What is HYBRID and Why?

Jason Wang (LSTC)

1/24/2018

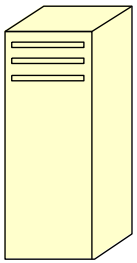
http://ftp.lstc.com/anonymous/outgoing/jason/Slides

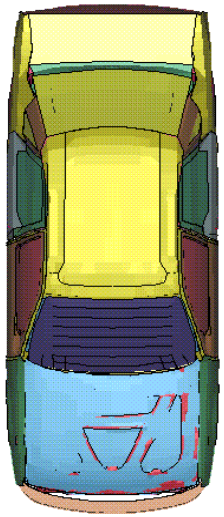
Outline

- Introduction
- Numerical variations
- Evolution of crashworthness modelling
- Evolution of CPU technology
- SMP and MPP
- Summary

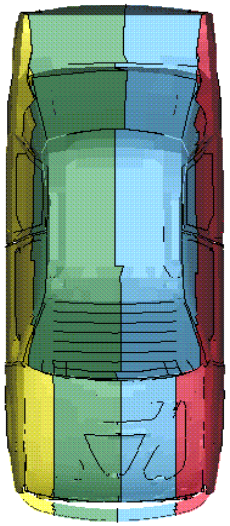
Introduction - SMP vs MPP

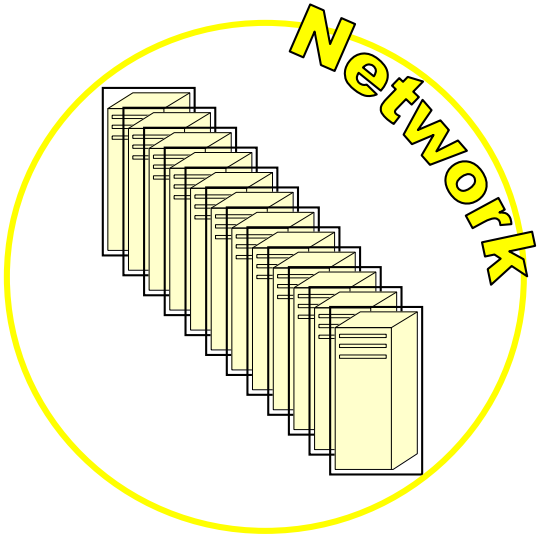
Shared memory





Clusters





SMP (Shared Memory Parallel)

- Start and base from serial code
- Using OpenMP directives to split the tasks

  (fine grained parallelism)
- *Consistent results with different cores*
- Only run on SMP (single image) computers
- Scalable up to ~16 CPUs

$T_{elapse}=T_{cpu}+T_{sys}+T_{omp}$

MPP (Message Passing Parallel)

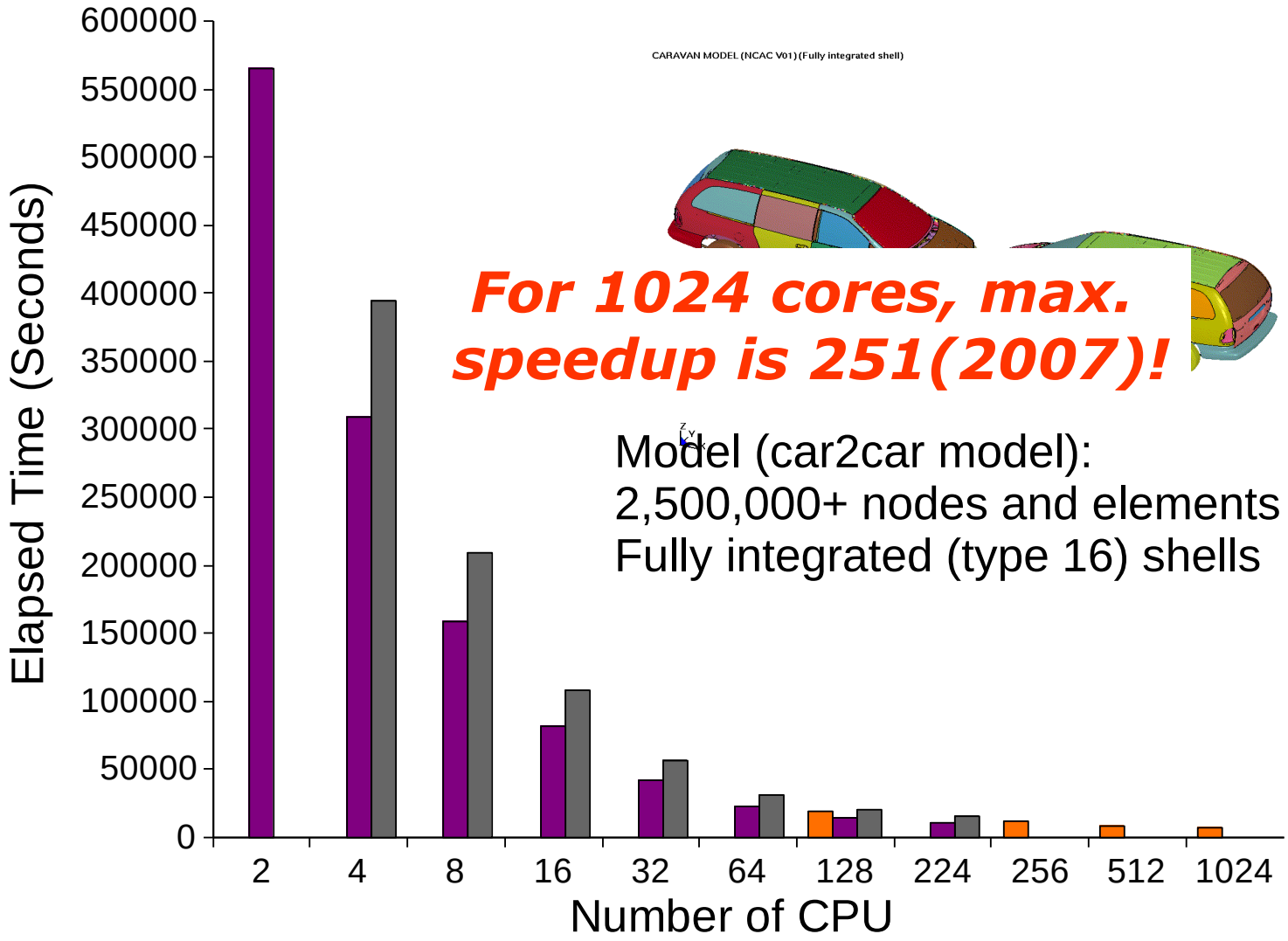
- Using domain decomposition
- Using MPI for communications between sub-domains

  (coarse grained parallelism)
- *Results change with different cores*
- Work on both SMP machines and clusters
- Scalable >> 16 CPUs

$T_{elapse}=T_{cpu}+T_{sys}+T_{mpp}$

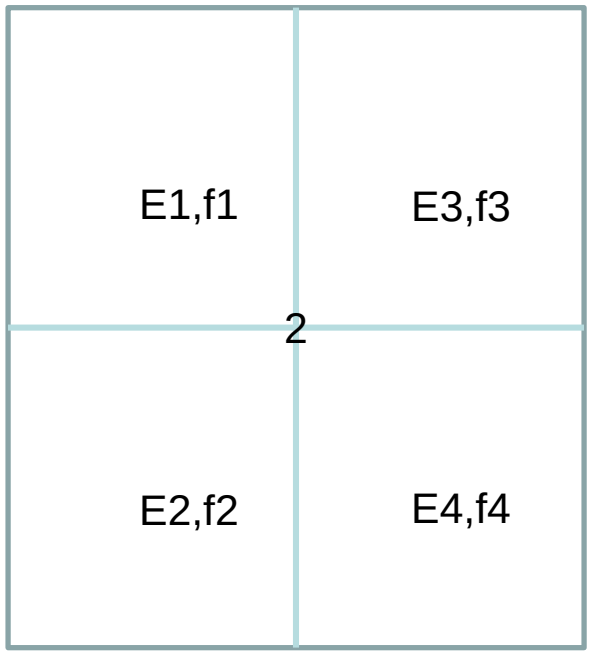
LSTC
Livermore Software Technology Corp.

MPP performance



Technology Corp.

Numerical Variations - SMP



MPP domain 0

4 elements using 4 SMP threads

SMP ncpu=4, no special treatment

Based on the system load with 4 different runs

Force at node 2 h2= ((f1 + f2) + f3) +f4

Force at node 2 h2= ((f3 + f1) + f2) +f4

Force at node 2 *h2*= ((f4 + f1) + f3) +f4

Force at node 2 **h2**= ((f4 + f2) + f3) +f1

h2 != h2 != *h2* ! = **h2**

It may create 4 different results!

SMP ncpu=-4, special treatment

Force at node 2 from e1 g1= f1

Force at node 2 from e2 g2= f2

Force at node 2 from e3 g3= f3

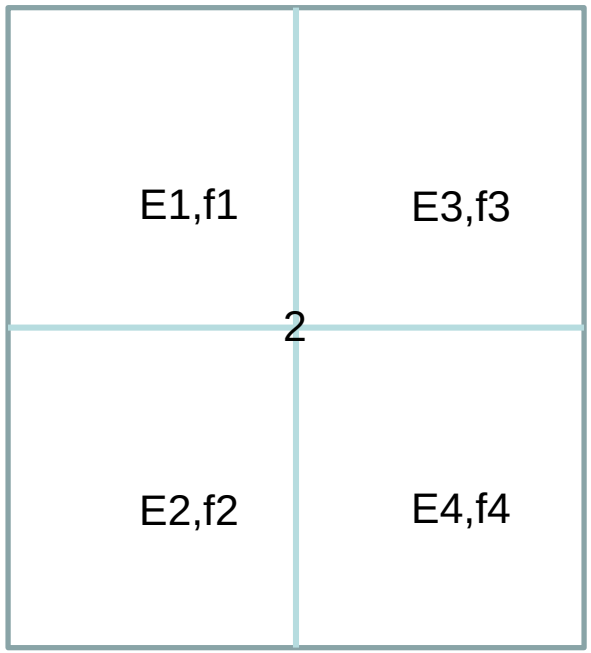
Force at node 2 from e4 g4= f4

Force assembly is outside of element loop:
Force at node 2 h2= ((g1+g2)+g3)+g4

Consistent results but ~10% slow down

LSTC
Livermore Software Technology Corp.

Numerical Variations - SMP



MPP domain 0

2/4 SMP threads

SMP ncpu=-2

Thread 1: Force at node 2 from e1 g1= f1

Thread 2: Force at node 2 from e2 g2= f2

Thread 1: Force at node 2 from e4 g4= f4

Thread 2: Force at node 2 from e3 g3= f3

SMP ncpu=-4

Thread 1: Force at node 2 from e1 g1= f1

Thread 2: Force at node 2 from e2 g2= f2

Thread 3: Force at node 2 from e4 g3= f3

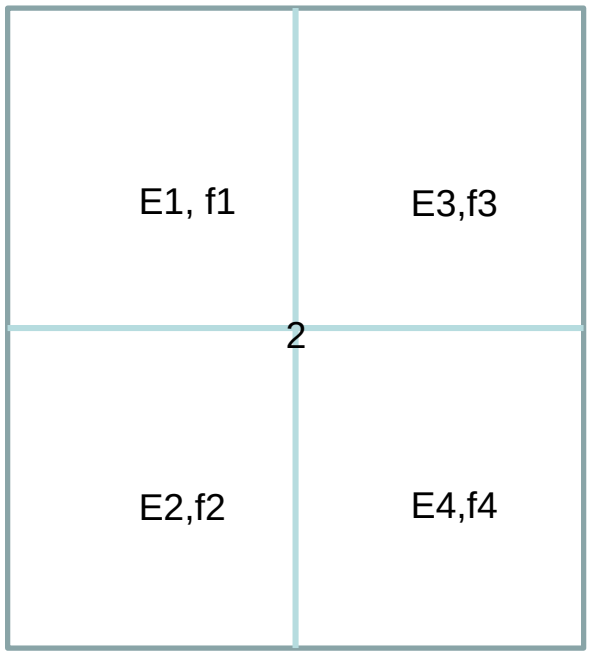
Thread 4: Force at node 2 from e3 g4= f4

Force assembly is outside of element loop
Summing up nodal force in a specific order:

Force at node 2 h2= ((g1+g2)+g3)+g4

The results from ncpu=-2 and ncpu=-4 are identical.

Numerical Variations - MPP



MPP domain 0

4 elements in one MPP domain

Force at node 2 h2=((f1+f2)+f3)+f4

h2=((10.1+4.01)+5.05)+3.06)
=(14.1+5.05)+3.06
=19.1+3.06
=22.1

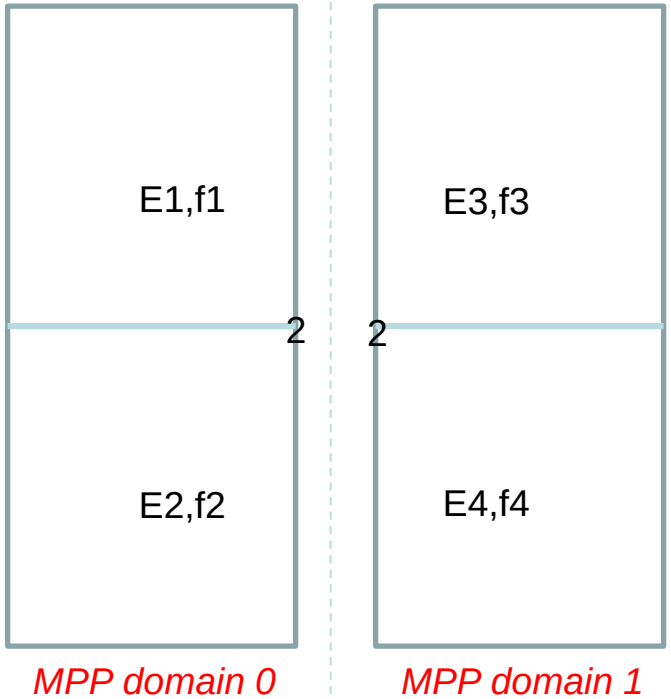
Change MPP ranks

Example with 3 significant digits:
f1=10.1
f2=4.01
f3=5.05
f4=3.06

h2 ! = h2

22.1 != 22.2

New domain line



MPP domain 0 *MPP domain 1*

Processor 0 node 2 p0=f1+f2
Processor 1 node 2 p1=f3+f4
Force at node 2 h2 = p0 + p1

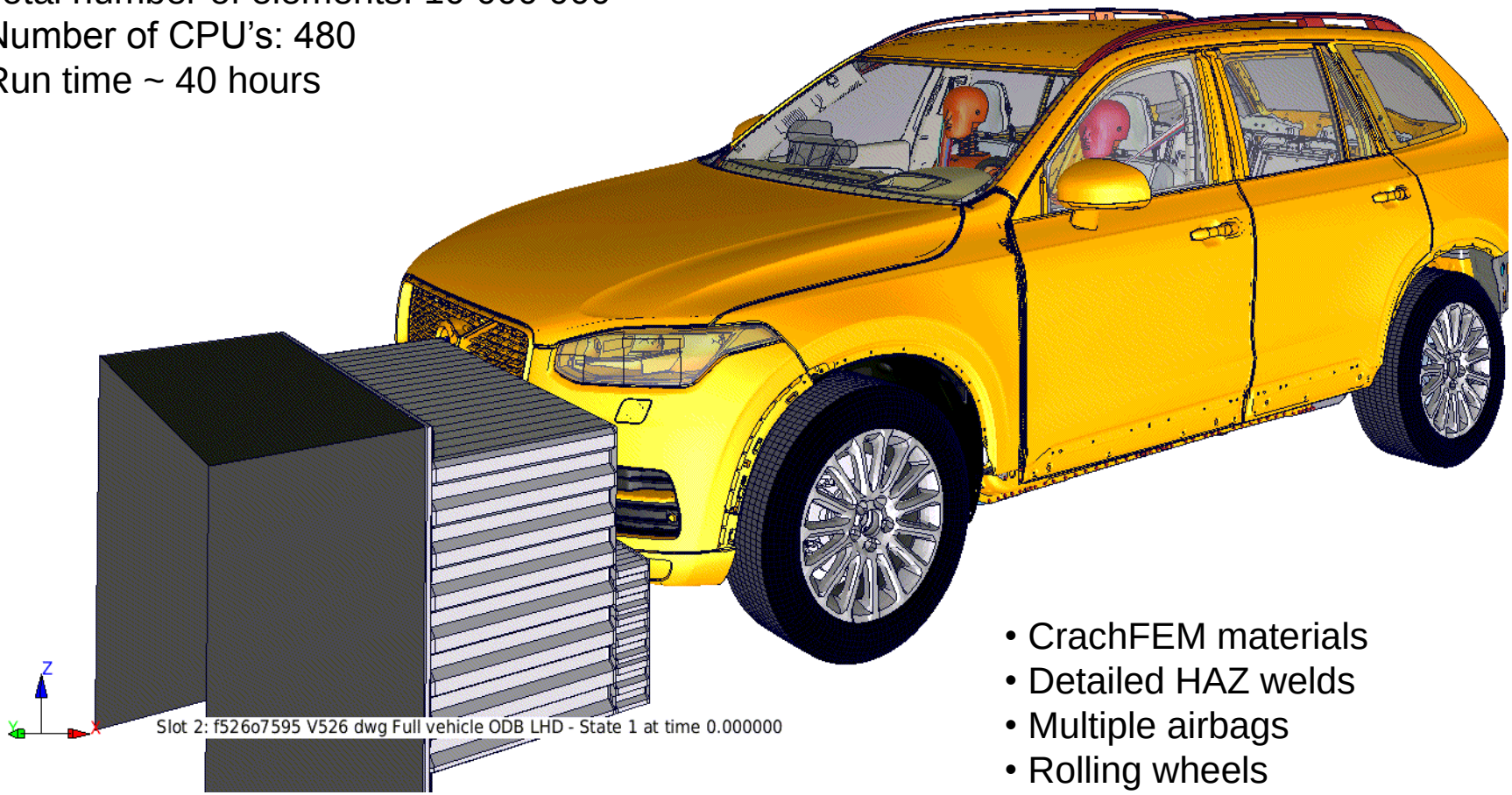
p0= 10.1+4.01= 14.1
p1= 5.05+3.06=8.11
h2 =14.1+8.11=22.2

MPP consistency? Very expansive!

LSTC Livermore Software Technology Corp.

VOLVO XC90 GEN II Crash CAE model 2014

- Total number of elements: 10 000 000+
- Number of CPU's: 480
- Run time ~ 40 hours

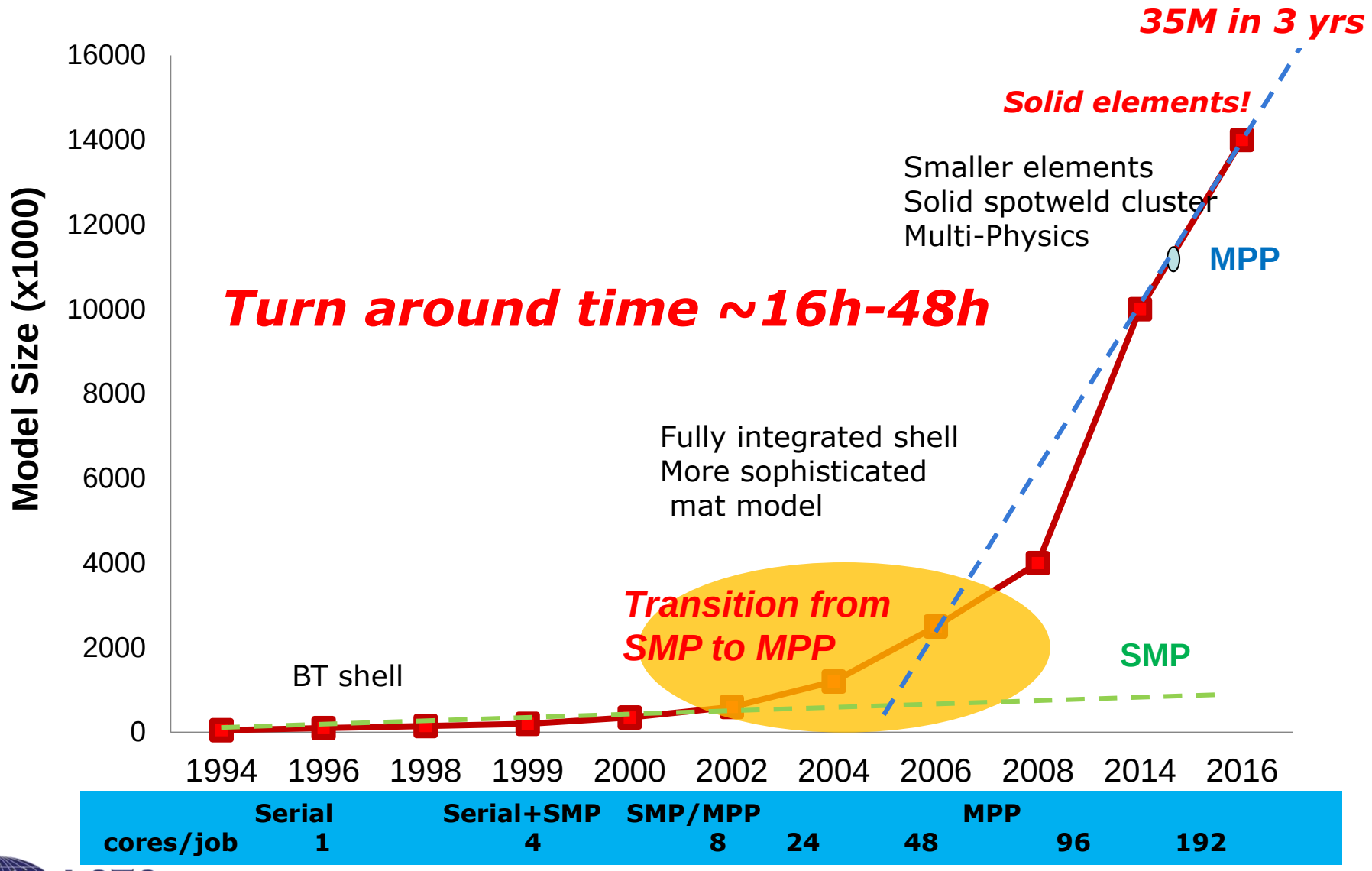


- CrachFEM materials
- Detailed HAZ welds
- Multiple airbags
- Rolling wheels
- New powertrain models
- New chassis models
- Steering mechanics from wheel to steering wheel

Courtesy of: VOLVO CAR CORPORATION

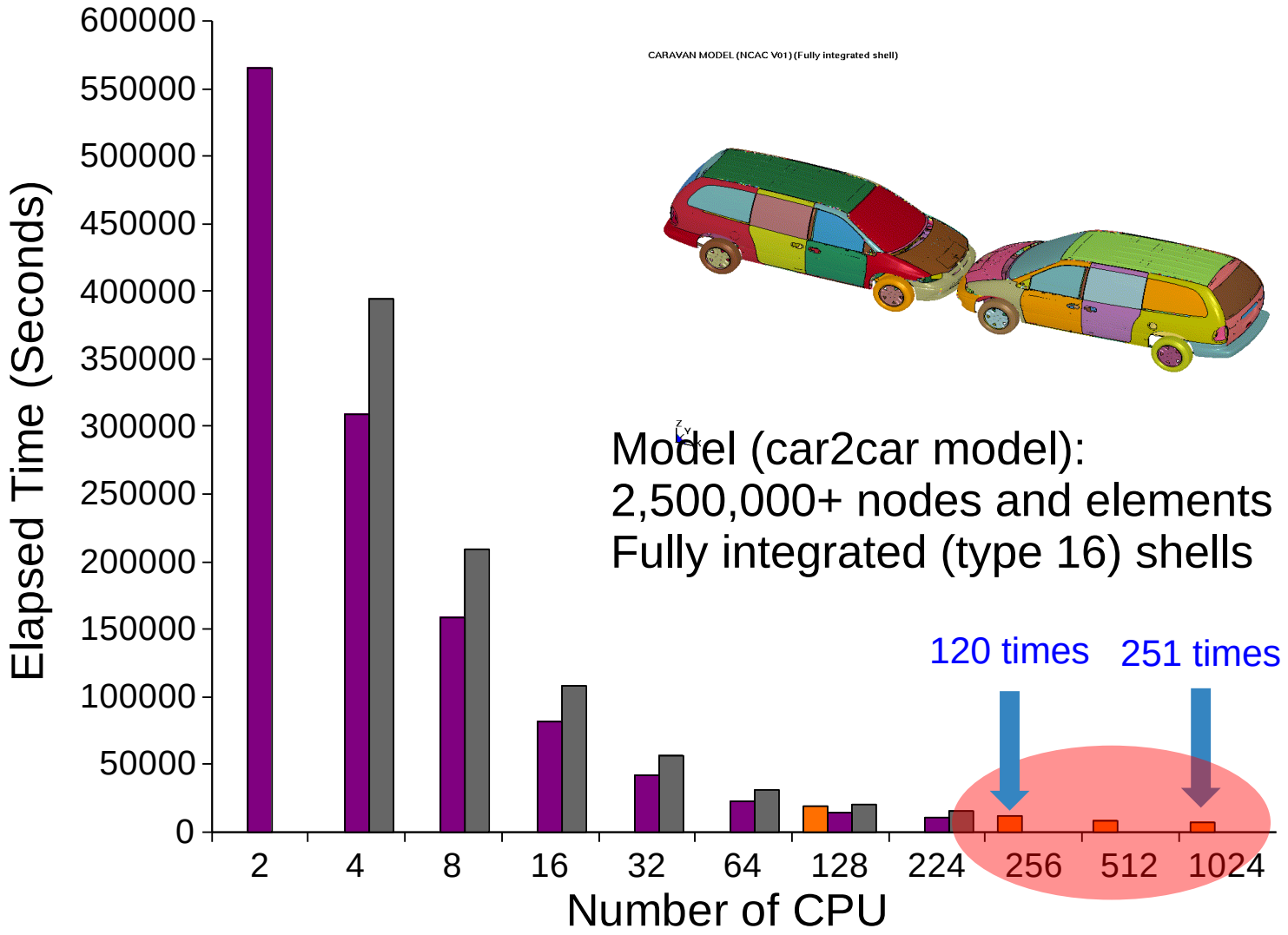
LSTC
Livermore Software Technology Corp.

Model size for Satety Analysis



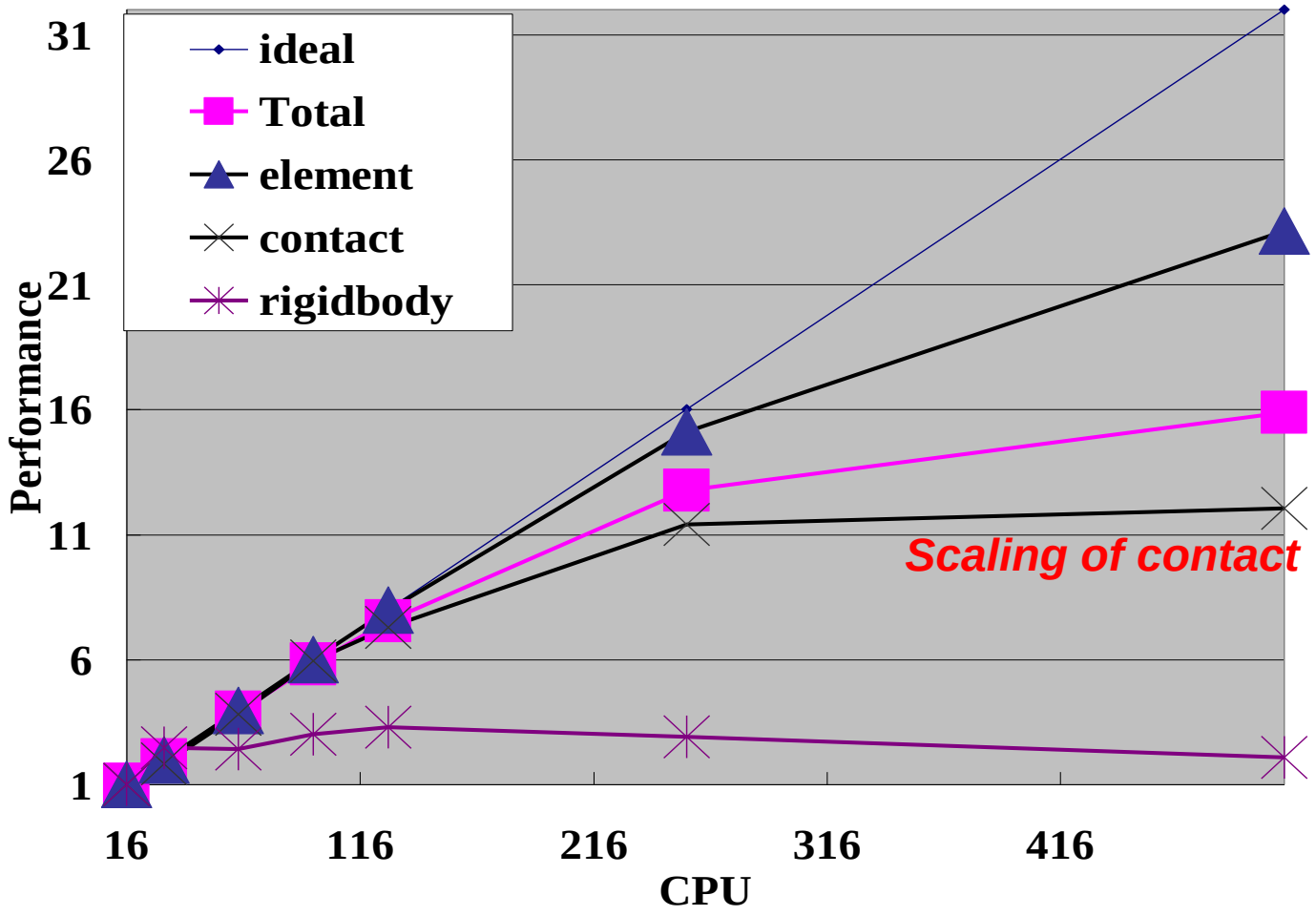
LSTC
Livermore Software
Technology Corp.

MPP performance – topcrunch.org (2007)



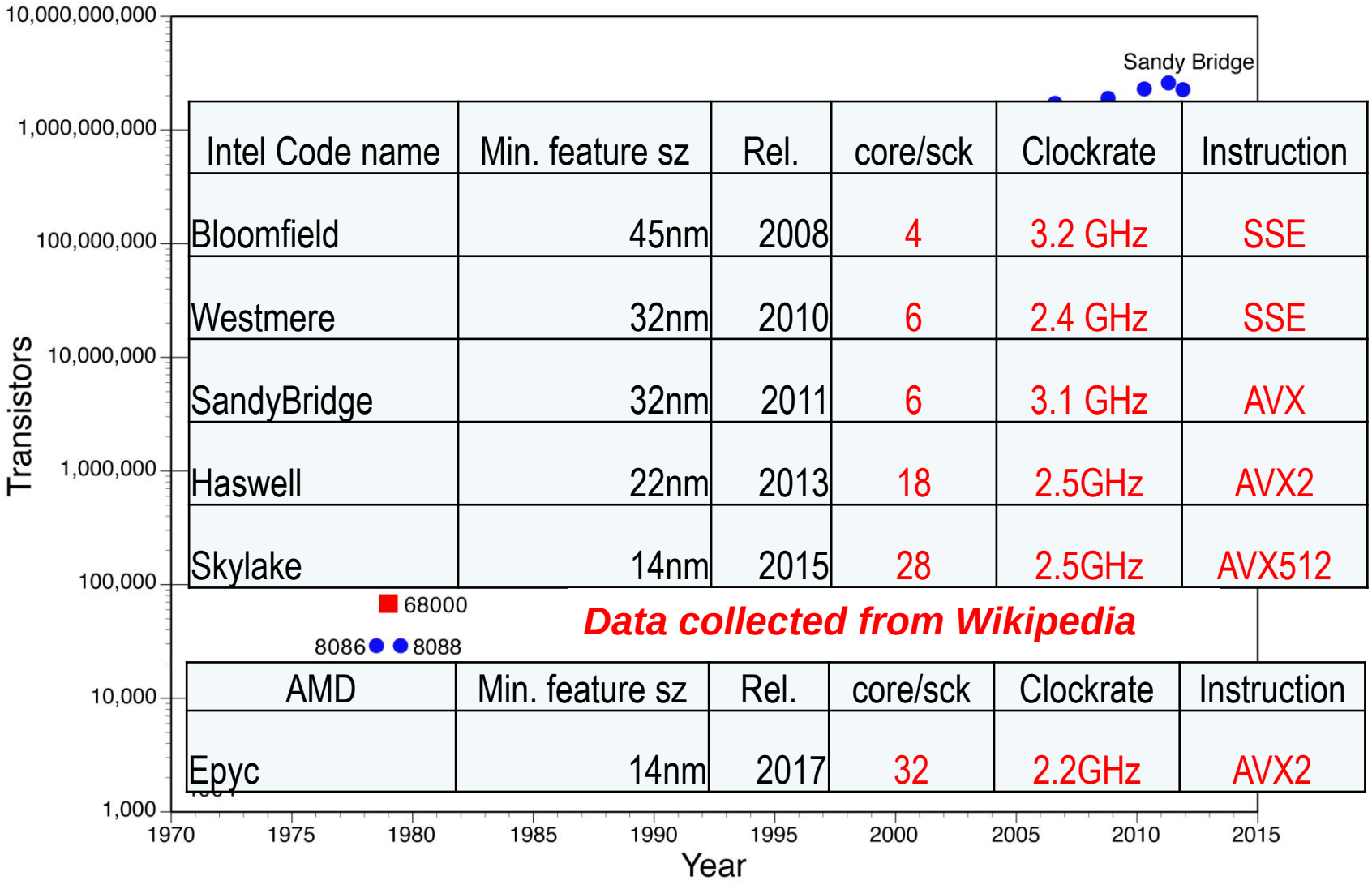
Livermore Software Technology Corp.

MPP scaling



LSTC
Livermore Software Technology Corp.

CPU technology - Moore's Law



Intel Code name	Min. feature sz	Rel.	core/sck	Clockrate	Instruction
Bloomfield	45nm	2008	4	3.2 GHz	SSE
Westmere	32nm	2010	6	2.4 GHz	SSE
SandyBridge	32nm	2011	6	3.1 GHz	AVX
Haswell	22nm	2013	18	2.5GHz	AVX2
Skylake	14nm	2015	28	2.5GHz	AVX512

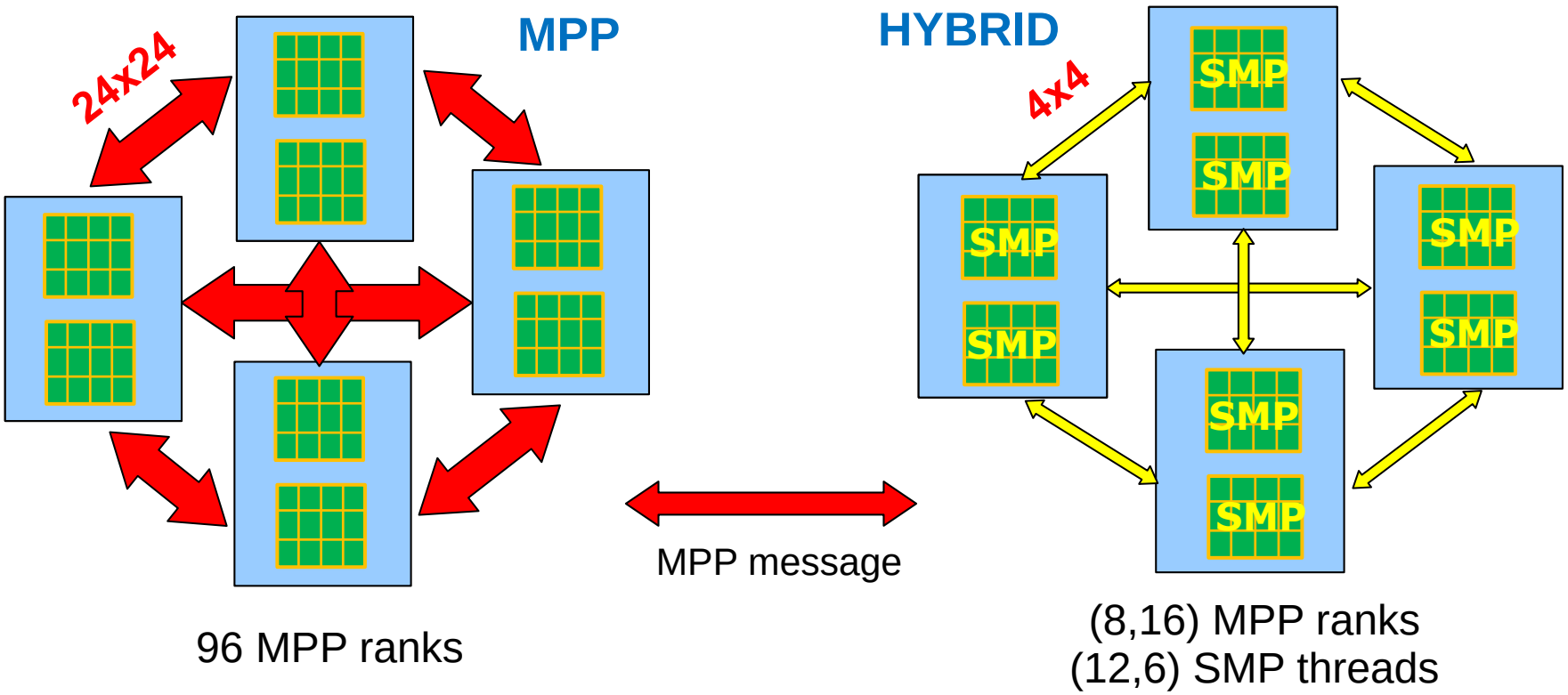
Data collected from Wikipedia

AMD	Min. feature sz	Rel.	core/sck	Clockrate	Instruction
Epyc	14nm	2017	32	2.2GHz	AVX2

LSTC
Livermore Software Technology Corp.

SMP and MPP – HYBRID

12core/2socket 4 nodes clusters

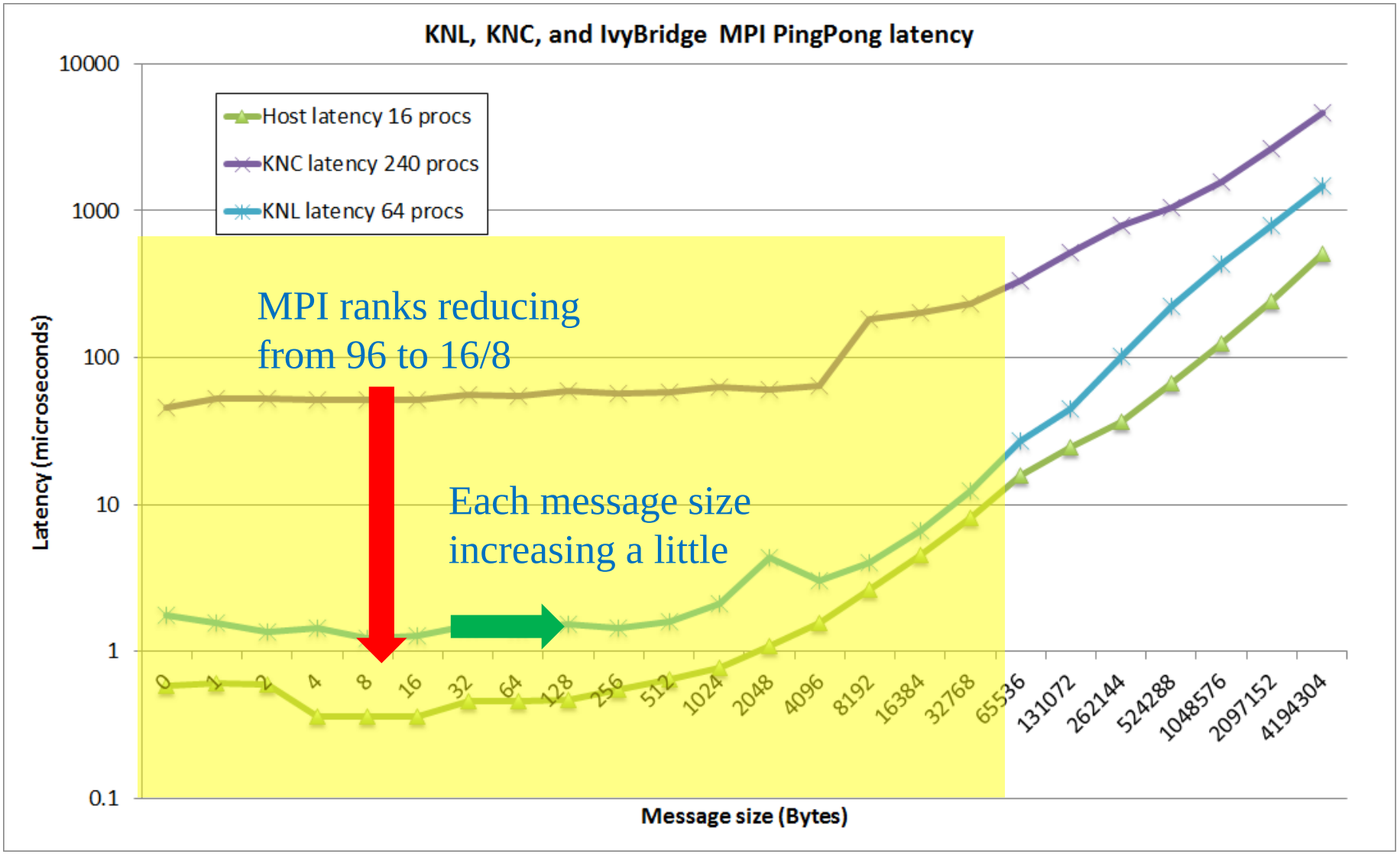


$T_{elapse}=T_{cpu}+T_{sys}+T_{mpp}$

$T_{elapse}=T_{cpu}+T_{sys}+T_{mpp}+T_{omp}$

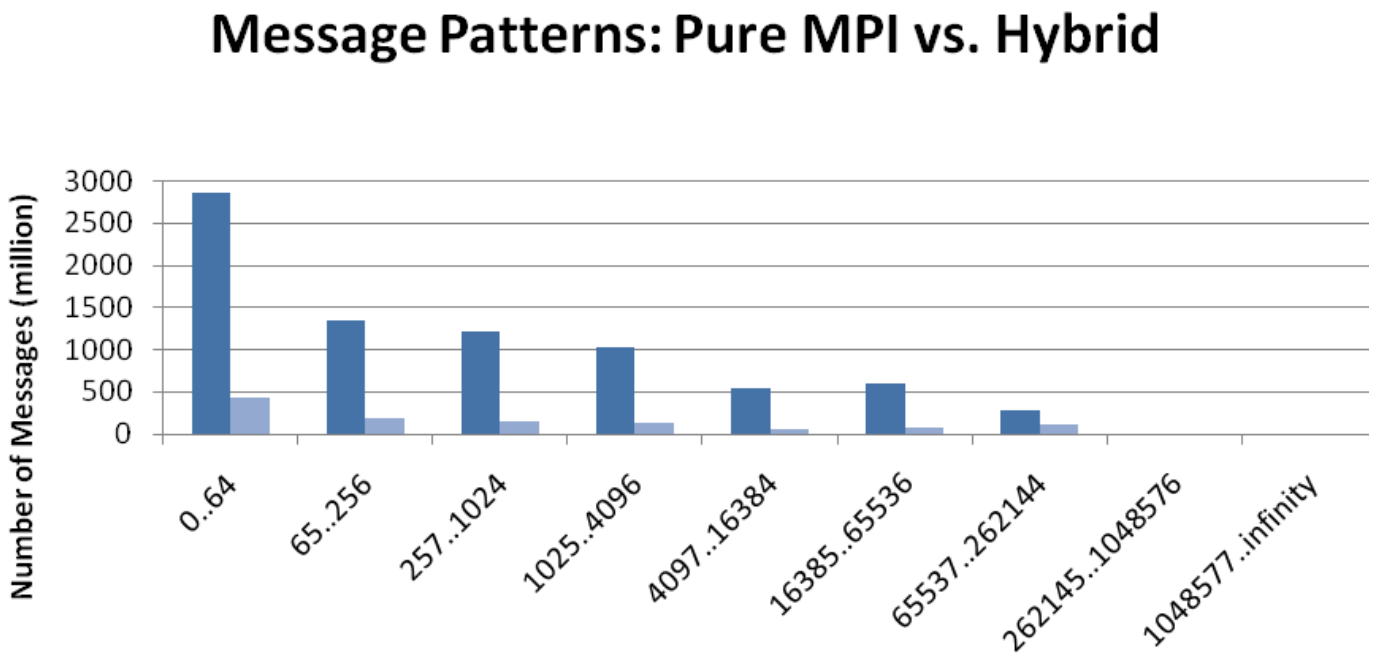
LSTC
Livermore Software Technology Corp.

Message latency - T_{mpp}

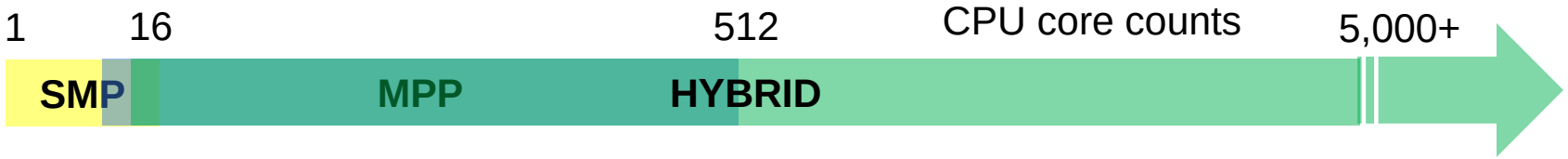


LSTC
Livermore Software
Technology Corp.

Message size comparison

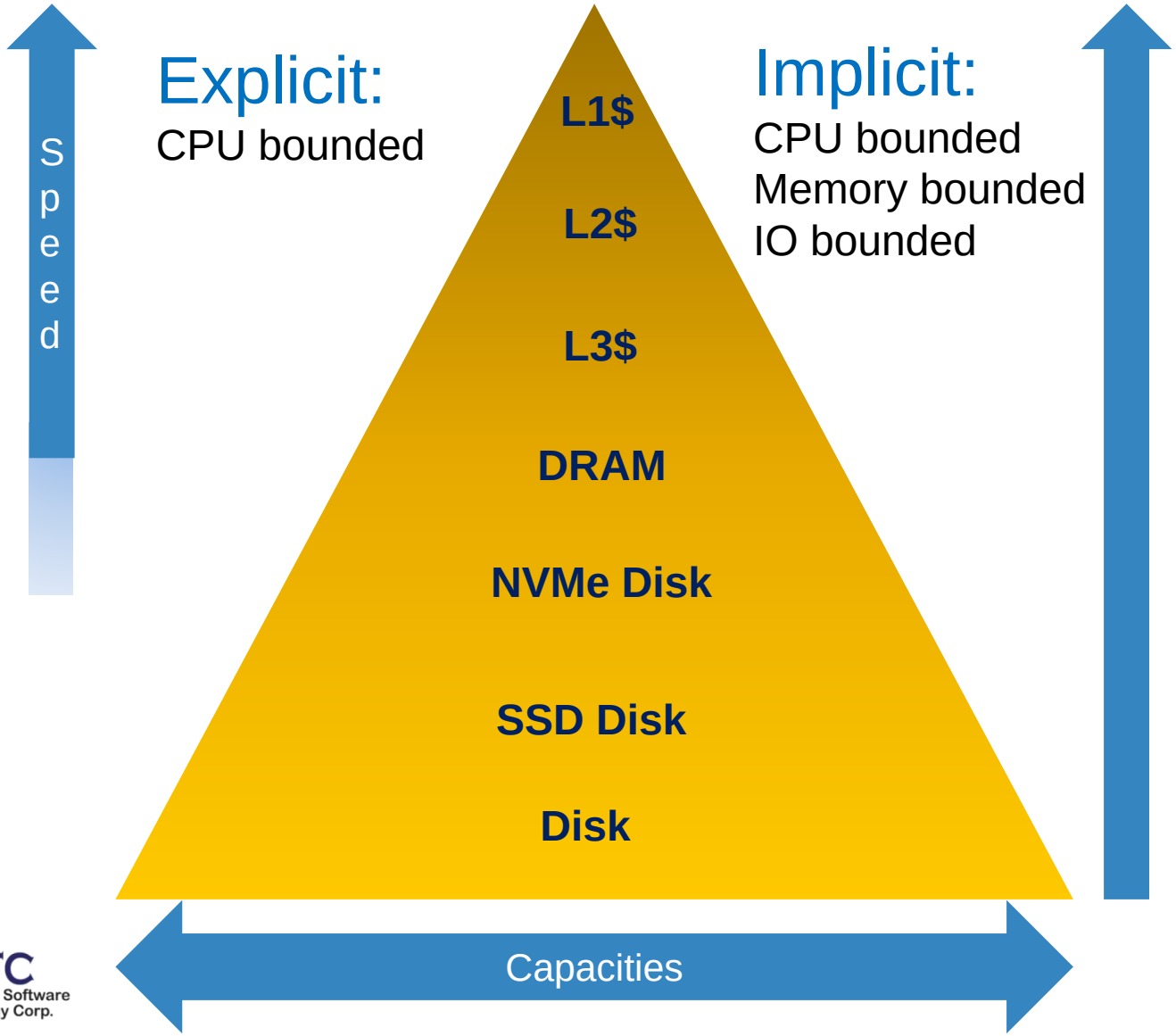


- Total message size and latency reduced by ~ thread^2 (T_{mpp}⬇)
- Number of local files reduced by ~ thread (T_{sys}⬇)



LSTC
Livermore Software Technology Corp.

Explicit vs Implicit



Numerical consistency

SMP

- ncpu= - # (~10% slower)

  To give identical results while changing threads

MPP - keep same number of MPP ranks

- lstc_reduce (~2% slower)

  To give identical results while changing CPU cores

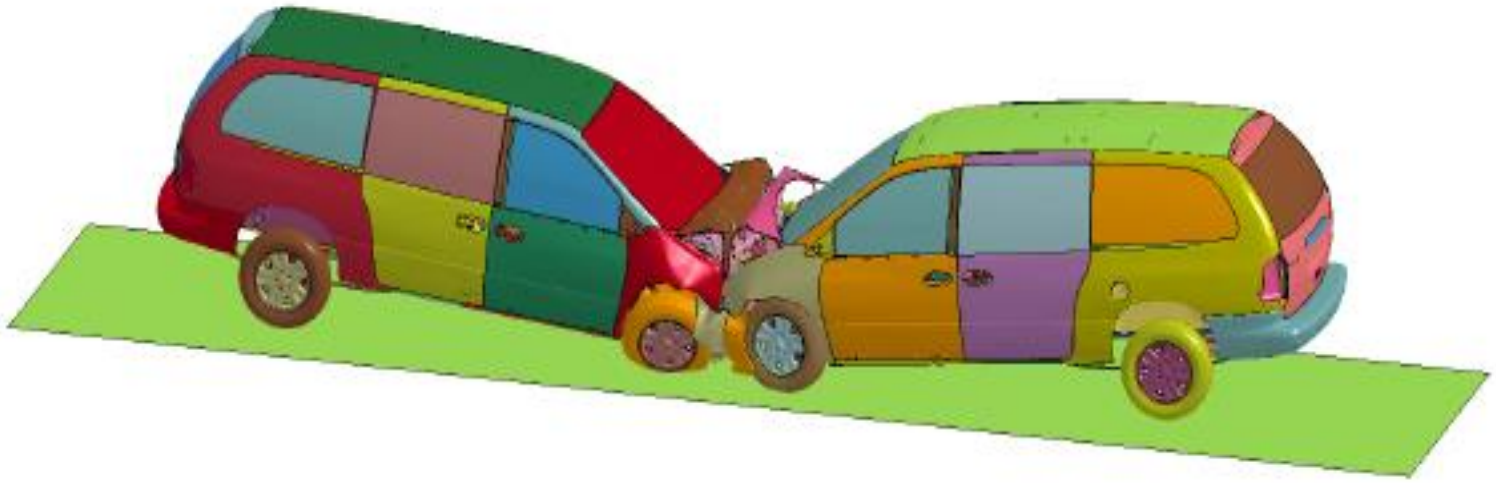
- rcblog

  To preserve the domain lines to reduce the noise from changing meshes

HYBRID = SMP + MPP

Explicit MPP/Hybrid Performance

Multi-core/Multi-socket clusters



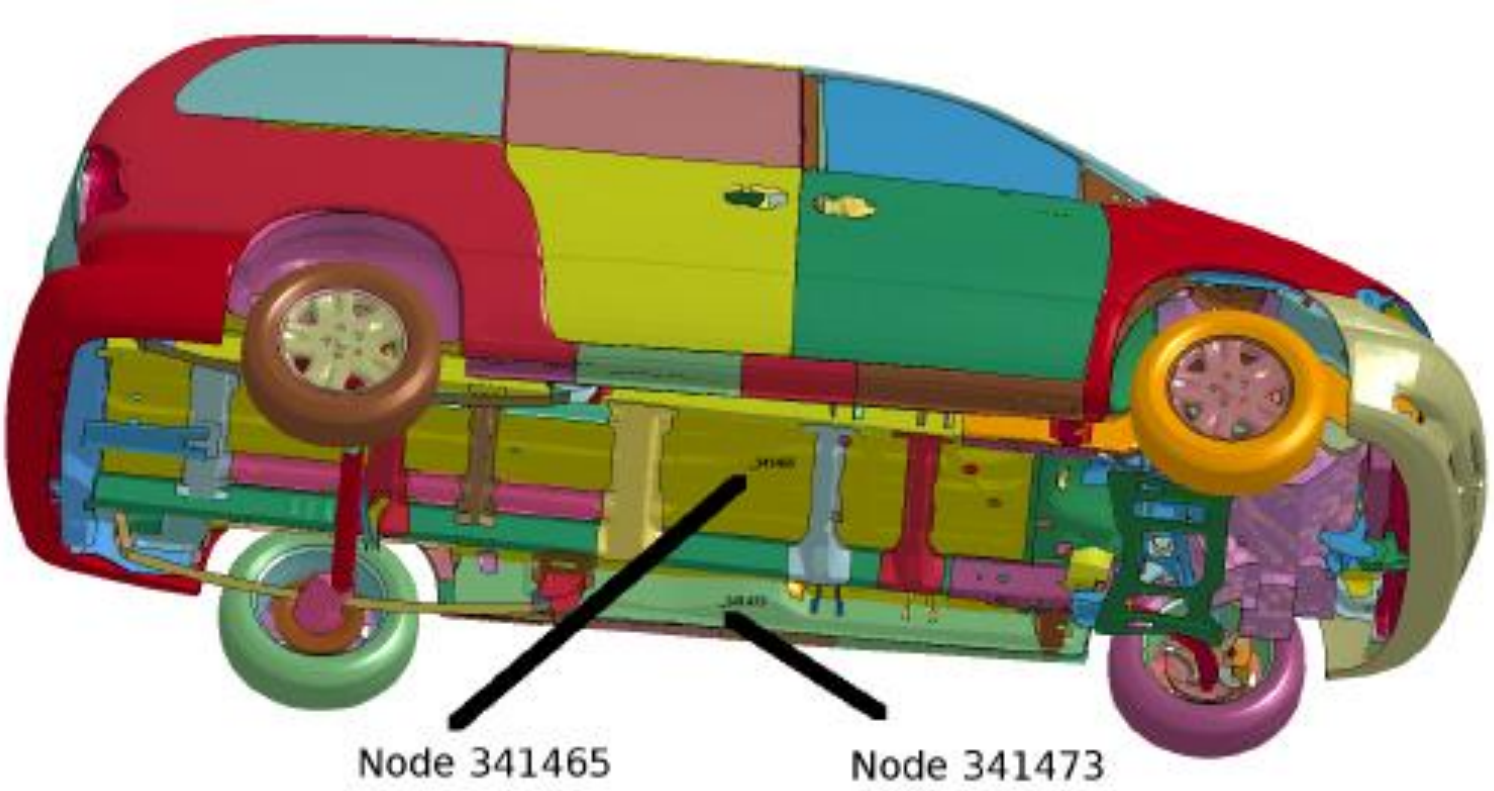
Car2Car model at time $t = 100\ [ms]$.

- NCAC model
- 2.5M elements, 53 contacts, type 16 shell
- 35 mph frontal
- 100 ms

LSTC
Livermore Software Technology Corp.

Explicit MPP/Hybrid Performance

Multi-core/Multi-socket clusters



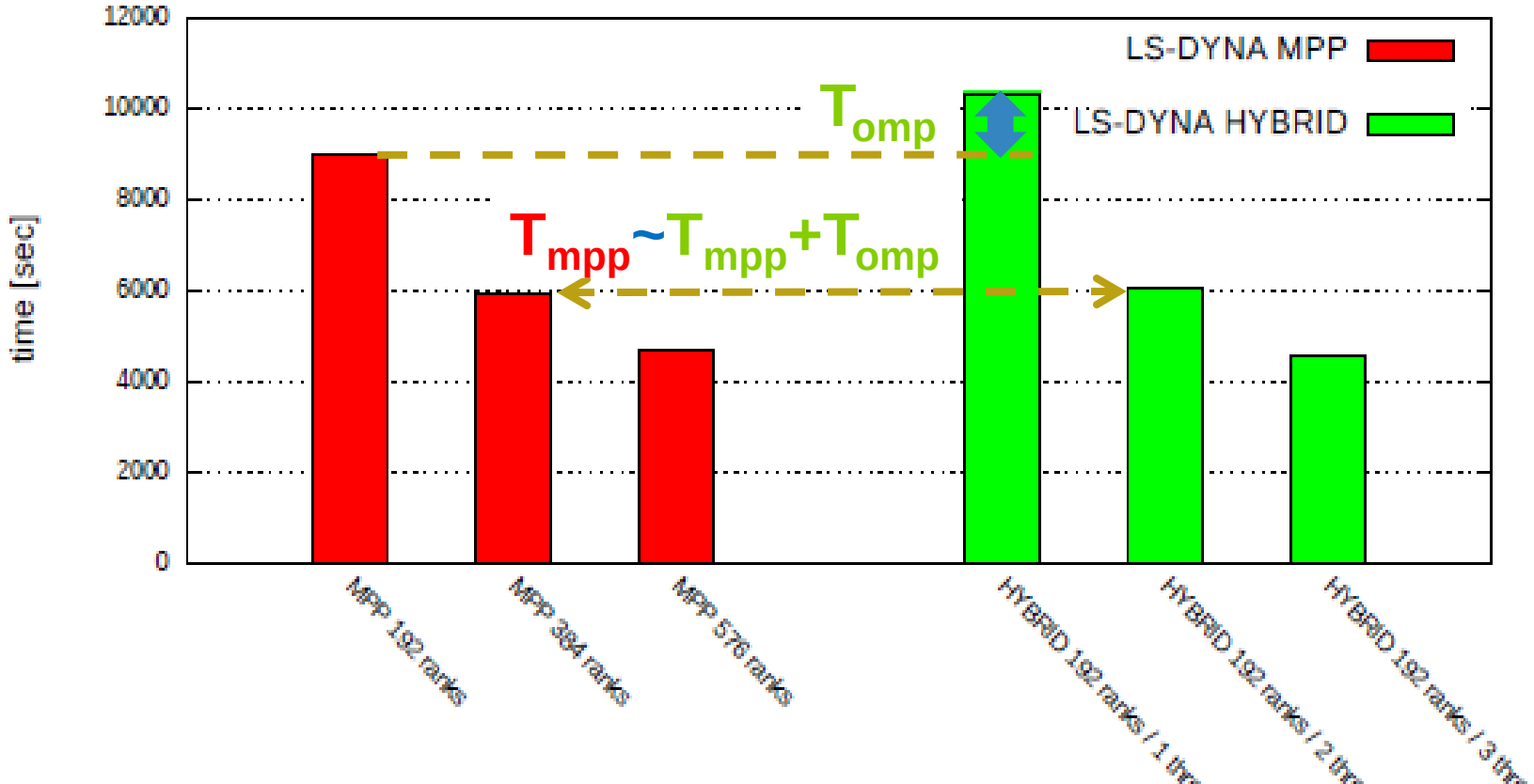
- Nodes to monitor

Explicit MPP/Hybrid Performance

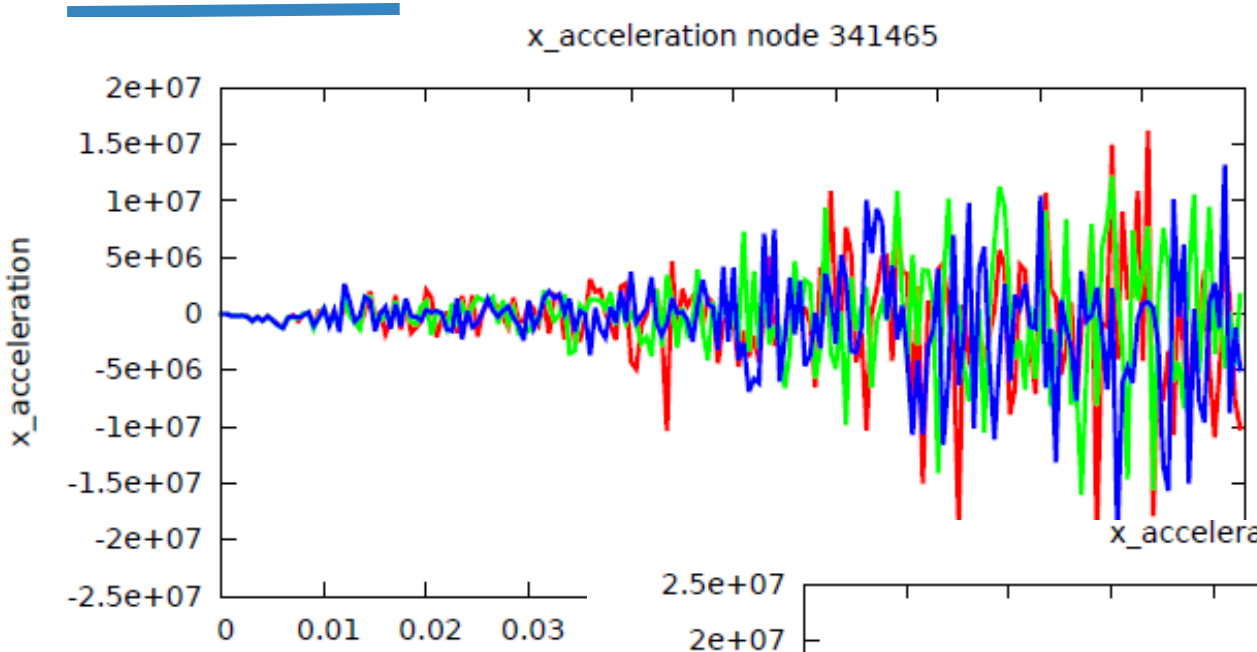
HYBRID 192 ranks series - Set up

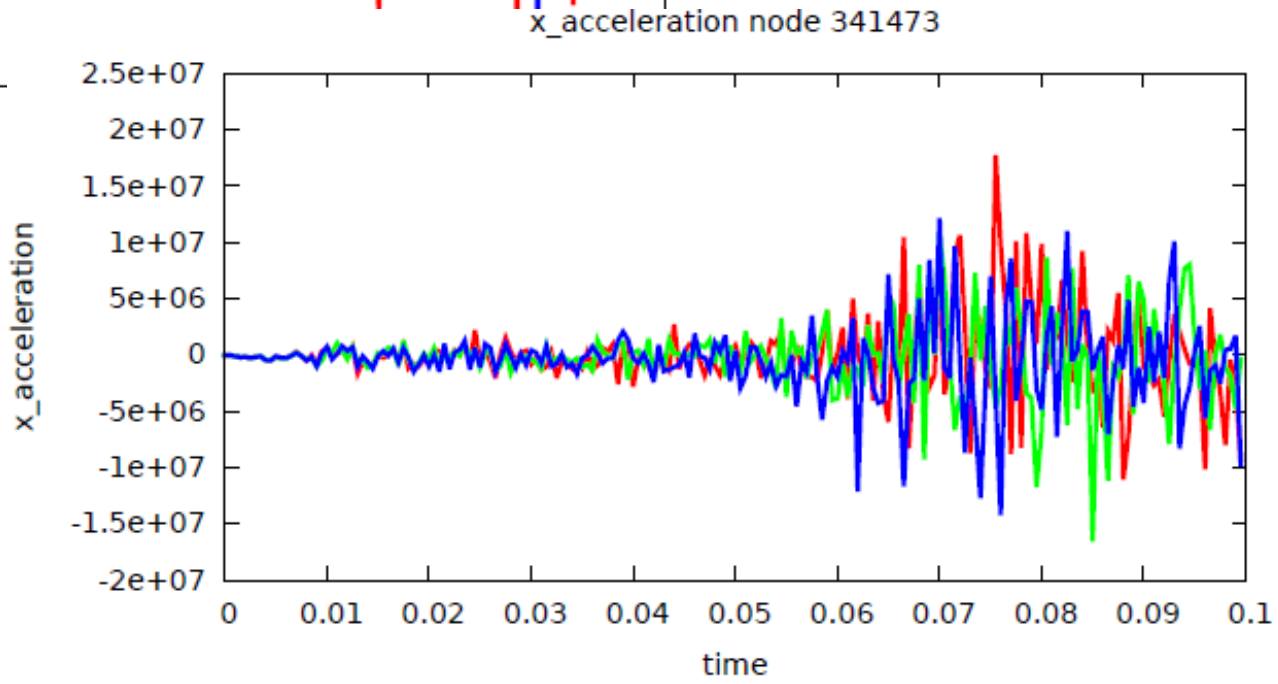
Total core count	LS-DYNA MPP MPP ranks	LS-DYNA HYBRID MPP ranks	LS-DYNA HYBRID SMP thread(s)
192	192	192	1
384	384	192	2
576	576	192	3

Varying SMP threads but fixed decomposition



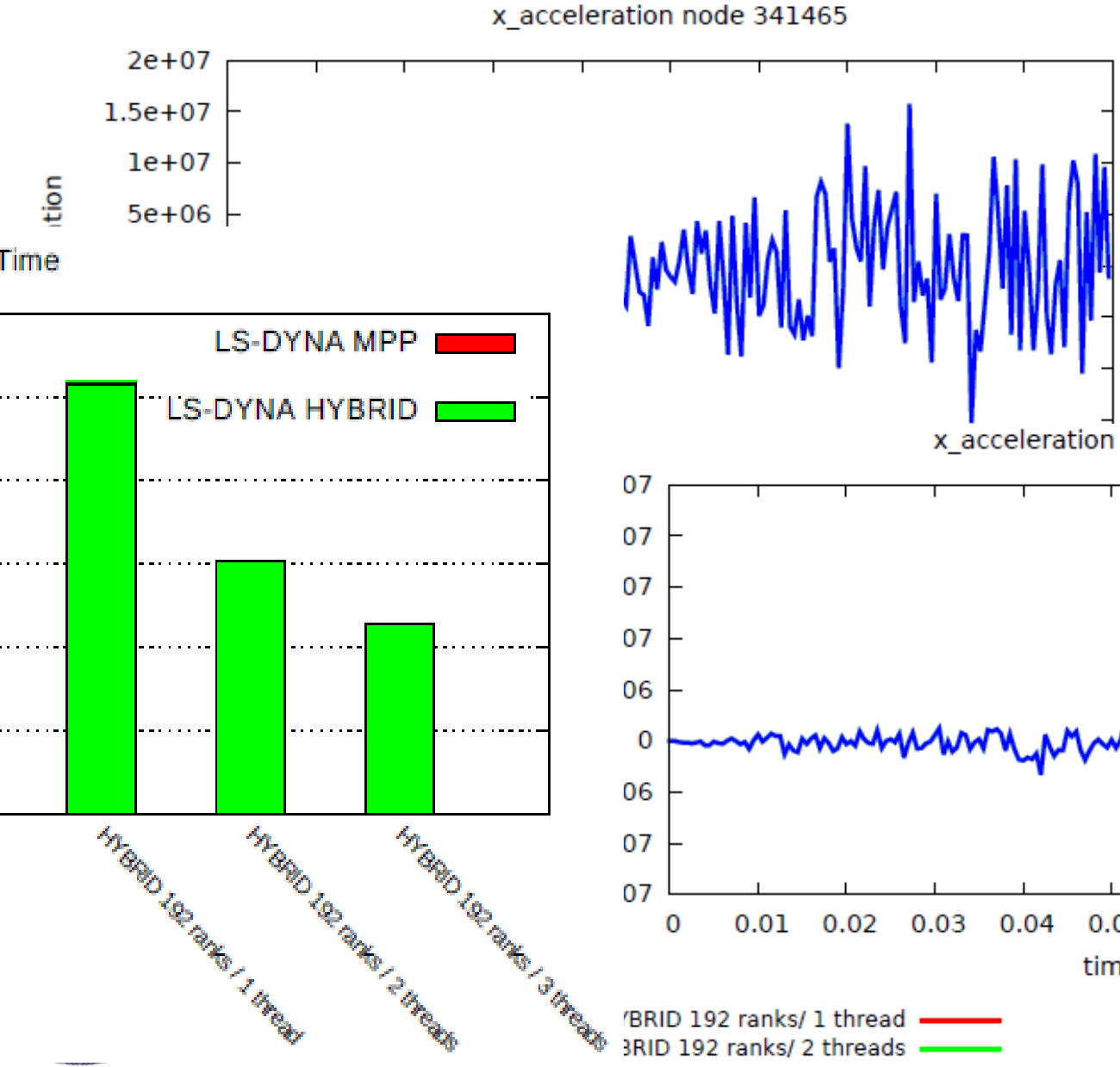
Explicit MPP/Hybrid Consistency





LSTC
Livermore Software Technology Corp.

Explicit MPP/Hybrid Consistency



Implicit Hybrid Performance

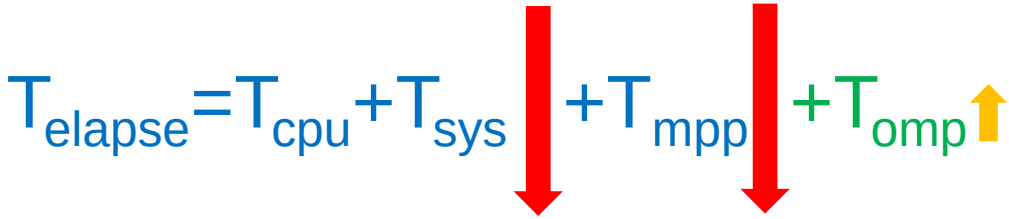
Model CYL0P5E6 – Intel 6core 2.95 GHz

Node x MPI x OMP	N=2	N=4	N=8	N=16
Nx1x1	5074	2761	1500	850
Nx1x4	1564	918	571	373
Nx1x6	1209	720	465	318

Implicit MPP/Hybrid Performance

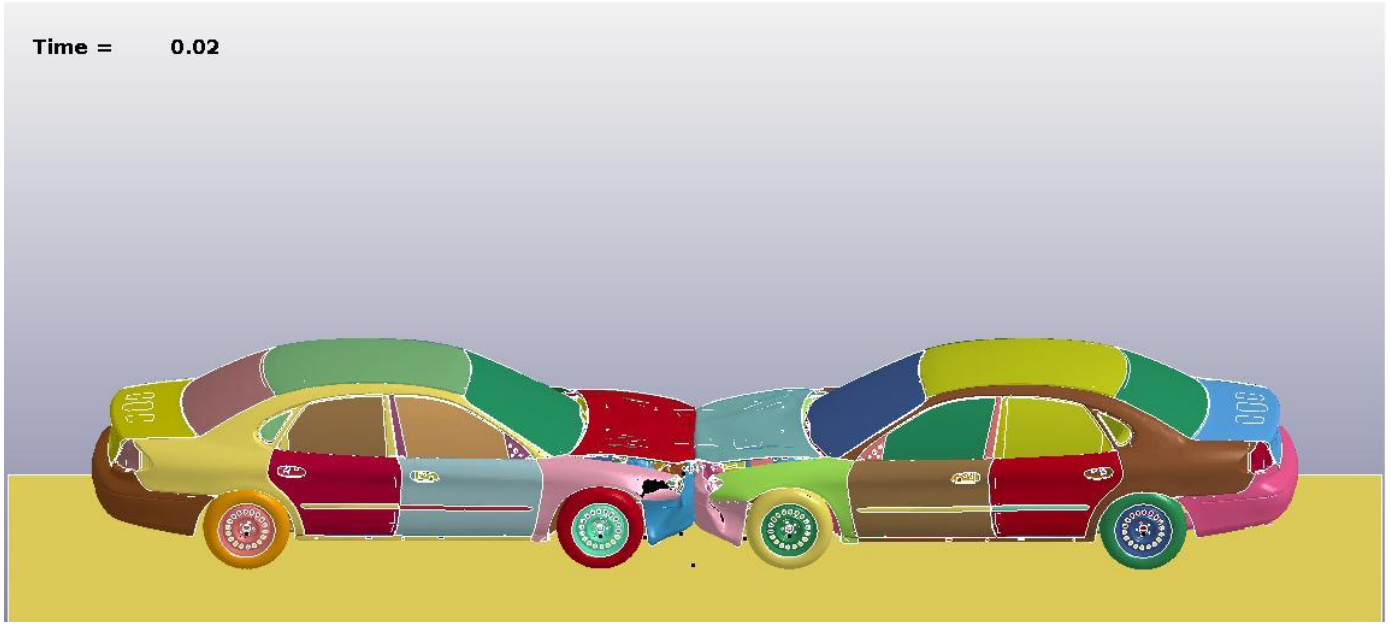
Performance on Linux AMD64 systems

No. of cores (node x socket x core)	WCT of Factor Matrix (seconds)	WCT for job to complete (seconds)
16 x 4 x 1	2055	14417
16 x 4 x 2	985	13290
16 x 4 x 4	582	29135
16 x 4 x 4omp (Hybrid)	960	9887



LSTC
Livermore Software Technology Corp.

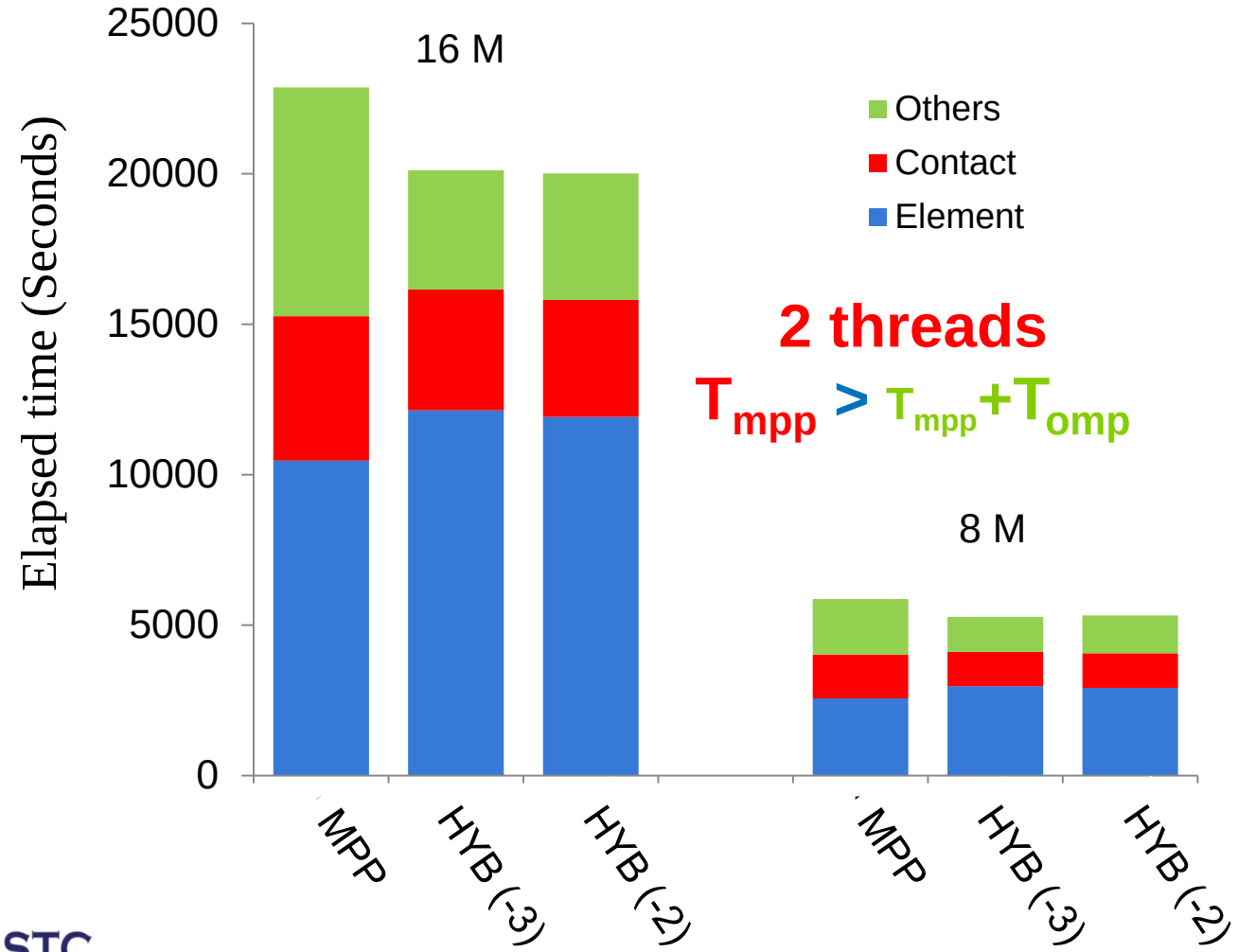
Hybrid Performance vs Model Size



- Dynapower 2cars models (Dr. Makino)
- 6 contacts, type 16 shell, 8 M/16M
- 35 mph frontal
- 20 ms

Explicit Hybrid Performance vs Model Size

Ls-dyna R9.1 SVN:117280, 240 cores/job



LSTC
Livermore Software Technology Corp.

How to choose your executable

Explicit analysis

- SMP - 1-2 cores
- MPP - 2-100 cores
- HYBRID
  - More than 100 cores
  - Run under different core counts but need consistent answers

Implicit analysis

- HYBRID
  - Keep as much data in memory for incore solution
  - Reduce IO bottleneck
  - Reduce amount of data transferring in the network

Summary

For better prediction and to reduce cost of prototyping

- Models involve coupled multi physics, HAZ, CPM airbag, etc.
- Mesh becomes finer and critical parts are using solids
- More sophisticated contact (mortar) for better behavior
- Gives better scaling for explicit analysis with more then 100 cores
- Gives better performance in most of the implicit analysis

Thank you !

- Hybrid has already used in production (stable)
- More capabilities will become OpenMP enabled to get better speedup

LSTC
Livermore Software Technology Corp.