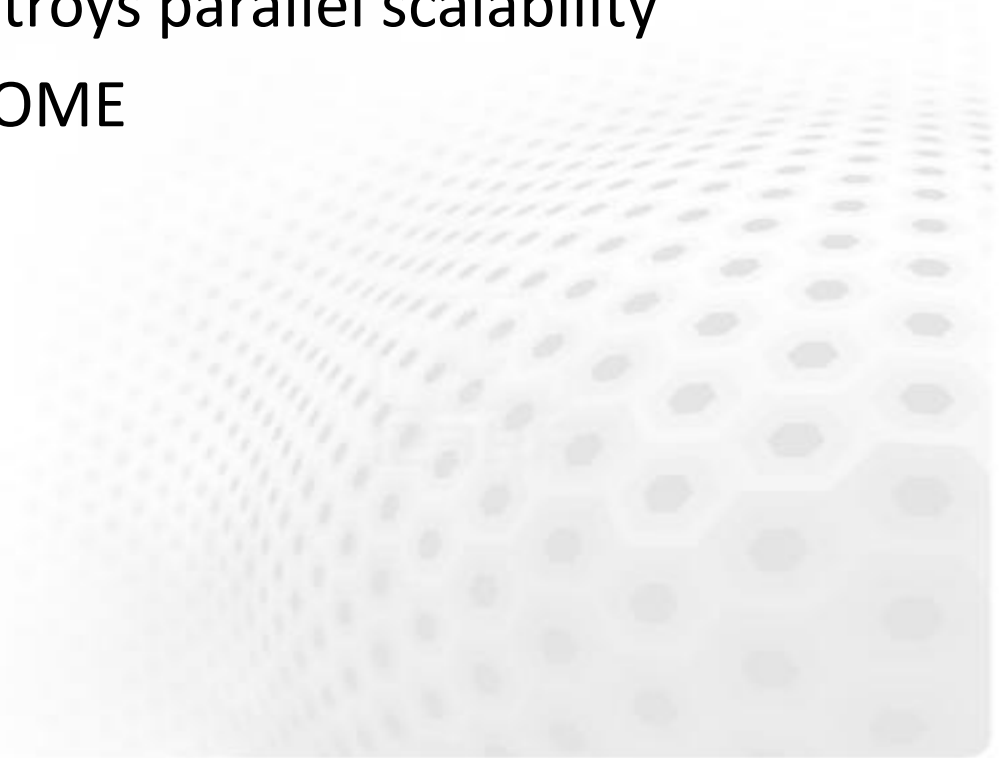




Offloading I/O in AROME

(Sami.Saarinen@csc.fi)

Outline

- Frequent field output destroys parallel scalability
 - Sub-space (SS) I/O for AROME
 - Preliminary results
 - Summary
- 

Acknowledgements

- This project was funded by PRACE under the title
“I/O-optimization to improve parallel scalability of
the meso-scale NWP-model HARMONIE”
- Joint effort between the Finnish Meteorological Institute
(FMI) and the CSC – IT Center for Science Ltd.
 - Sami Niemelä and Niko Sokka (FMI)
- Special thanks to HLRS for providing Cray XE6 resources

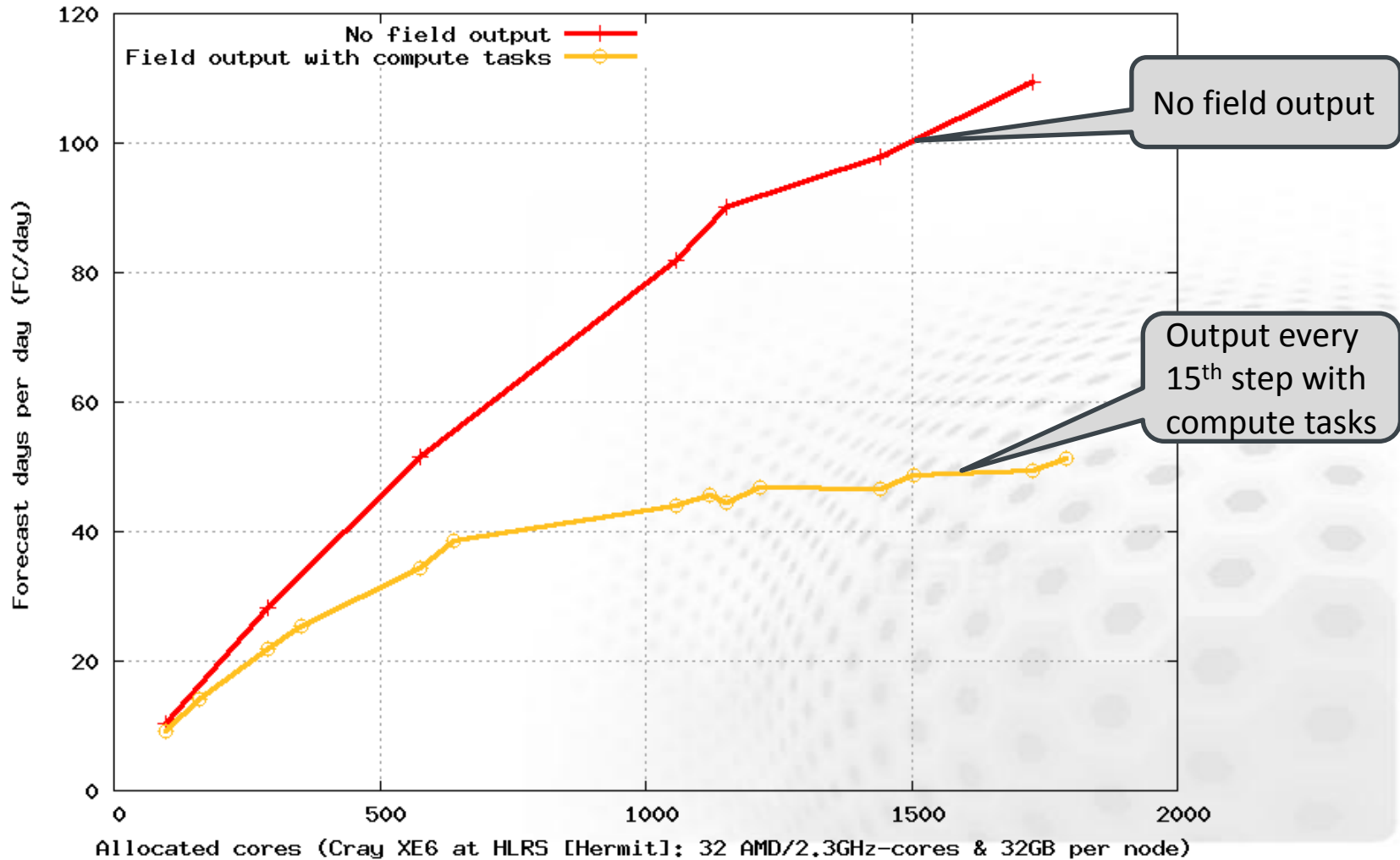
Frequent field output destroys parallel scalability

- The FMI requires relatively frequent field output from their AROME model : output at every 15th step is a typical value
 - Typically one minute time step is being used
 - The volume of FA-output in this work is ca. 500MBytes
- *The actual FA-output itself is not a big problem – instead*
- **gathering of the local fields (grid point & spectral) to the master task for the actual output is a big problem**
 - Prevents smooth progress of the forecast integration
 - Parallel scalability is therefore badly hit

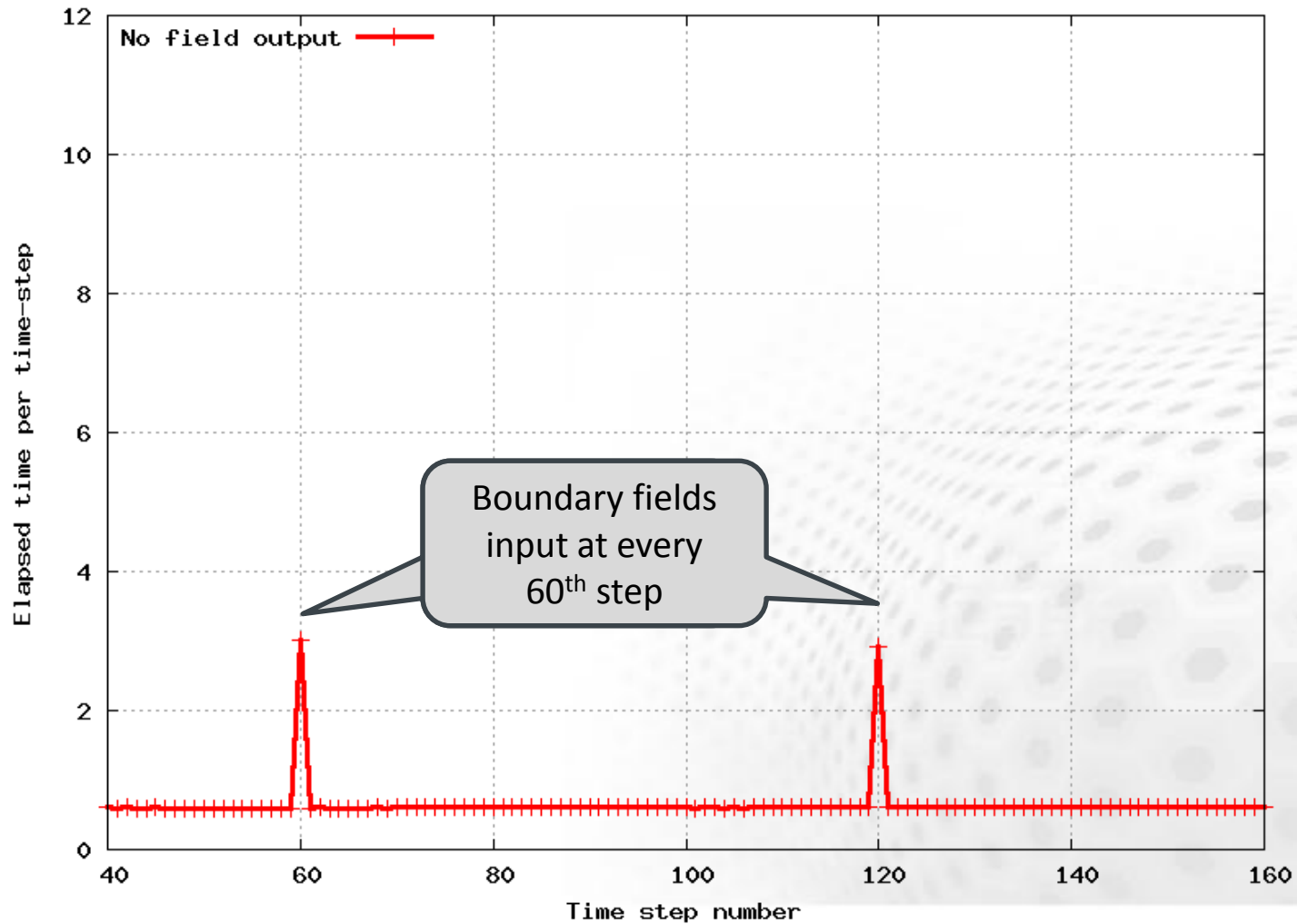
Some of the grim facts



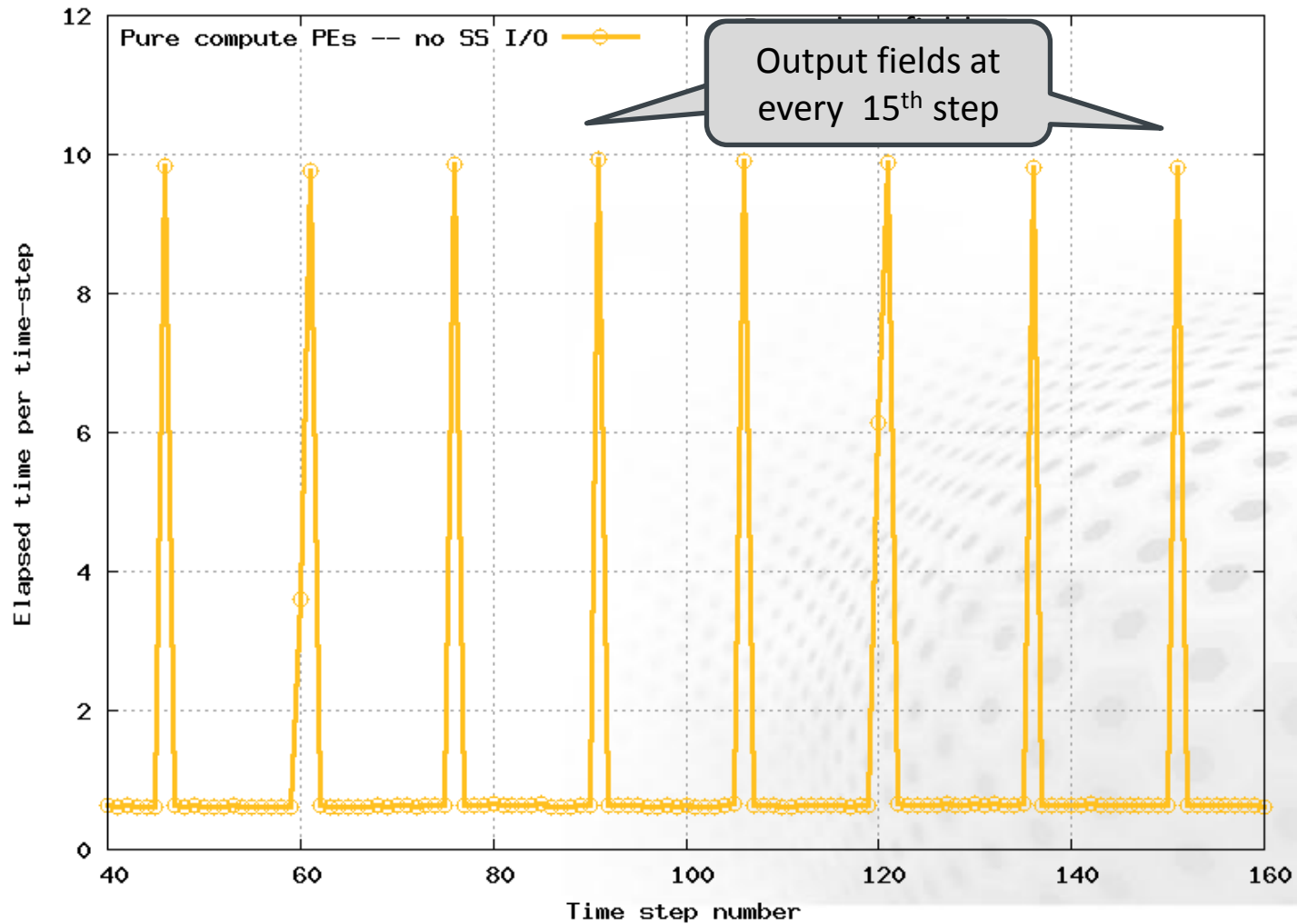
Harmonie/AROME pre-CY38 with sub-space (55) I/O at 512 x 600 x L60



Elapsed time per time-step (at 1152 compute tasks)



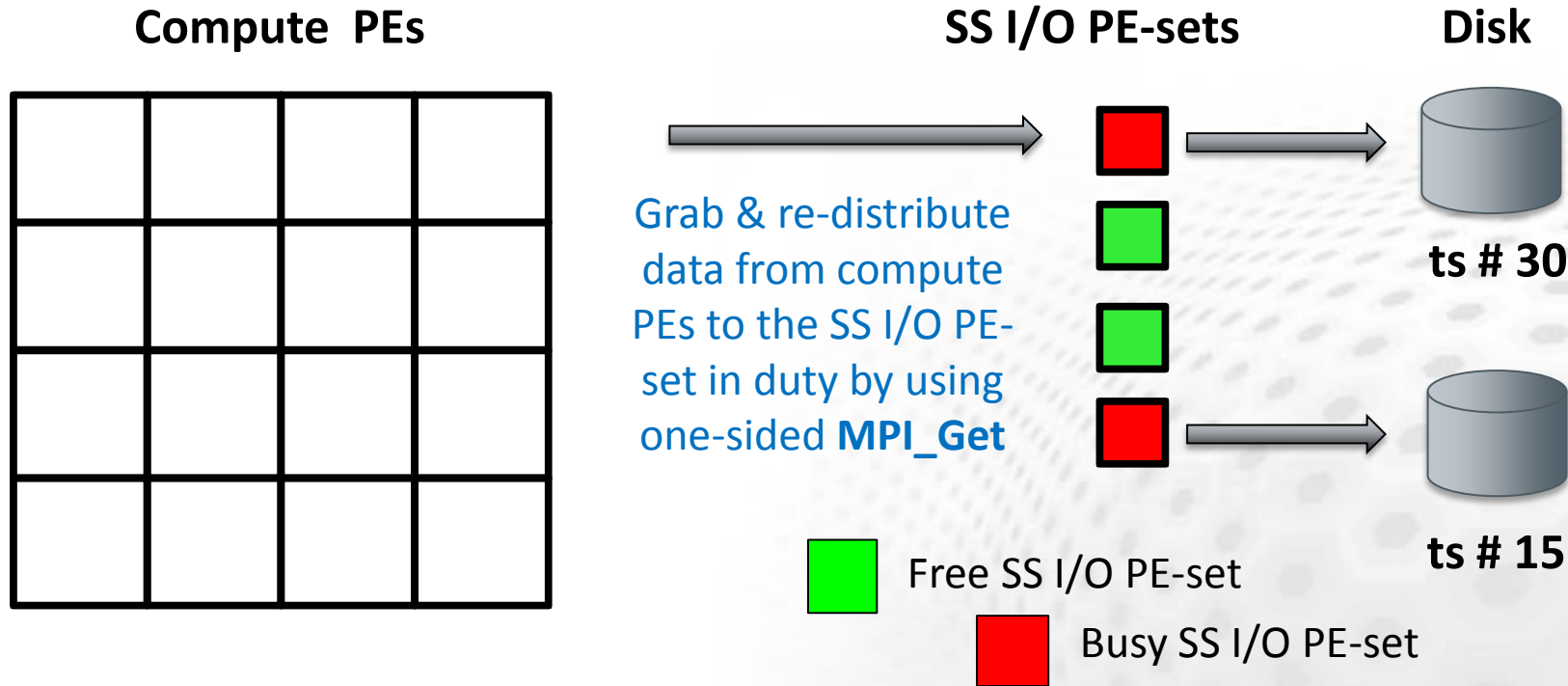
Elapsed time per time-step (at 1152 compute tasks)



Improving scalability with sub-space (SS) I/O

- Recall: the I/O itself is not a big problem – instead data gathering to the master compute task is ... indeed
- We found a scheme, where data gathering as well as field output can be **offloaded** away from the compute PEs
- In AROME this is possible by reorganizing calls to the WRSPECA and WRGP2FA – i.e. the main spectral and grid point field data gathering and output routines
- Routines are “overloaded” by a sub-space of I/O PEs letting computation to carry on – almost w/o disruption

The SS I/O scheme for AROME



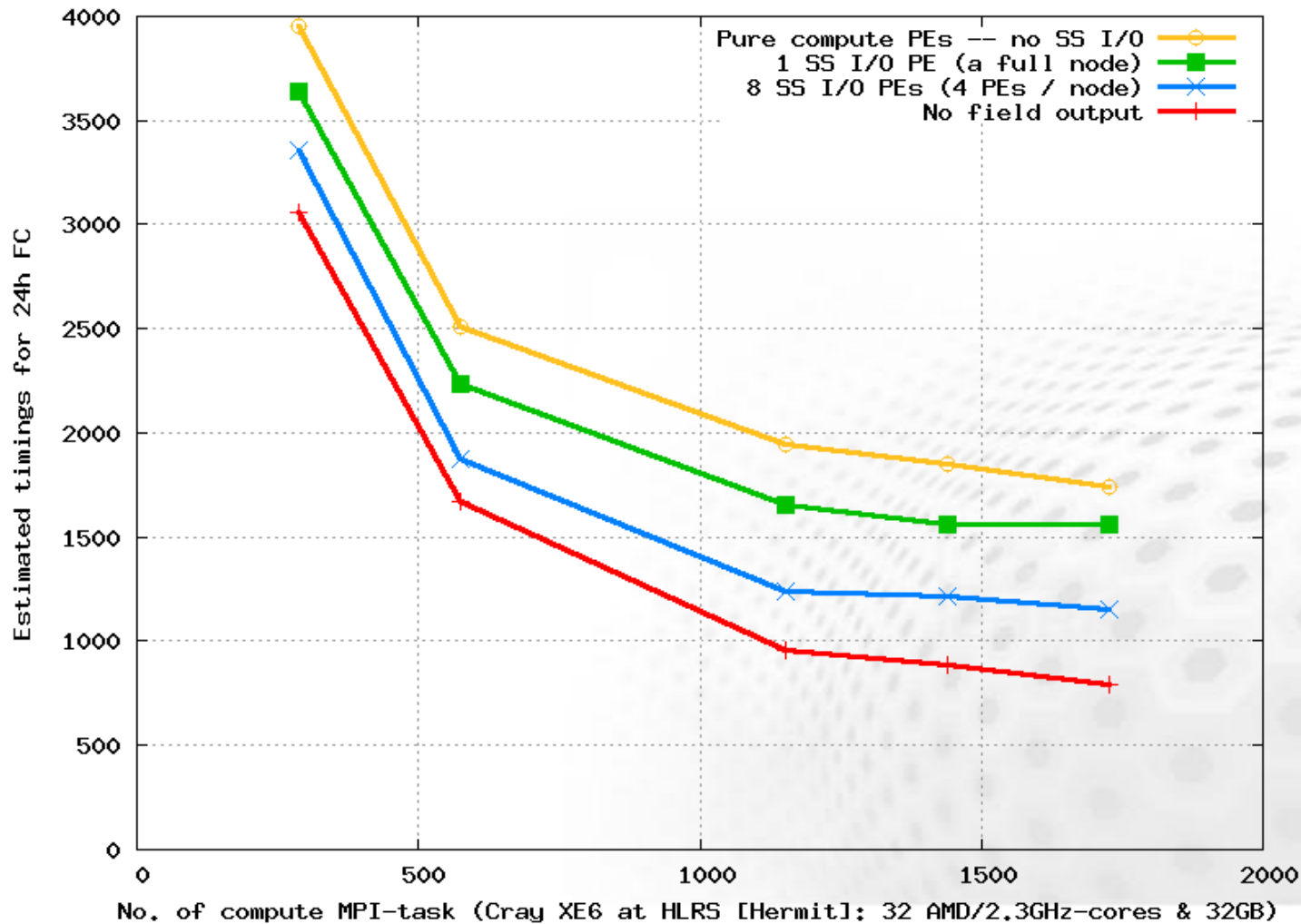
Preliminary results

- ➊ A test-suite with model size 512 x 600 x L60 was used
 - Field output to FA-files every 15th time-step
 - No SURFEX output, no radiation scheme on
- ➋ On Cray XE6 at HLRS (Hermit), Germany
 - AMD Interlagos 2.3GHz, 32 cores/node, 32GBytes/node
- ➌ Using SS I/O with 1, 2, 4 and 8 PE-sets each having just one MPI-task, but occupying up to 2 extra full XE6-nodes
 - sacrificed up to 64 cores extra (<< compute cores)

Estimated 24h FC timings (est. from 6h FC)

Compute MPI-tasks	Without fld output	Std Run, no SS I/O	Number of SS I/O sets, 1 MPI-task / set			
			1	2	4	8
288	3055	3951	3638	3232	3171	3355
576	1674	2510	2235	2077	1856	1871
1152	960	1948	1656	1528	1356	1236
1440	884	1849	1563	1450	1235	1218
1728	789	1745	1562	1391	1198	1155

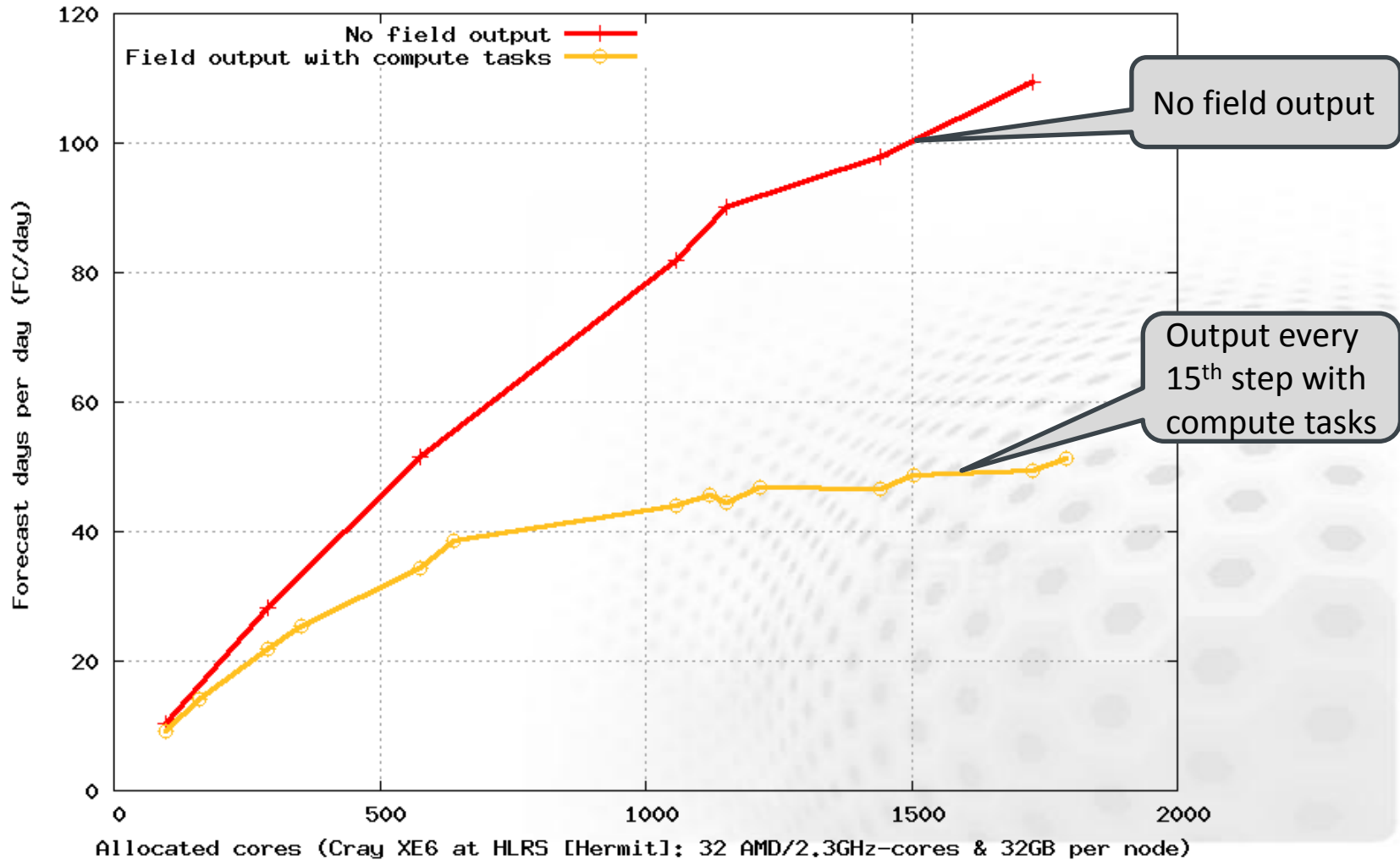
Harmonie/AROME pre-CY38 with sub-space (SS) I/O at 512 x 600 x L60



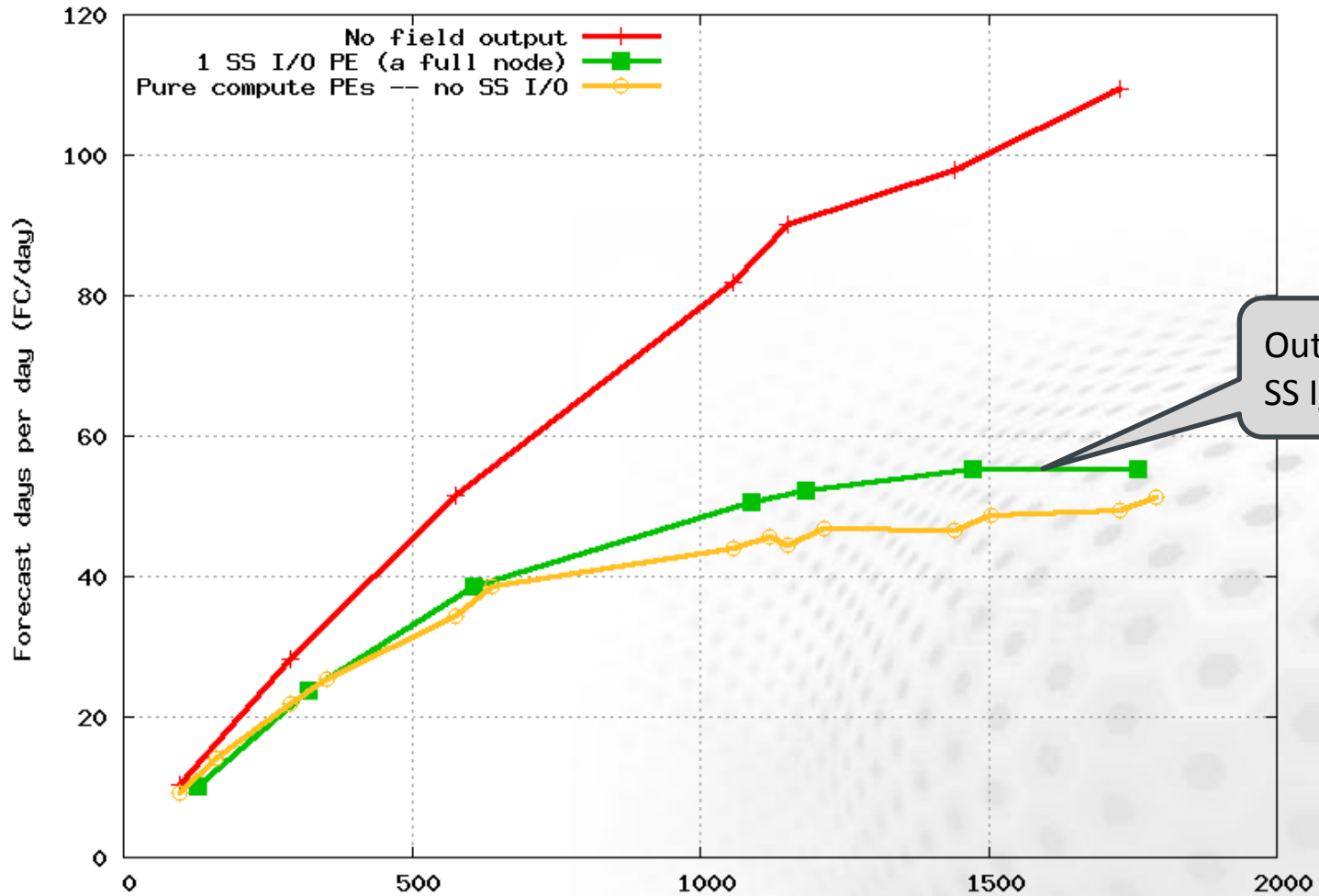
Forecast days per day

A decorative background on the right side of the slide. It features a grid of dots that transitions into a pattern of hexagons, creating a sense of depth and perspective. The colors are light and muted, blending into the white background.

Harmonie/AROME pre-CY38 with sub-space (55) I/O at 512 x 600 x L60



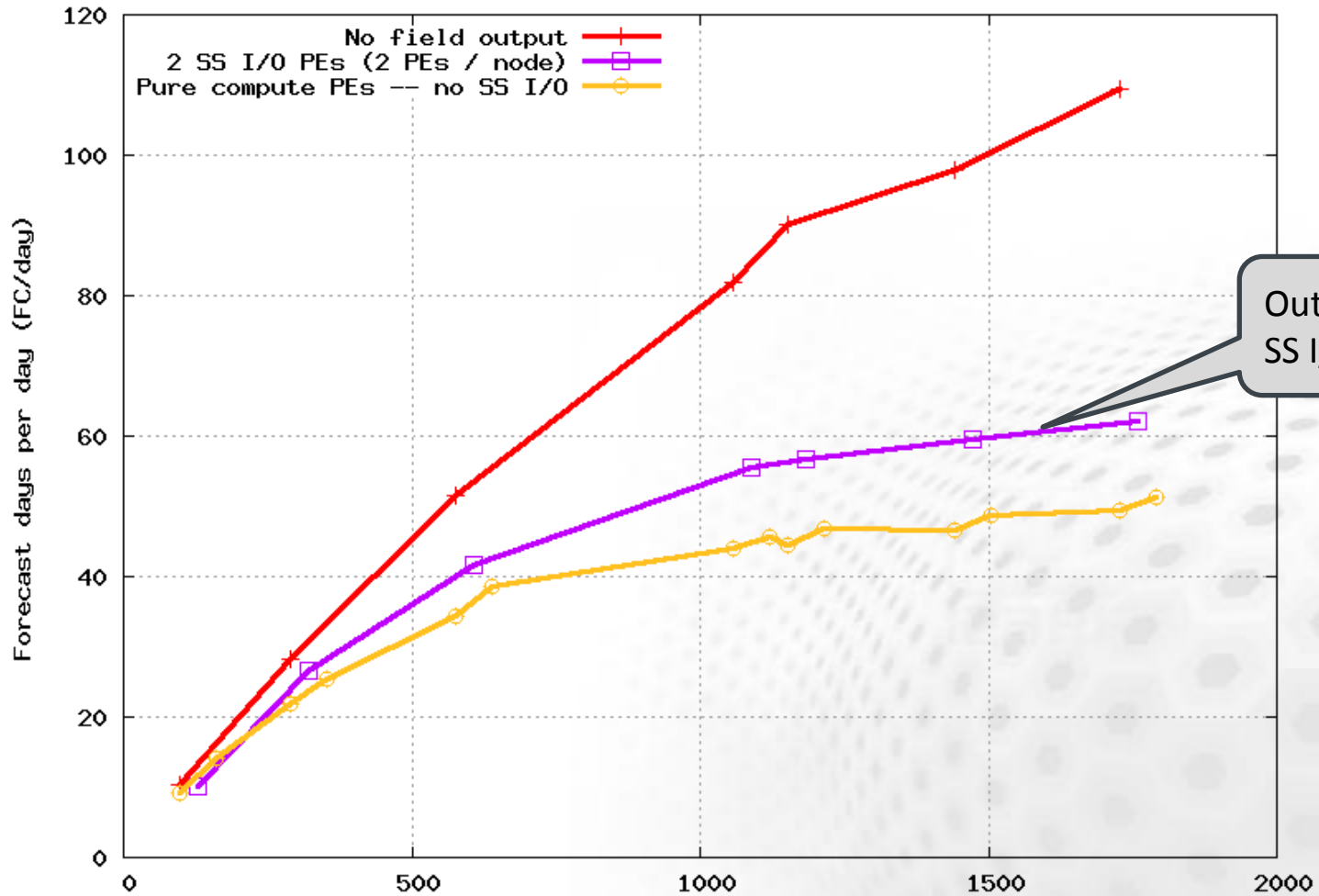
Harmonie/AROME pre-CY38 with sub-space (SS) I/O at 512 x 600 x L60



Output with 1 SS I/O PE-set

Allocated cores (Cray XE6 at HLRS [Hermit]: 32 AMD/2.3GHz-cores & 32GB per node)

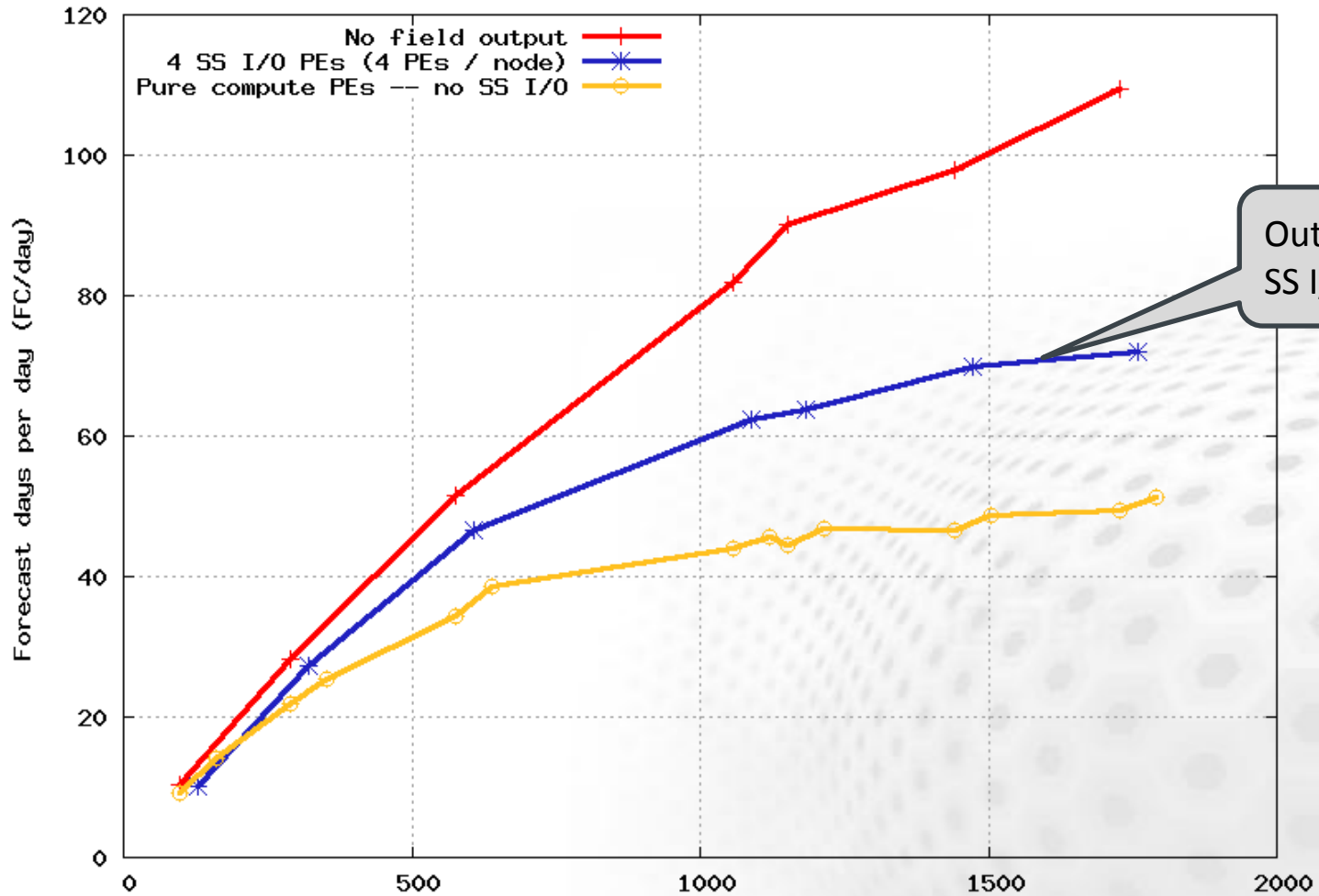
Harmonie/AROME pre-CY38 with sub-space (SS) I/O at 512 x 600 x L60



Output with 2 SS I/O PE-sets

Allocated cores (Cray XE6 at HLRS [Hermit]: 32 AMD/2.3GHz-cores & 32GB per node)

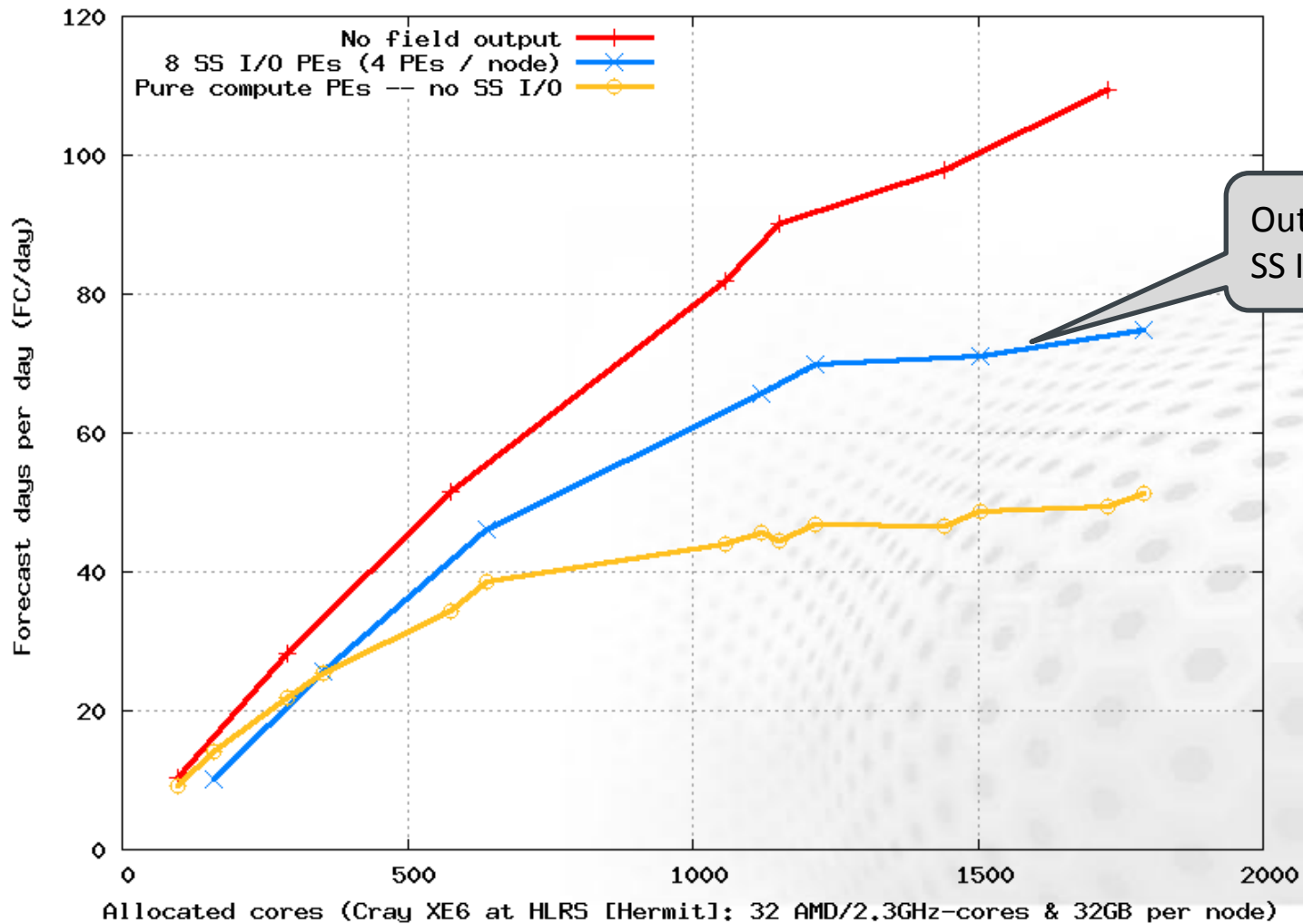
Harmonie/AROME pre-CY38 with sub-space (SS) I/O at 512 x 600 x L60



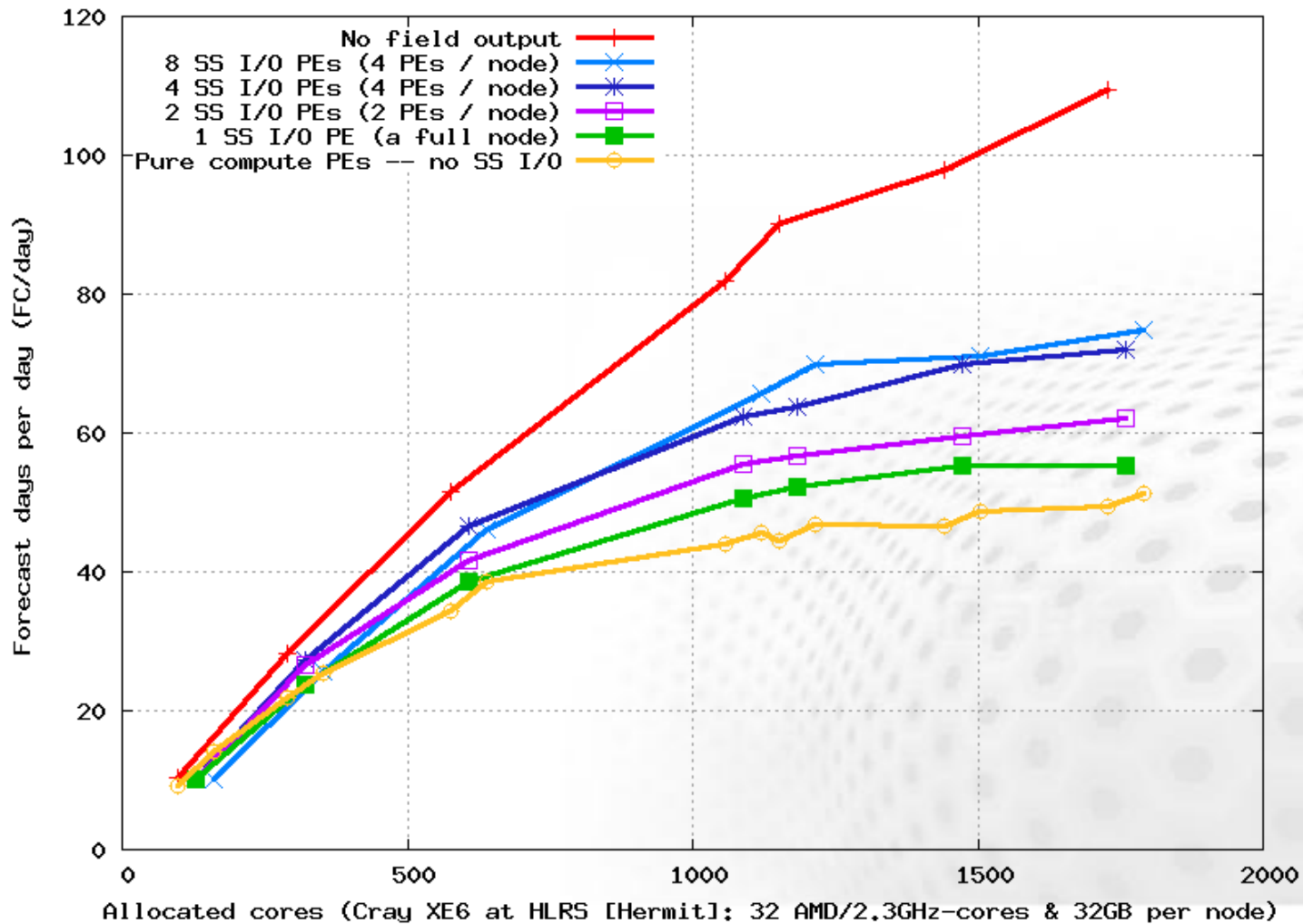
Output with 4 SS I/O PE-sets

Allocated cores (Cray XE6 at HLRS [Hermit]: 32 AMD/2.3GHz-cores & 32GB per node)

Harmonie/AROME pre-CY38 with sub-space (SS) I/O at 512 x 600 x L60



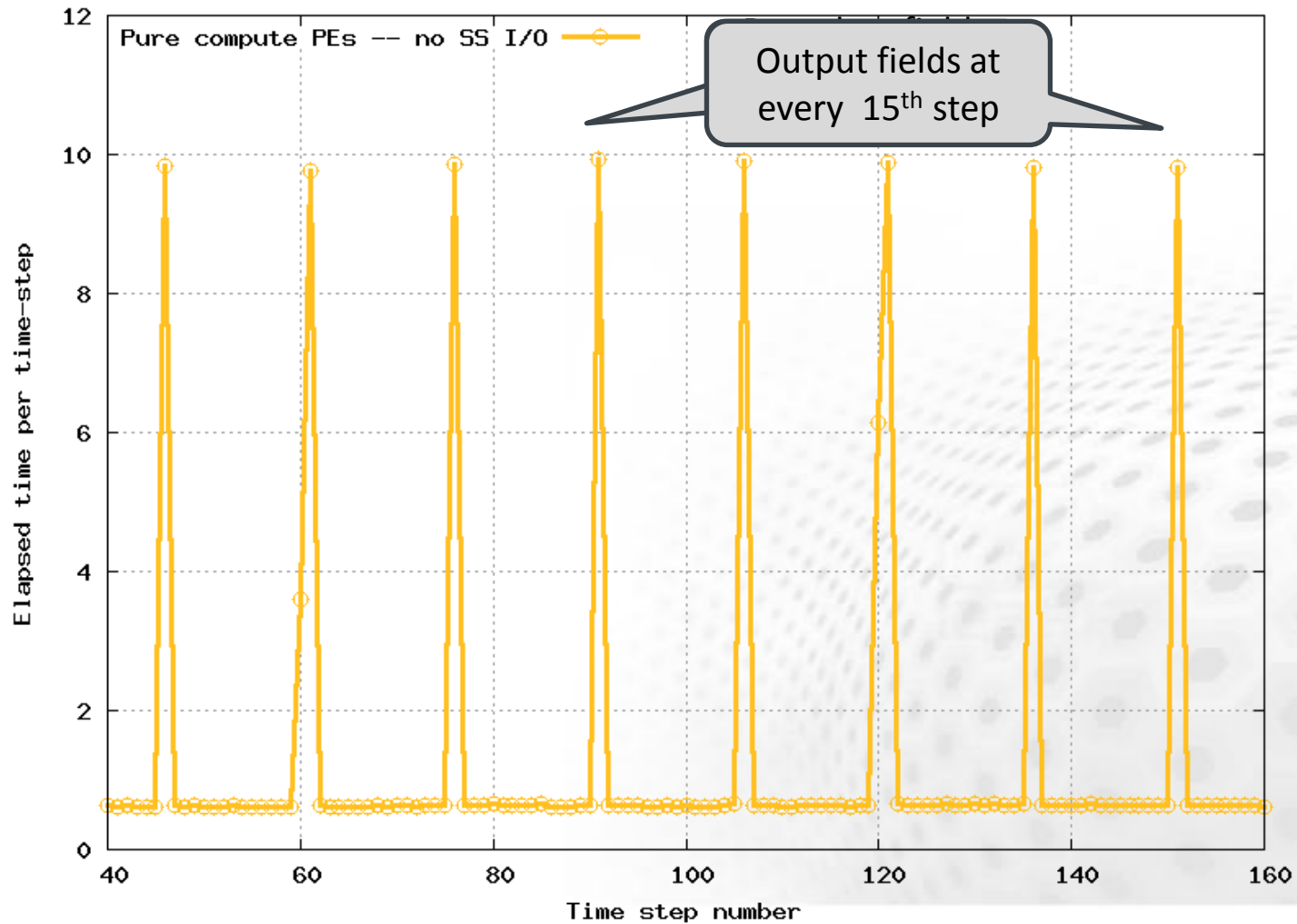
Harmonie/AROME pre-CY38 with sub-space (SS) I/O at 512 x 600 x L60



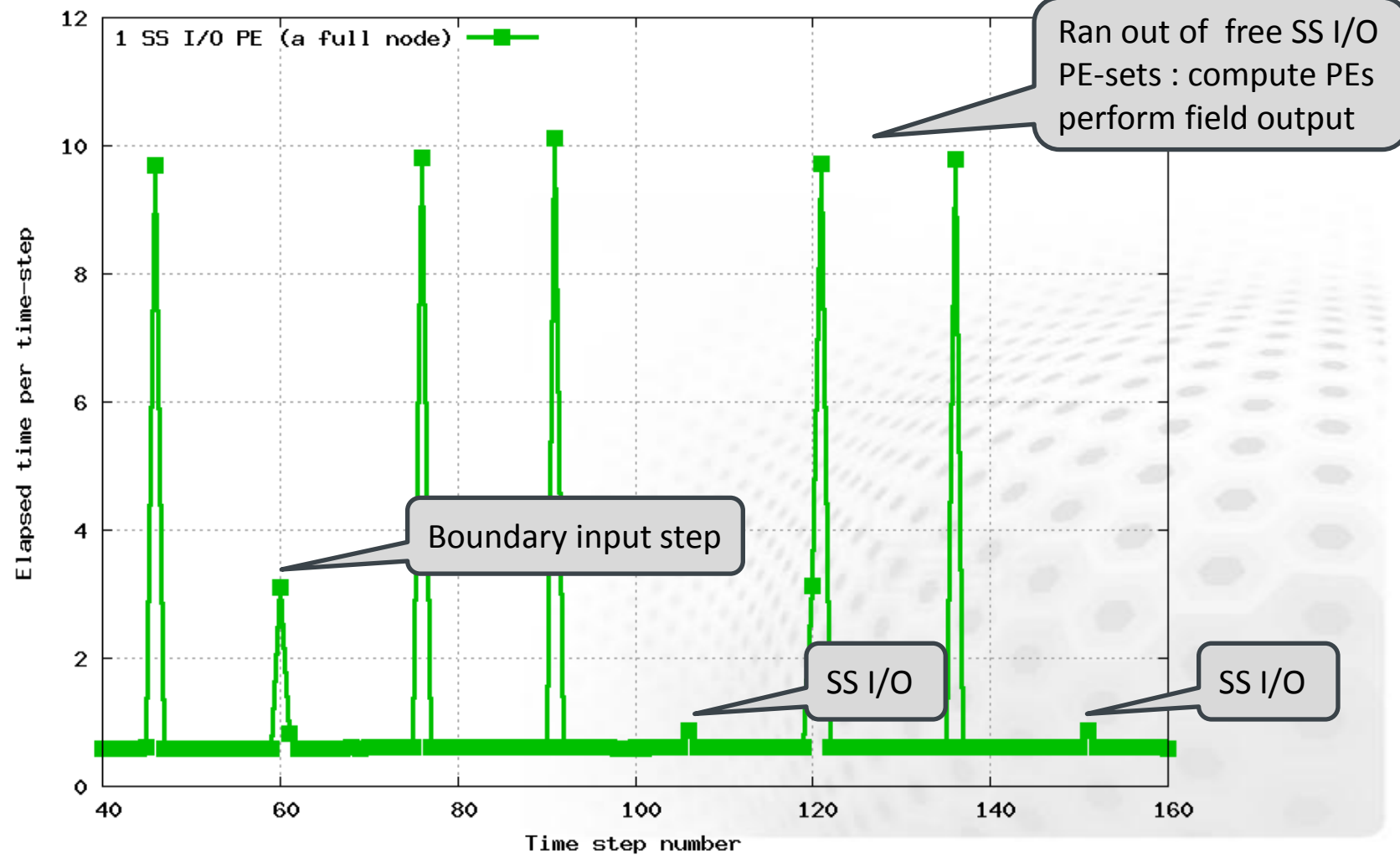
Elapsed time per time-step

A decorative background graphic on the right side of the slide. It features a grid of small, light gray circles that form a perspective effect, receding into the distance. The circles are arranged in a pattern that suggests a 3D grid or a surface with a repeating pattern.

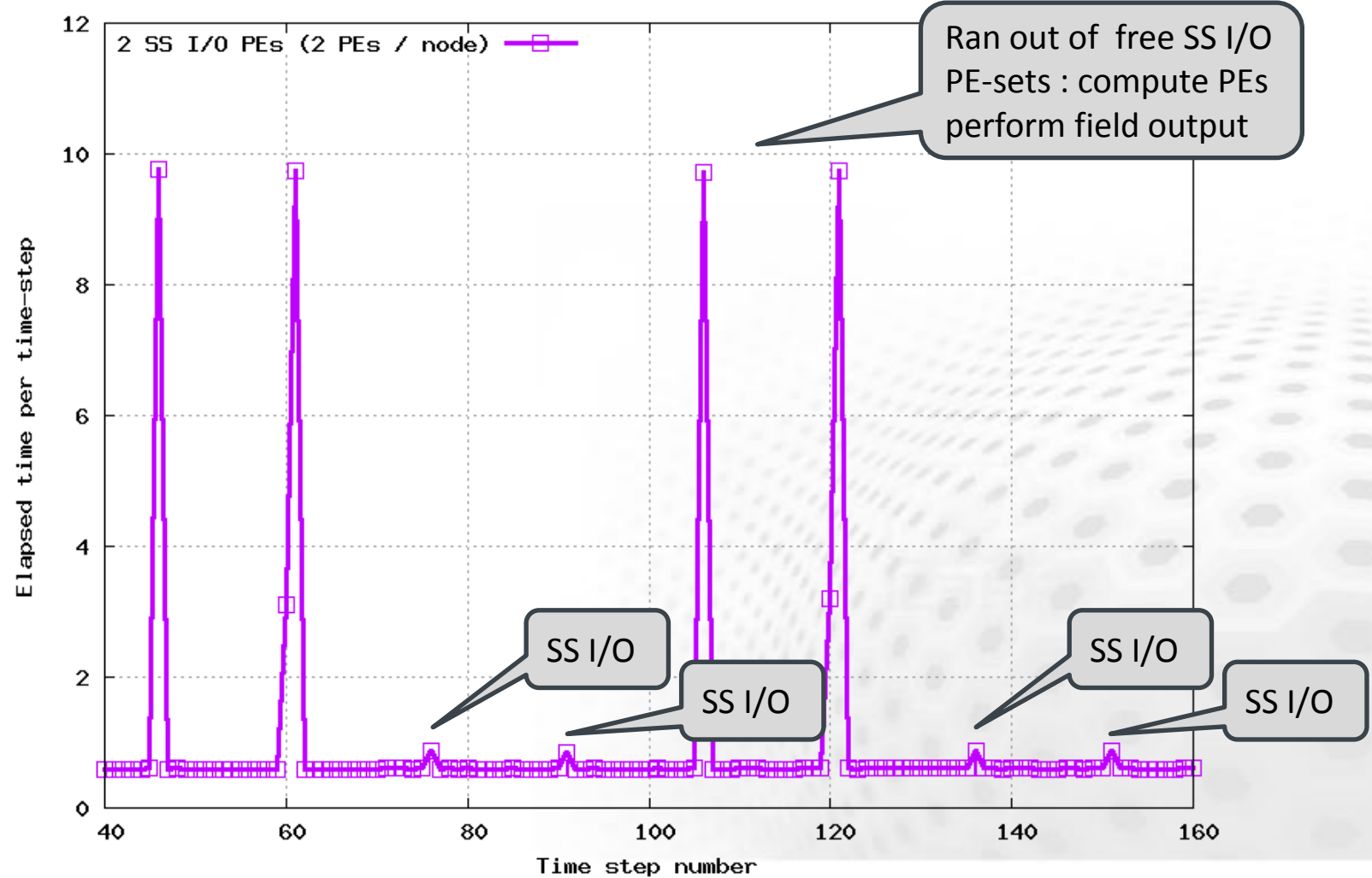
Elapsed time per time-step (at 1152 compute tasks)



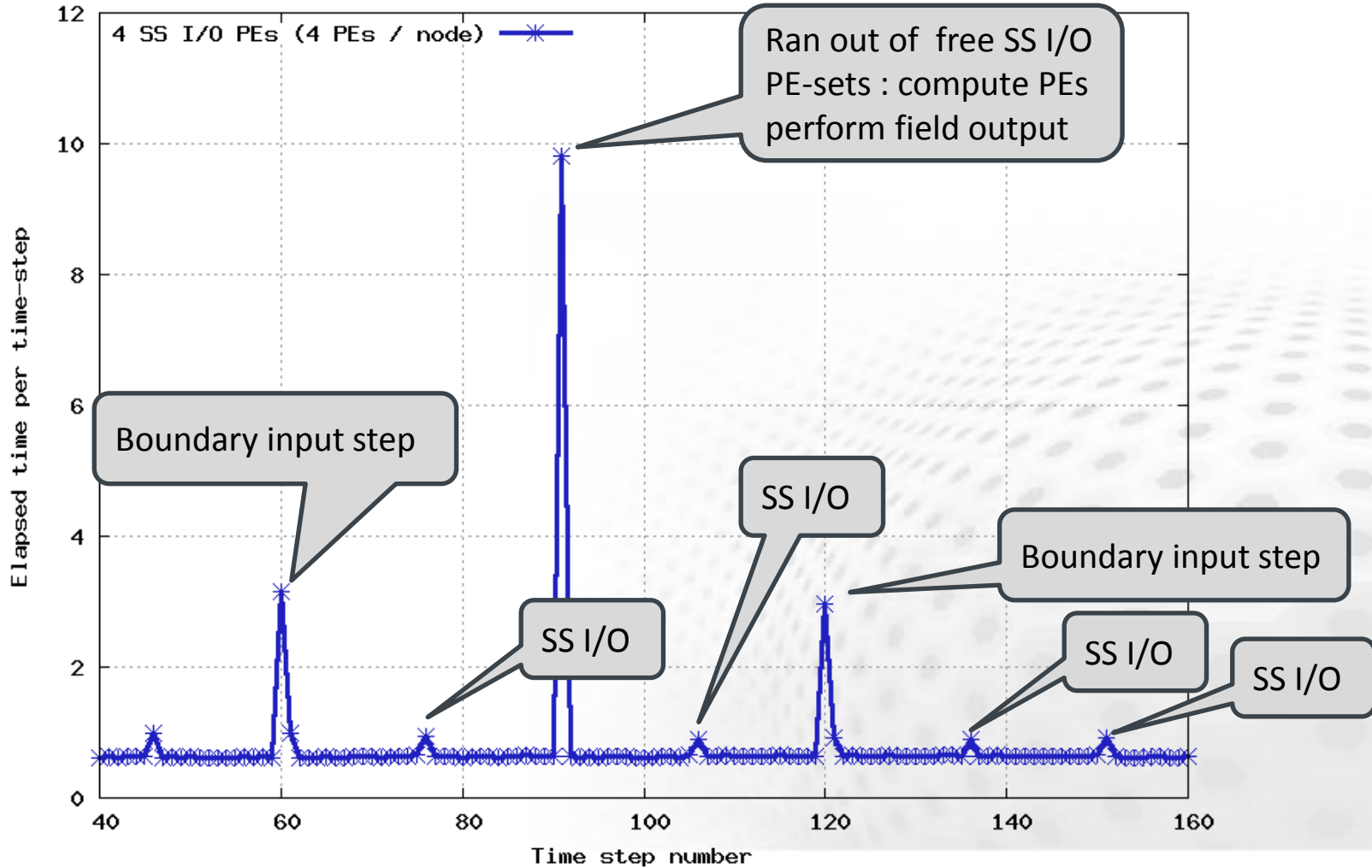
Elapsed time per time-step (at 1152 compute tasks)



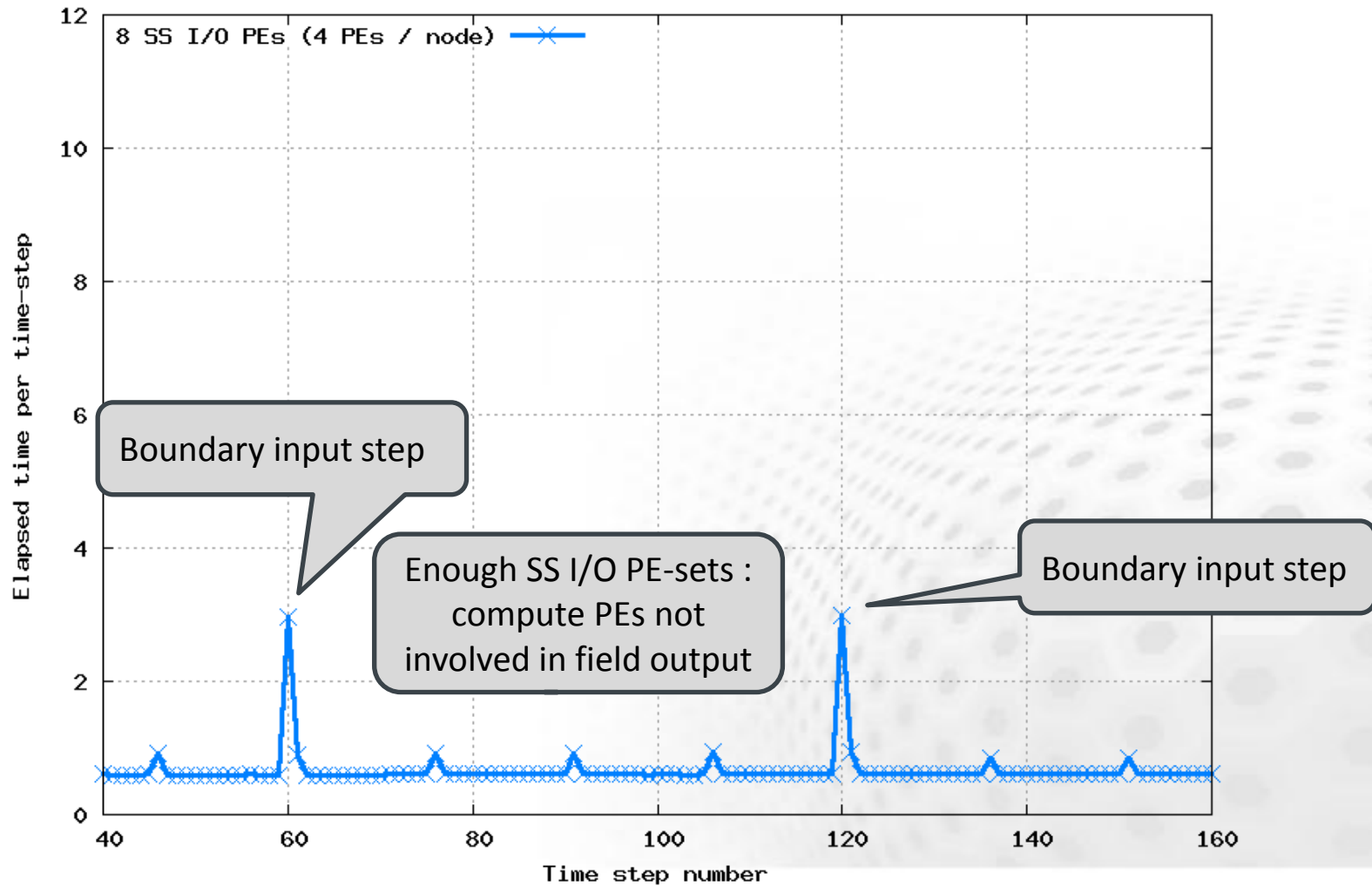
Elapsed time per time-step (at 1152 compute tasks)



Elapsed time per time-step (at 1152 compute tasks)



Elapsed time per time-step (at 1152 compute tasks)



Summary

- Frequent field output destroys parallel scalability – and yet it has to be provided for smooth forecast animations
- Offloading output to a sub-space of I/O-tasks using passive one-sided MPI-communication releases compute tasks from I/O-burden and greatly improves scalability
- The SS I/O is especially suitable for limited area models

Supplementary slides

A decorative graphic on the right side of the slide, consisting of a grid of small, light gray hexagons that recede into the distance, creating a sense of depth and perspective.

Implementation details

- Compute PEs and SS I/O-tasks initialize MPI-processing together and afterwards I/O-tasks are put in a wait loop
- SS I/O-tasks are divided into so PE-sets (or I/O-clusters), each of them containing one or more MPI-tasks
- The master task of the compute PEs have a common communicator with each task in a PE-set
- The default communicator within compute PEs operates with compute task only → no code modifications
- Each PE-set also uses default comm. within its I/O-cluster

Implementation details (cont'd)

- Upon opening field output file, the master compute task checks whether next I/O-cluster is available – or too busy
 - If available, then each compute task places copy of its local contribution to GP and SP-fields into its one-sided communication buffer (a local memory copy) and notifies the master task of the I/O-cluster in duty
 - If the I/O-cluster is busy, then I/O gets processed in traditional synchronized way : using compute tasks only
- Implemented routines : WRSPECA and WRGP2FA

Implementation details (cont'd)

- A typical configuration has perhaps 2 .. 8 SS I/O-sets and each set (i.e. cluster) usually has only one MPI-task (so called master I/O PE – per cluster)
- We can allocate the whole SMP-node (or parts of it) for SS I/O's disposal – e.g. 32 cores on Cray XE6 at HLRS
 - Such MPI-task is also available for OpenMP, e.g. 32-way 😊
- This allocation strategy guarantees a plenty of memory for the current limited area model resolutions
 - We have run successfully up to 1600 x 1600 x L65