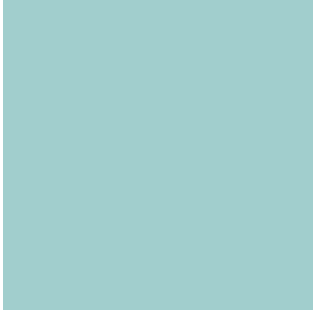




Analyzing LS-DYNA performance on clusters

Jimmy Forsberg

Concept of this presentation

- What to look for in output files for LS-DYNA analysis?
- How to 'fix' (change) it.
- Rank - one MPP domain
 - MPP LS-DYNA → one rank, one core
 - Hybrid LS-DYNA → one rank, several cores
- Timescales
 - Communication on same CPU - nanoseconds
 - Communication over network - microseconds

When to shift to hybrid?

- Find sweetspot for MPP using your model
- Too reduce communication over network
- When the number of elements on each rank are too few (20 000-50 000?)
- Large number of cores

Basic idea for a load balanced model

- There are especially 2 instances to check for load balancing of a problem in LS-DYNA:
 - Element treatment
 - Contact treatment
- Typically this can only be analyzed on an average basis (after a simulation).
- The user can only help LS-DYNA to do a better decomposition as of now.
- There are tools in LS-DYNA to measure element load balancing and make use of those timing reports (can improve performance a lot on some models).
- When it comes to effective contact performance - contact pair on same CPU as much as possible.
- In future, LS-DYNA will be able to do re-decomposition (ongoing work).
- In future, LS-DYNA will be able to judge when to do re-decomposition (ongoing work).

Effective simulation

- Is not a concern for this presentation
- There are fast features and slow features in LS-DYNA
 - LSTC always work to improve performance as bottlenecks are revealed
 - Add MPP capability
 - Add MP communication capability
 - Vectorize code
 - Better support for hardware features like AVX2
 - Etc, etc...
- Memory utilization - improve (reduce) memory needs

Vital output

■ From LS-DYNA:

- GLSTAT
 - nzc
- Load_profile.xy
- Cont_profile.xy
- D3hsp
- Message files

■ MPI

- In depth studies requires some output from MPI
 - Message size
 - Amount of time spent in different MPI calls

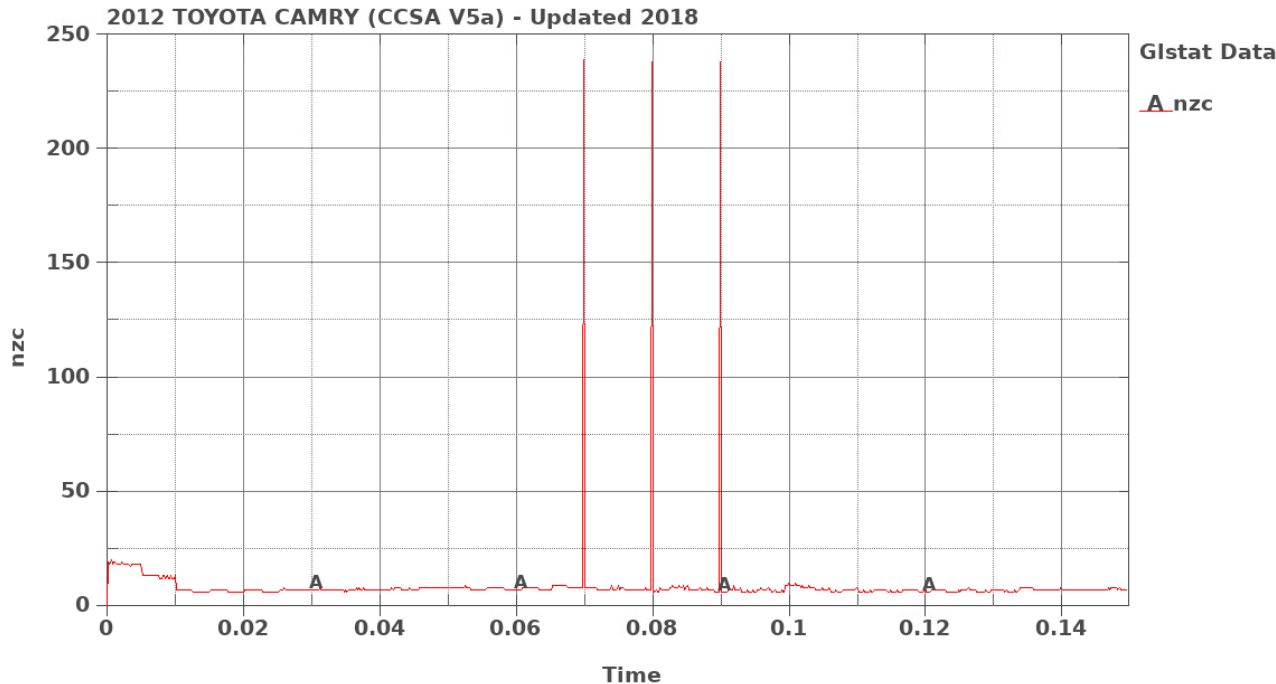


■ Synchronization point in LS-DYNA

- Time step : if core finished with elements it will collect time here. Hence high value here compared to rest of cores → waited a long time for other cores to finish their element calculation routines.
- Decomposition controls this
 - The cost of the element varies over time in the simulation.
 - User-Defined material - LS-DYNA do not know the cost.
- Rigids
 - Small synchronization point - indication of contact un-balanced
 - Others - synchronization after contacts - contact un-balanced
- Contacts
 - If many contacts in one rank - possible wait between contact evaluation for information from other rank.
- Number per zone cycle - time for cycle/'number of elements'. Preferably fairly constant through out simulation. Likely to increase at bsort cycle or load unbalancing due to severe deformation.
If not - explain it.

Overall performance during simulation - GLSTAT file

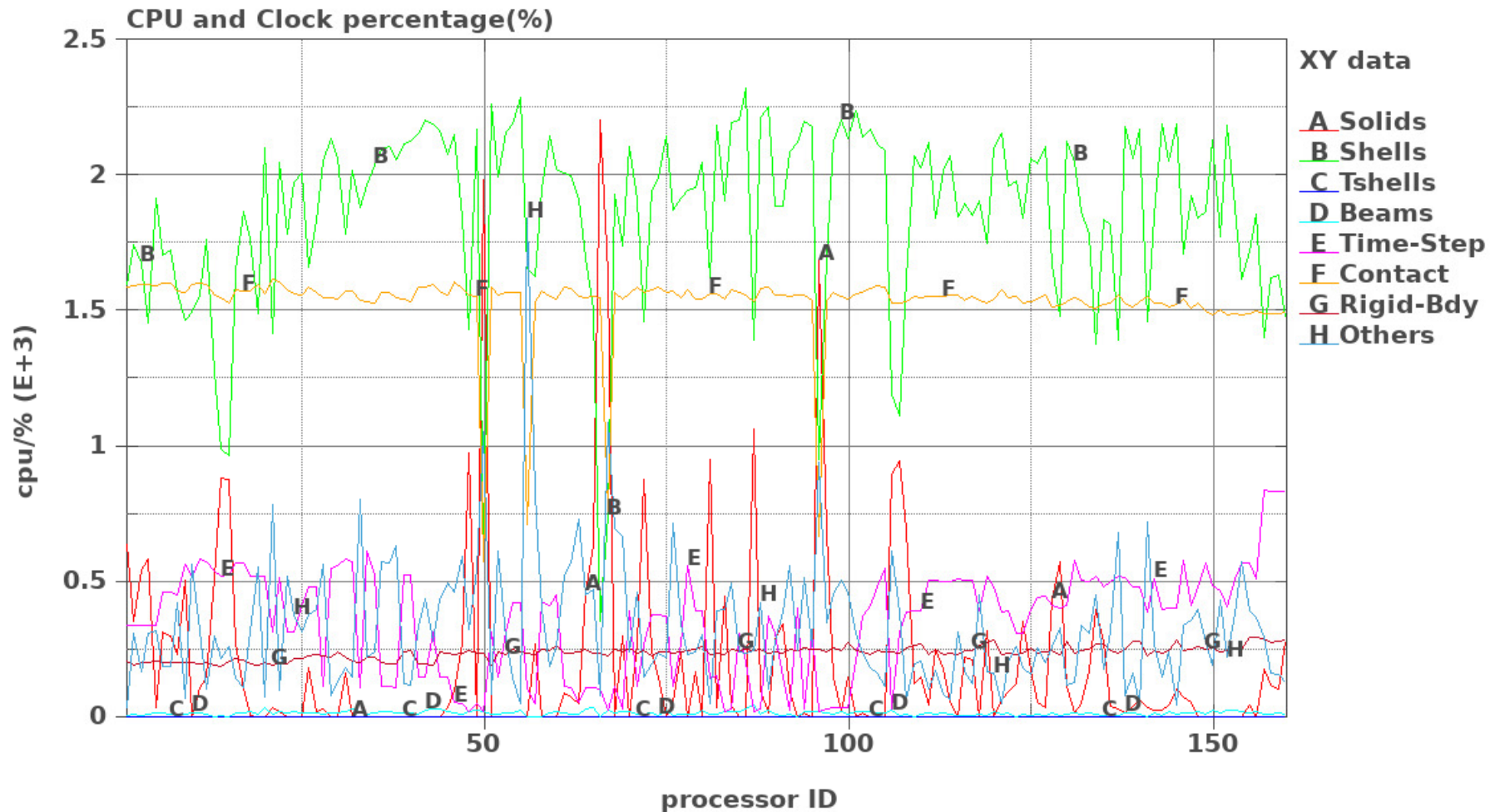
- Nzc - number zone cycle (time for a cycle/'number of elements')
- Preferably fairly constant
- Slight increase with complicated deformation.
- Will be higher during burst of contacts.



Load profile

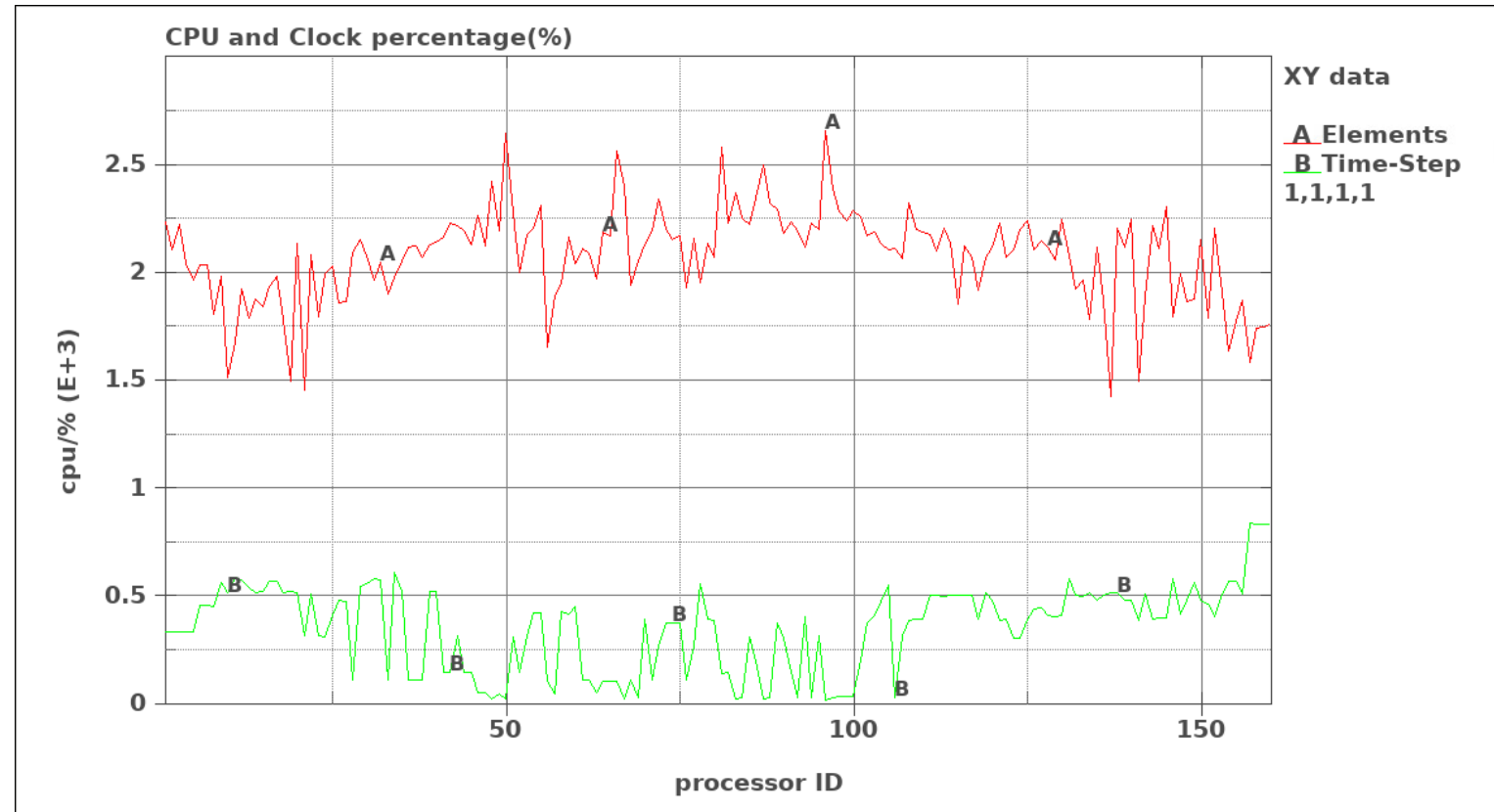
- X-axis: rank
- Y-axis: timing data from table in mes****. In CPU seconds or % of total time.
 - Solids
 - Shells Fully parallel – should scale as long as sufficient number of DOFs/rank.
Depends on:
 - Decomposition
 - Estimated element+material cost at t=0 (and if this changes over time)
 - Tshells
 - Beams
 - Timestep ← If element load is not equal – the rank will stand here.
Rank with high time here → too few elements
 - Contact
 - Rigid-Body ← If high values found here → interpreted as many contacts on rank and waiting for information
→ groupable!
 - Others ← If high values found here → interpreted most often as contact imbalance between ranks
- Summerized from t=0 to end of simulation.
 - An averaged performance is what you get from a simulation.

Analyzing element distribution: Load profile view



Load profile - element distribution

- Sum all elements
(shells/solid/tshell/beam)
- Compare to 'timestep'
 - High value on timestep on core → waiting
- The curves are 'inverses' of each other



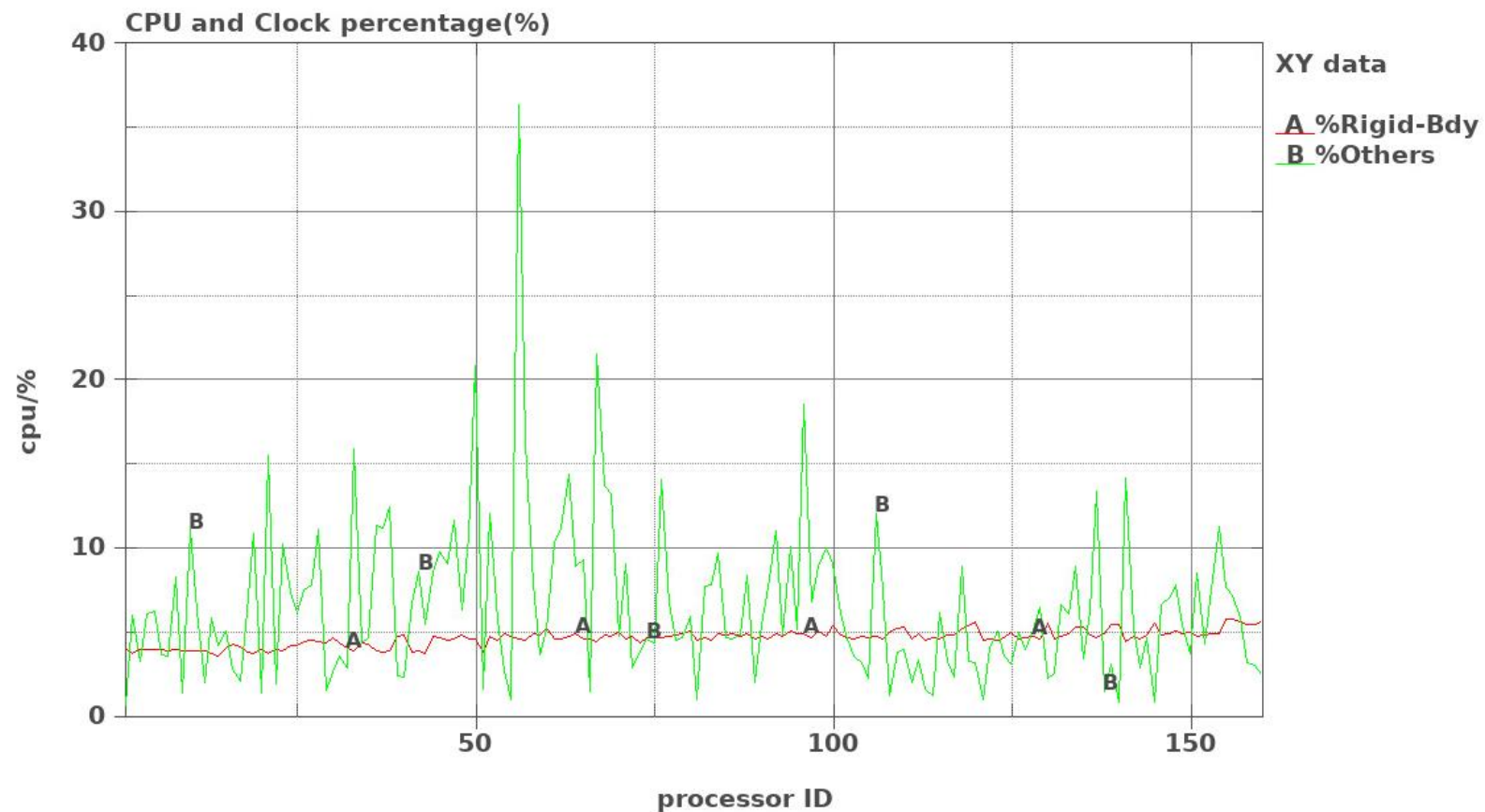
How to fix - decomposition

- Help LS-DYNA by distributing elements more properly.
 - Manual: user material to all cores, known cpu intensive material models, known material models where the cpu needed varies heavily on current state (changes through the simulation). Collect in part set and distribute to all processors using:
`*CONTROL_MPP_DECOMPOSITION_ARRANGE_PARTS`
 - Use of previous runs: timing_profile
 - Possible if the models are similar (also when it comes to PID numbering)
 - Measure time spent on each part on each cpu from a simulation.
 - User specification when to measure - use enough time (0-set deformation and then some).
 - Re-use this timing information in secondary run to help LS-DYNA distribute elements better.

Analyzing contact work distribution - Load profile

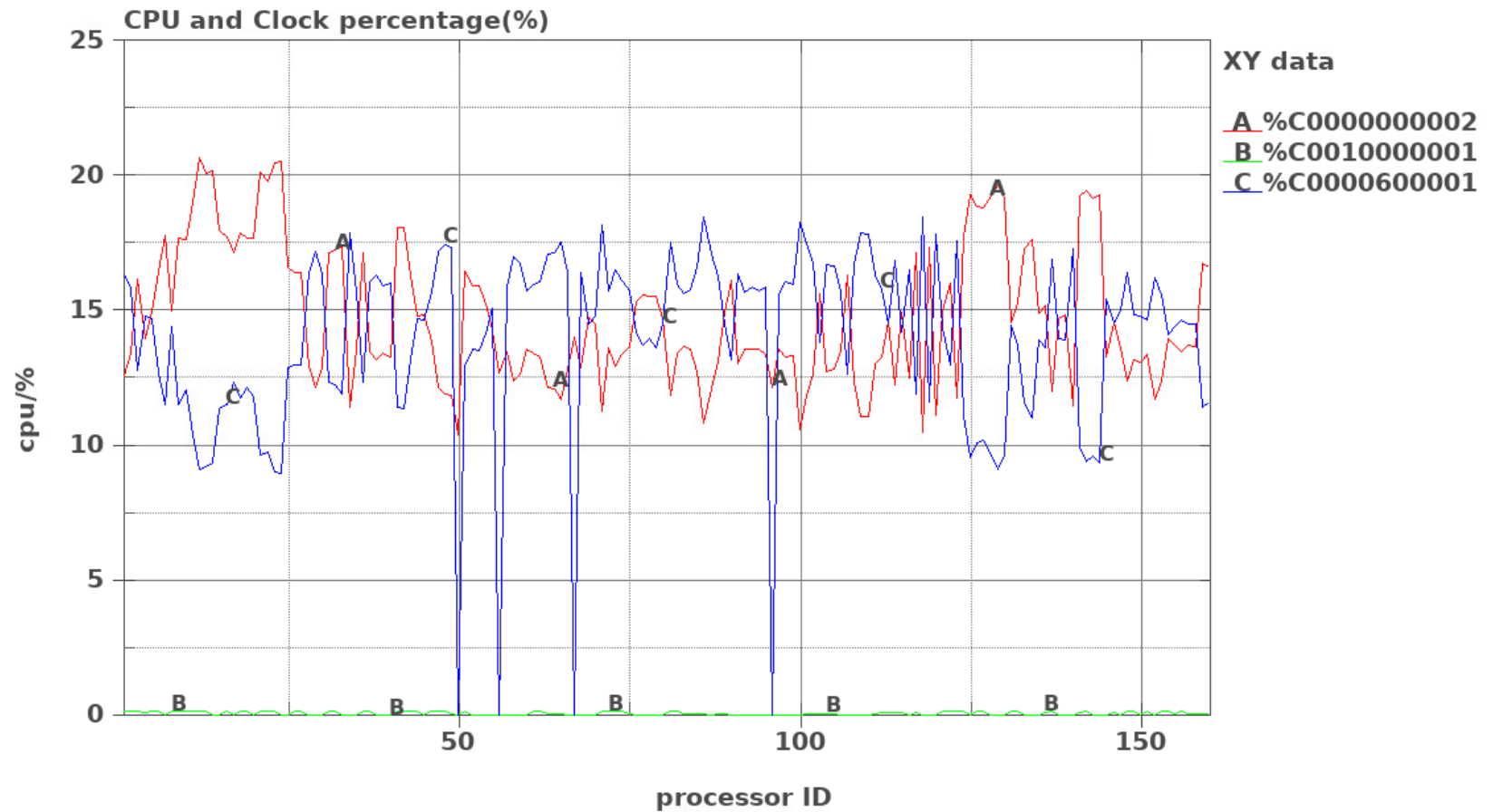
■ Load_profile.xy - Rigid and others

- Rigid 5% ok? Experience..
- Other - some cores waits...



Contact profile

- X-axis - rank
- Y-axis - any of the non-tied contacts in the model
 - Reveals a few cores with less contact.



How to fix?

■ Decomposition

- Distribute contact?
- Think contact pairs on same rank if possible, same CPU if possible, same node if possible.

■ Contact settings

- Groupable - if many different contact definitions on same rank

■ What makes contact performance deteriorate

- Size and shape of buckets
 - Mesh size differences in contact
 - Free flying elements in model

Thank you!



Your LS-DYNA distributor and
more

