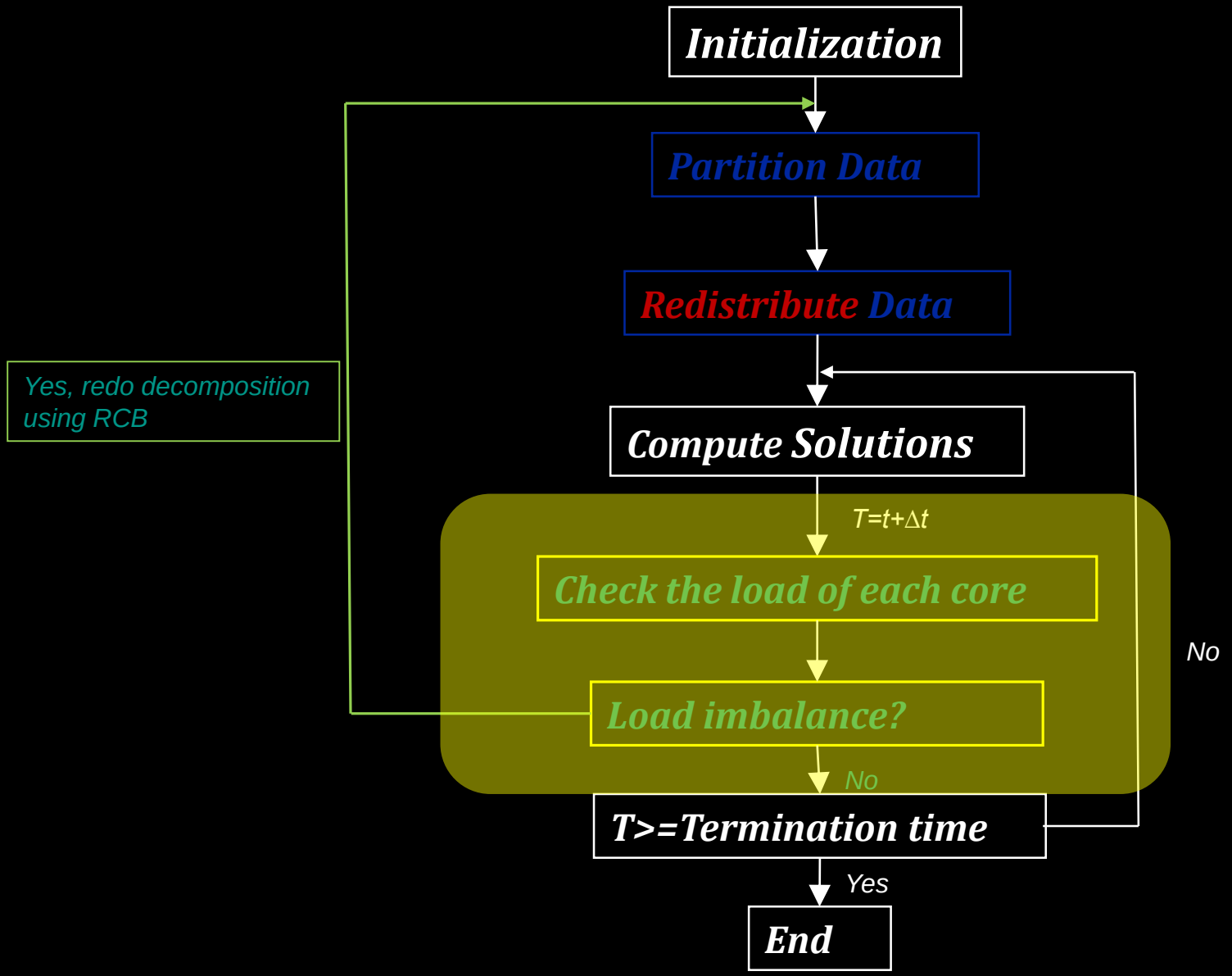
- Evenly distribution of cost across all nodes in order to optimize
- So far, division of a problem into a fixed number of processes
- Issues:
  - amount of work is often not known prior to execution
  - load situation changes dynamically
- objective: load distribution or load balancing strategies

RCB Decomposition

- **Recursive Coordinate Bisection: recursively divide subdomain into two equal parts using a cutting plane orthogonal to a coordinate axis.**



Dynamically Rebalancing: Partition is performed dynamically to balance workload and keep communication cost low



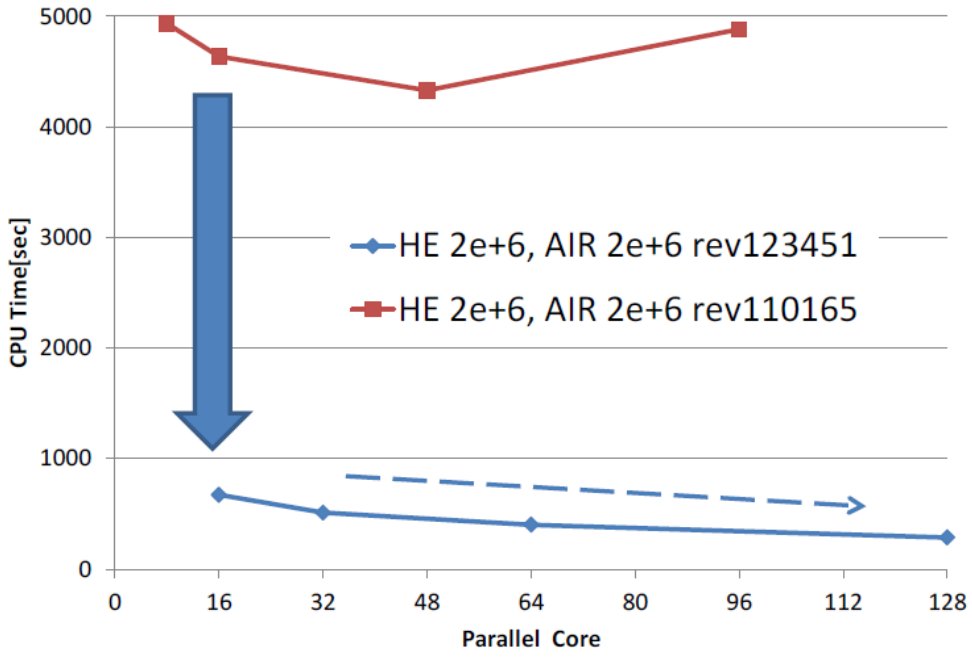
Dynamically Rebalancing: DEM example

Dynamically Rebalancing: CPM example

- *Particle motion requires migration of particles between processors and can thus cause an imbalance of the number of particles assigned to the nodes*
- *Our method utilizes flexible RCB decomposition, which dynamically adjusting the particle sub-domain boundaries to guarantee the particles are uniformly distributed to each processor.*

Benchmark study with dynamic rebalancing

Parallel Performance



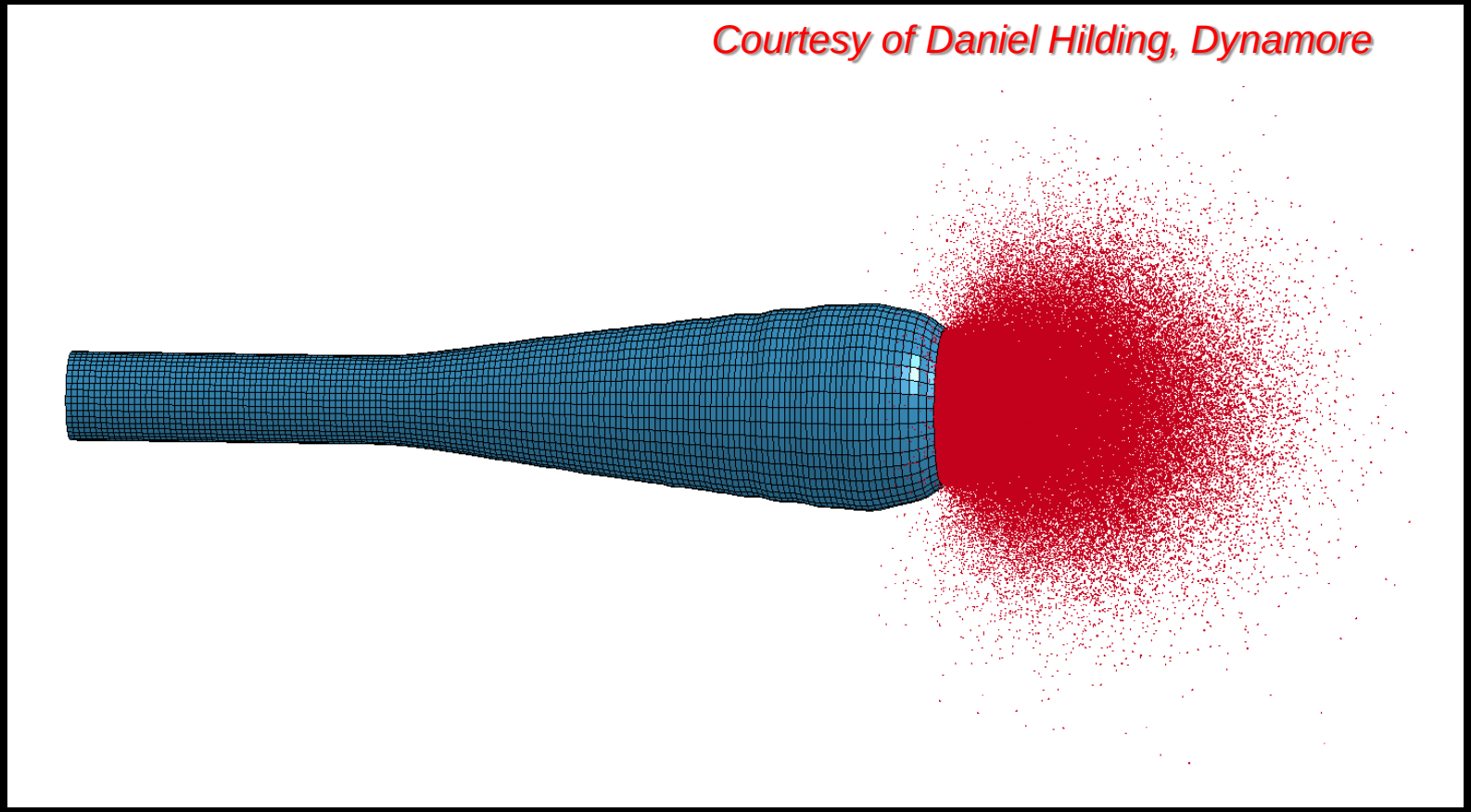
Core	rev110165	Speed Up
8	4933 sec	-
16	4636 sec	0.940
48	4330 sec	0.878
96	4882 sec	0.990

Core	rev123451	Speed Up
16	672 sec	-
32	511 sec	0.760
64	402 sec	0.598
128	286 sec	0.426

- ~17 times speedup when using ~100 cores
- Still scale up to 128 cores

株式会社JSOL

C4 Cylinder Test



Number of HE particles: 5,000,000

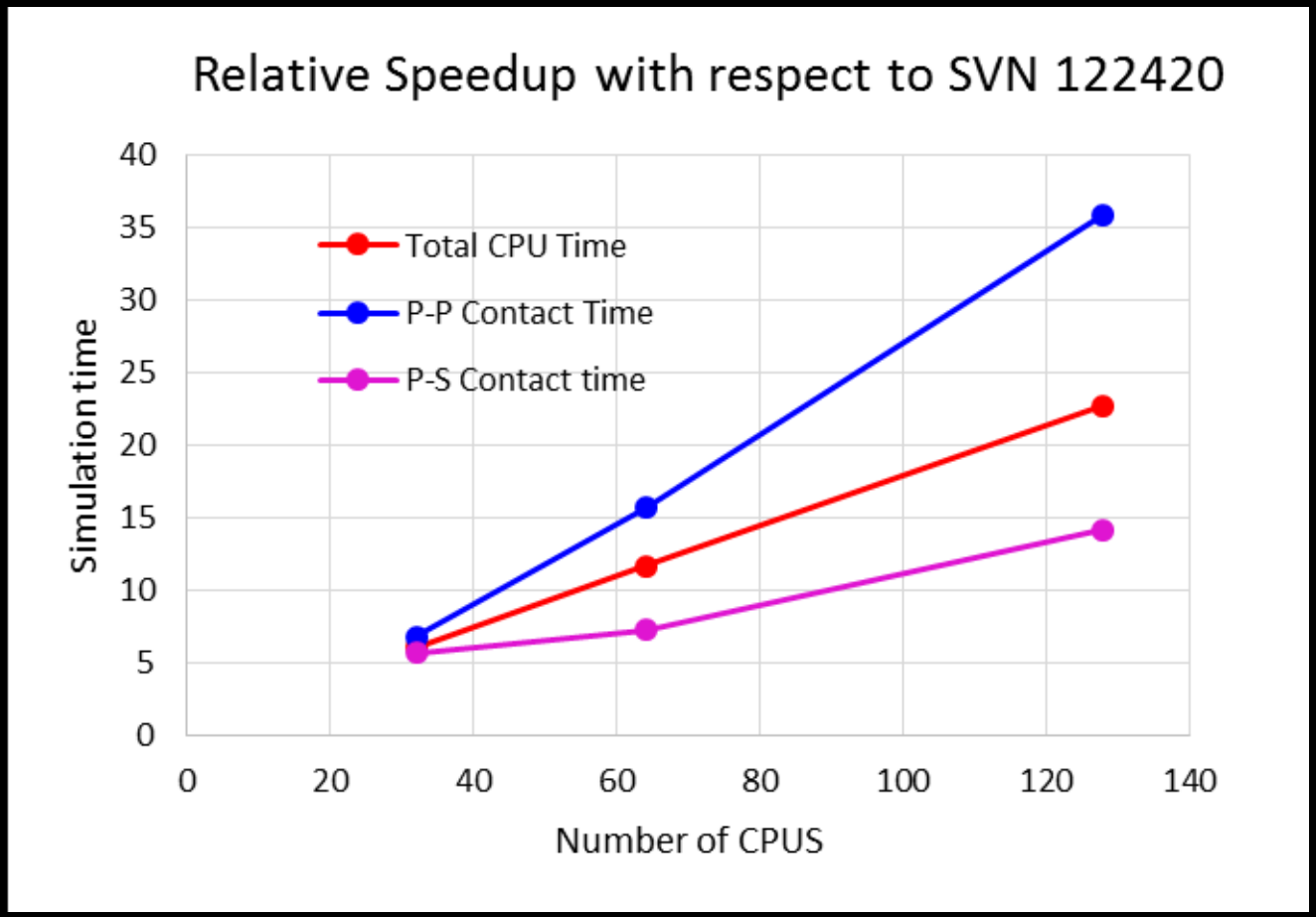
C4 Cylinder Test

#of CPUs	Total CPU time(s)		
	SVN 122420	RB	Speedup
32	10098	1665	6.06
64	10058	862	11.67
128	9690	427	22.7

#of CPUs	Particle to Particle Contact time(s)		
	SVN 122420	RB	Speedup
32	8189	1204	6.80
64	8380	533.5	15.7
128	8067	225	35.9

#of CPUs	Particle to Structure Coupling time(s)		
	SVN 122420	RB	Speedup
32	1380	242.8	5.68
64	1197	164.7	7.3
128	1212	85.2	14.2

C4 Cylinder Test



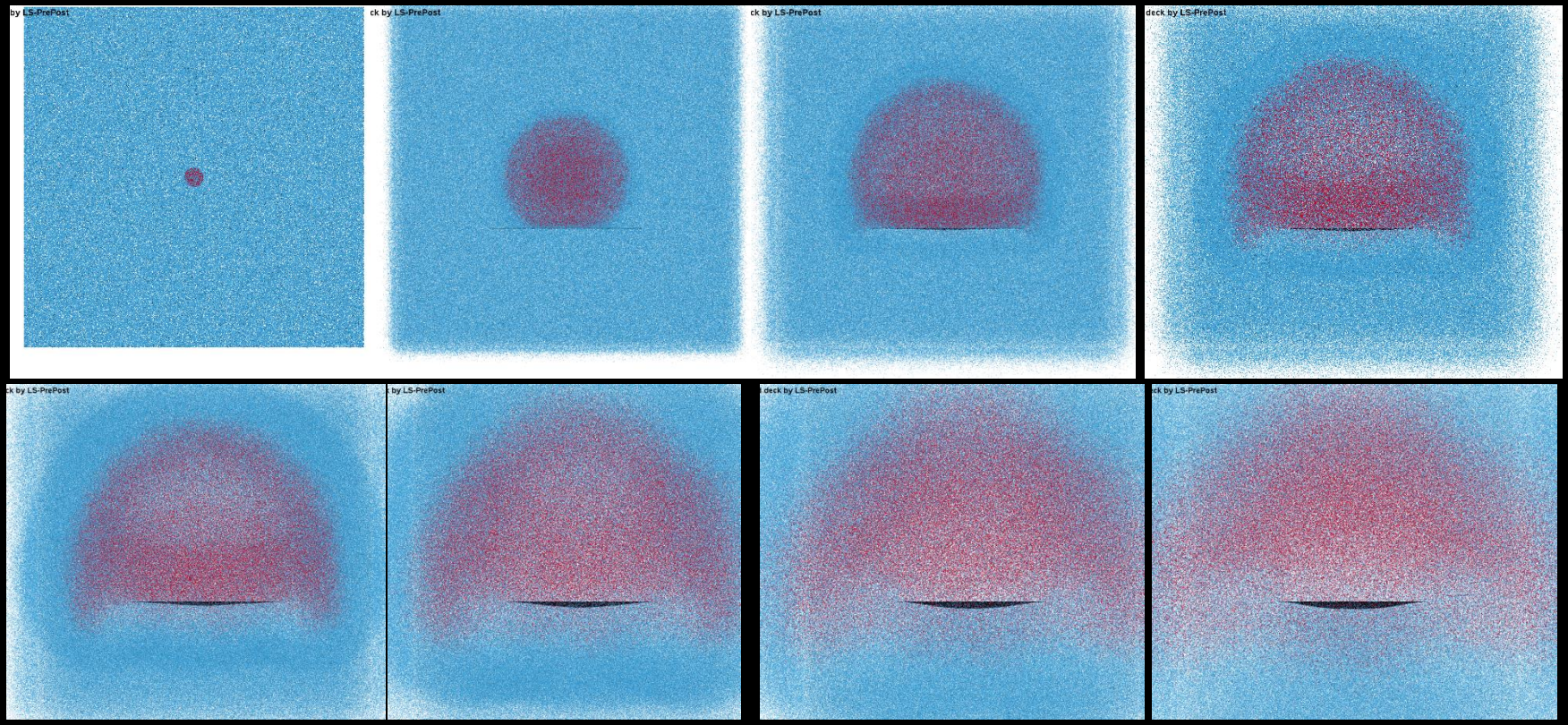
Air blast with spherical shaped charge

Number of HE particles: 0.2M

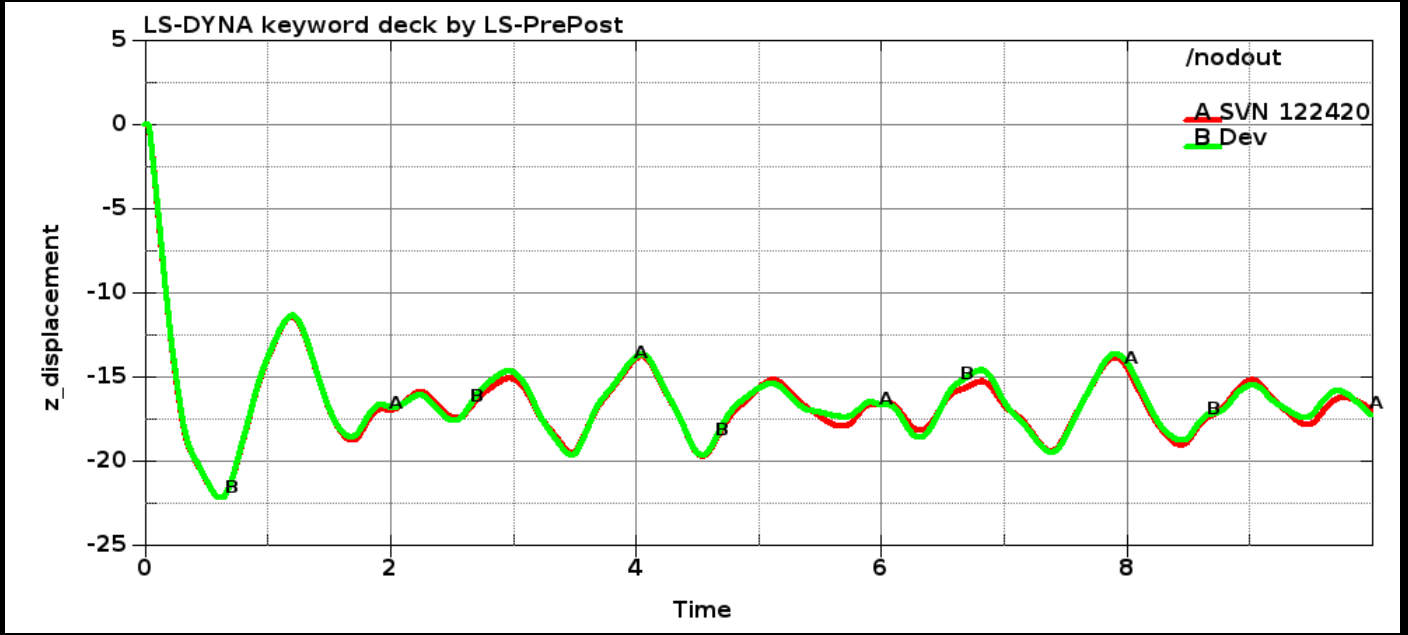
Number of Air particles: 1.7M

Courtesy of Daniel Williams, Supacat

Christian Mæhle Kaurin et. at: Blast loading on square steel plates; A comparative study of numerical methods, 2010



Air blast with spherical shaped charge



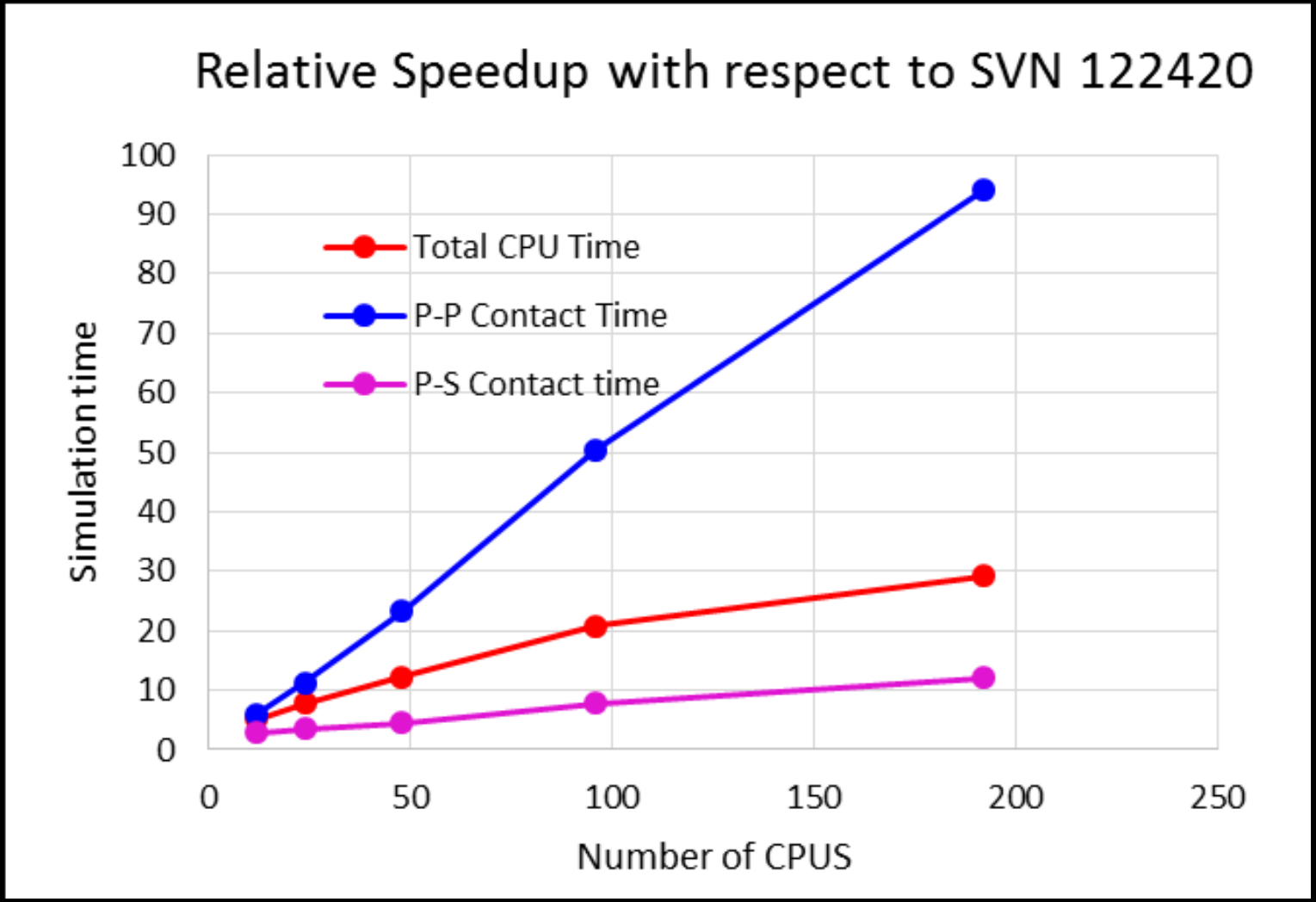
Air blast with spherical shaped charge

#of CPUs	Total CPU time(s)		
	SVN 122420	Dev	Speedup
12	25190(7h0m)	4916(1h22m)	5.12
24	24681(6h51m)	3188(53m)	7.77
48	26540(7h22m)	2168(36m)	12.24
96	30719(8h32m)	1482(25m)	20.73
192	35627(9h54m)	1217(20m)	29.27

#of CPUs	Particle to Particle Contact time(s)		
	SVN 122420	Dev	Speedup
12	15506	2596	5.97
24	15809	1411.2	11.20
48	17183	739.7	23.23
96	19912	394.9	50.42
192	22468	239.1	93.97

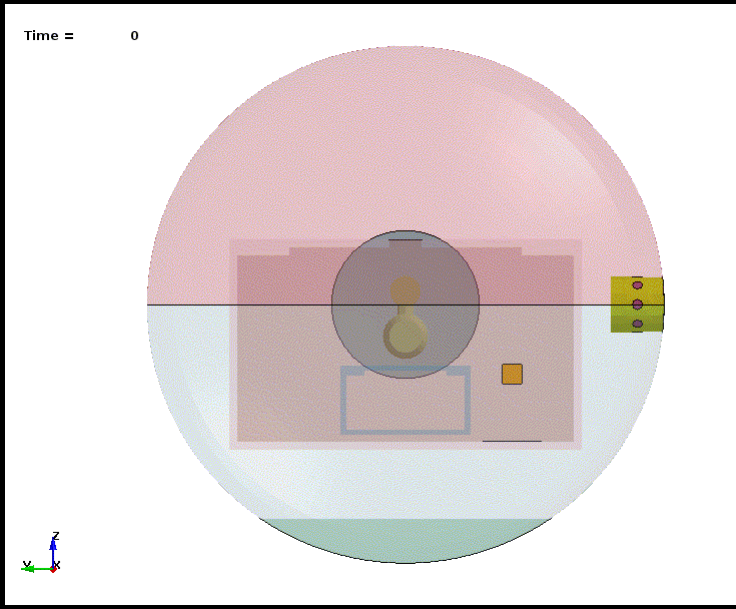
#of CPUs	Particle to Structure Coupling time(s)		
	SVN 122420	Dev	Speedup
12	4982.6	1714.1	2.91
24	4742.3	1350.7	3.51
48	5118.1	1127.5	4.54
96	6585.0	849.2	7.75
192	8863.7	731.9	12.11

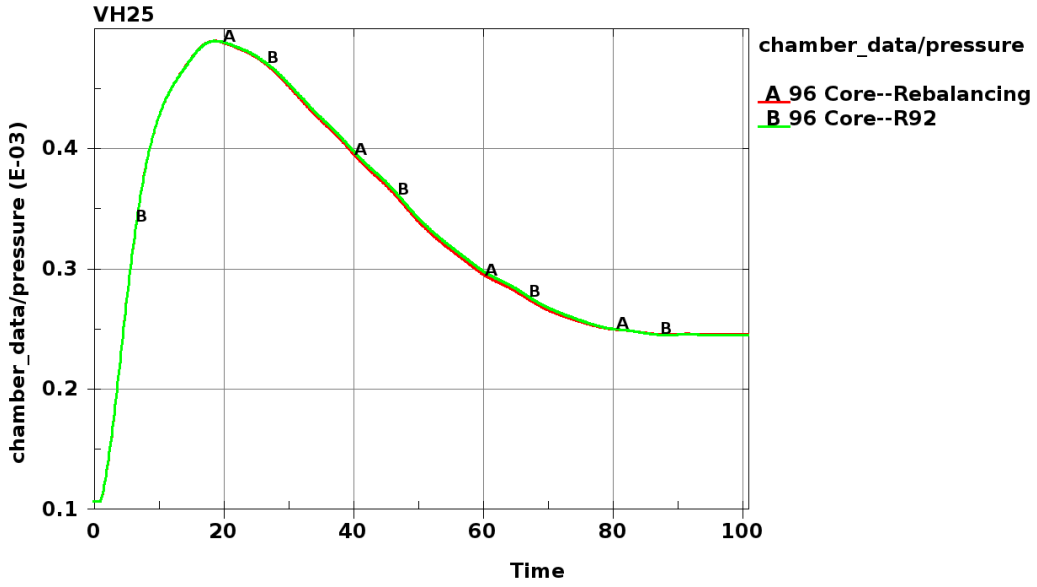
Air blast with spherical shaped charge



- *Numerical study of CPM problems*
  - *Tank Test*
  - *Curtain airbag (CAB)*

Case 1: Tank Test (from Mazda)



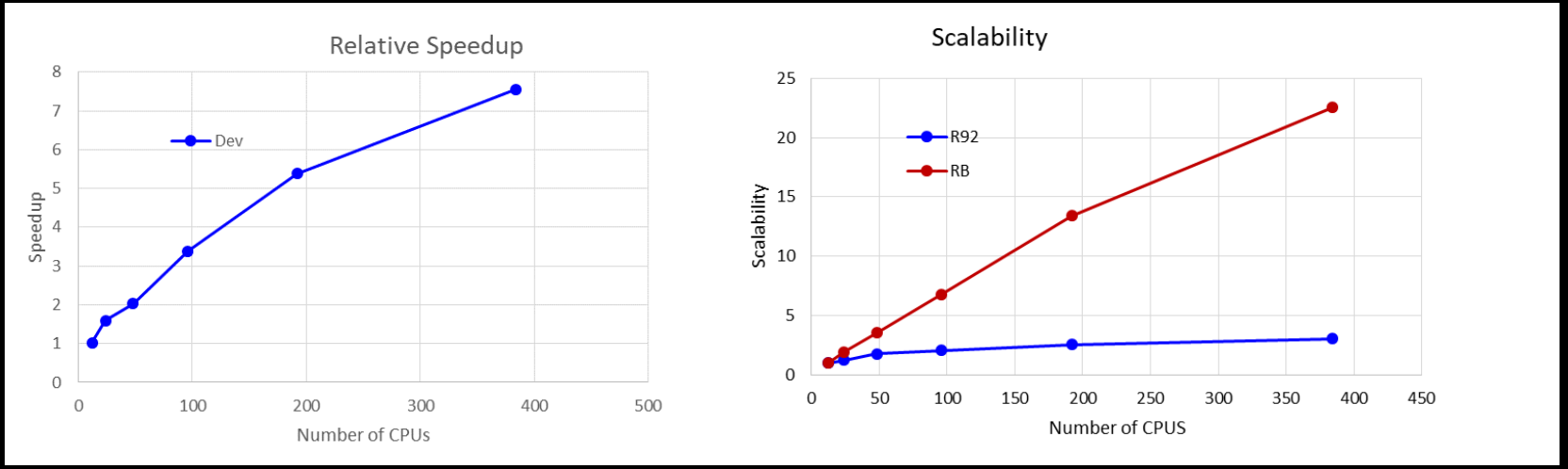


- *Number of Air particle: 94,400， number of inflated Particle: 528,000.*
- *Rigid structure*
- *Particle distribution is quite nonuniform*
- *Same results as Released version*

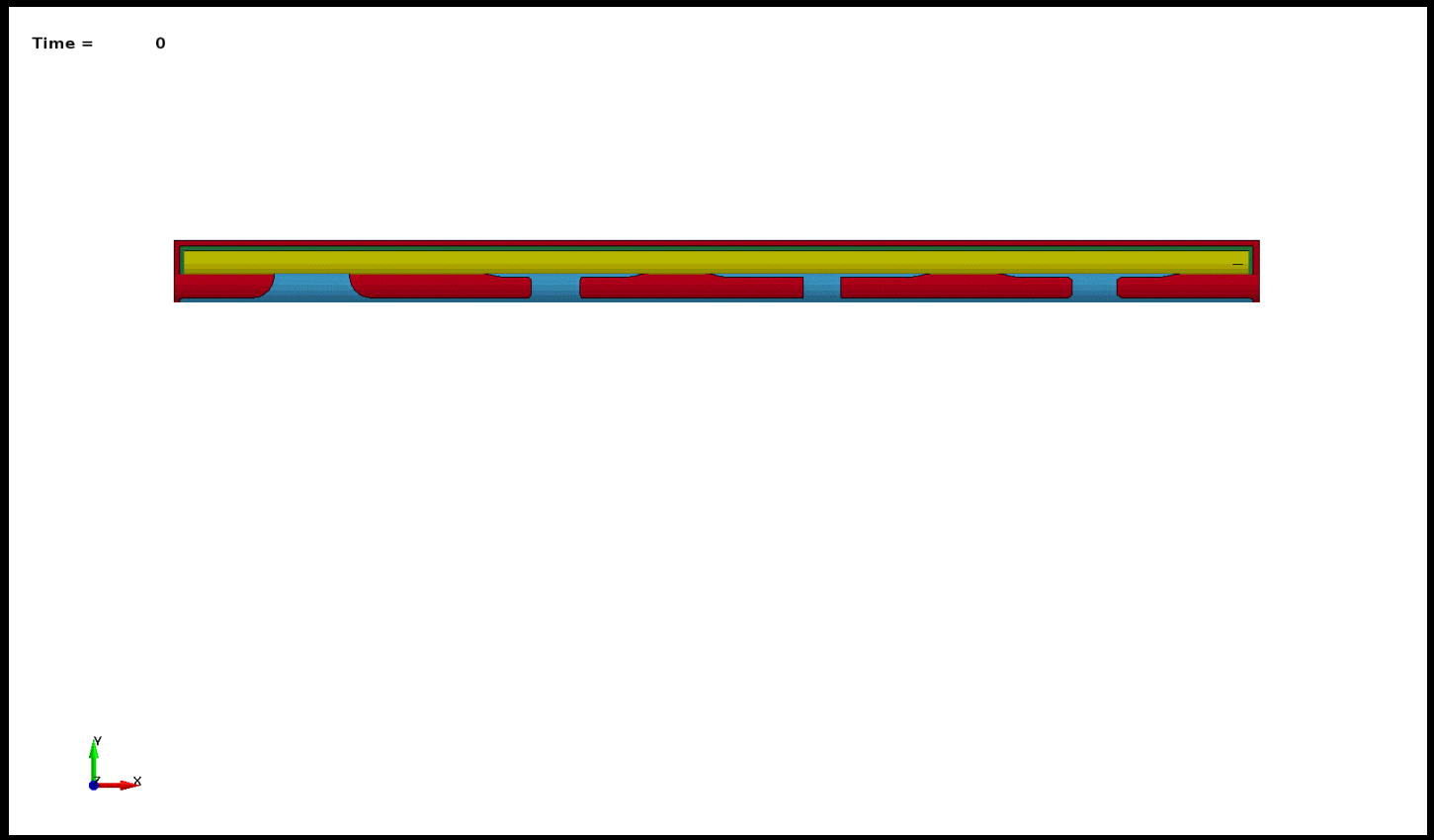
Case 1: Tank Test

#of cores	Total CPU time(s)		
	R92	RB	Speedup
12	61060 (17hrs)	59850	1.02
24	48891	30759	1.59
48	34214	16869	2.03
96	29861	8845	3.38
192	24031	4467	5.38
384	20052(5.5hrs)	2656 (44min)	7.55

#of cores	Scalability	
	R92	RB
12	1.	1
24	1.2489	1.9458
48	1.7846	3.5479
96	2.0448	6.7665
192	2.5409	13.398
384	3.0451	22.534

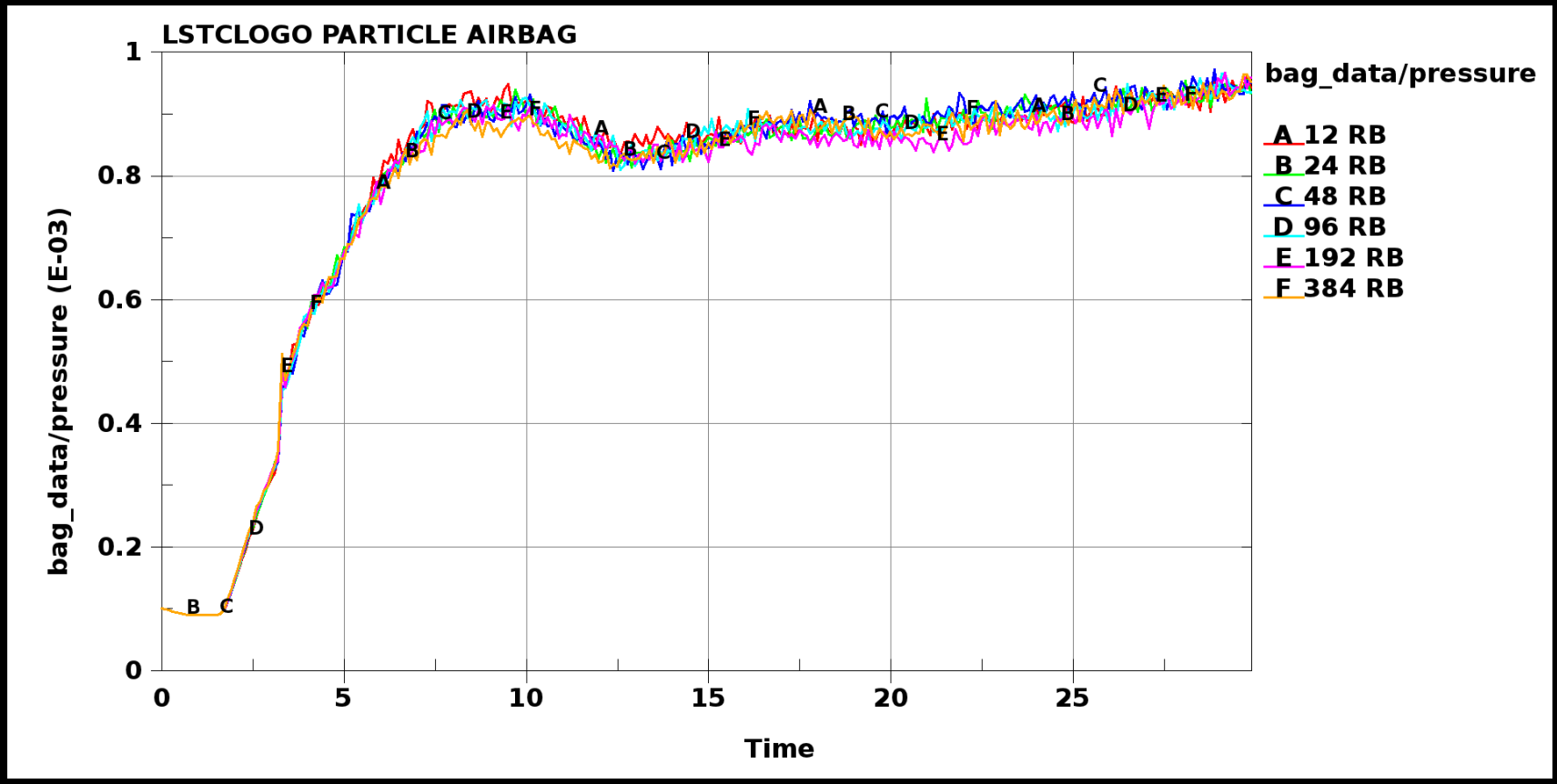


Case 2: Curtain Airbag---LSTC Logo



Number of particle: 1,000,000

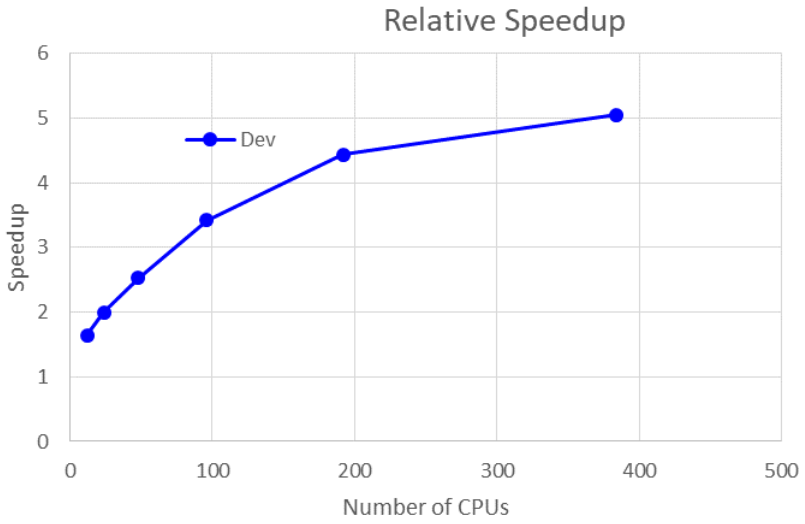
Case 2: Curtain Airbag--LSTC Logo

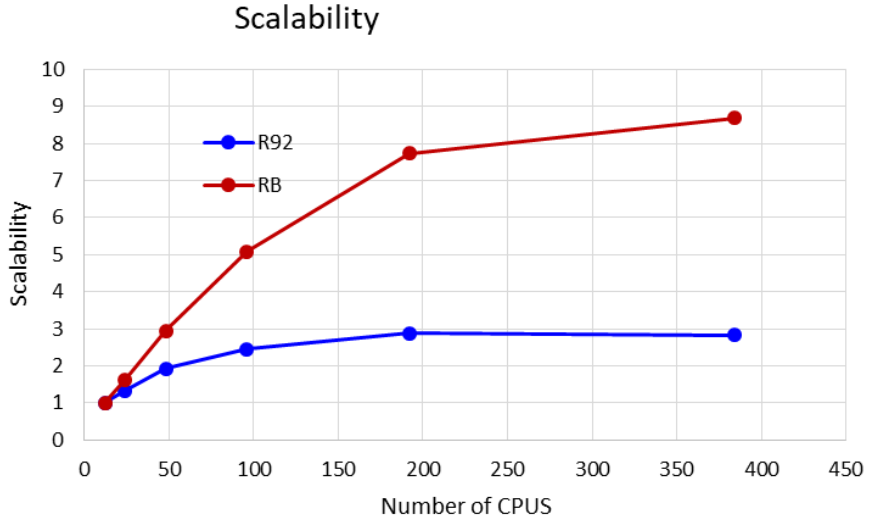


Case 2: Curtain Airbag--LSTC Logo

#of cores	Total CPU time(s)		
	R92	RB	Speedup
12	9281	5632	1.64
24	6967	3476	2.00
48	4841	1909	2.53
96	3786	1107	3.42
192	3232	728	4.44
384	3274	632	5.18

#of cores	Scalability	
	R92	RB
12	1.	1
24	1.3321	1.6203
48	1.9172	2.9502
96	2.4514	5.0876
192	2.8716	7.7363
384	2.8348	8.6914





Conclusion

- *With dynamic rebalancing, excellent scalability is obtained for CPM & PBM:*
  - *For PBM, >20 speedup is obtained for with ~100 CPUs*
  - *>4~5 times speedup is obtained for CAB*
  - *~2~3 times speedup for DAB and PAB*

- *The same idea of rebalancing can be applied to other mesh free method*
  - *DEM (in progress…)*
  - *SPH*
  - *SPG…*
- *The current approach is limited to particles only*
  - *Particle-structure interaction..*