DSM2 performance improvements

Nicky Sandhu

State of affairs: DSM2

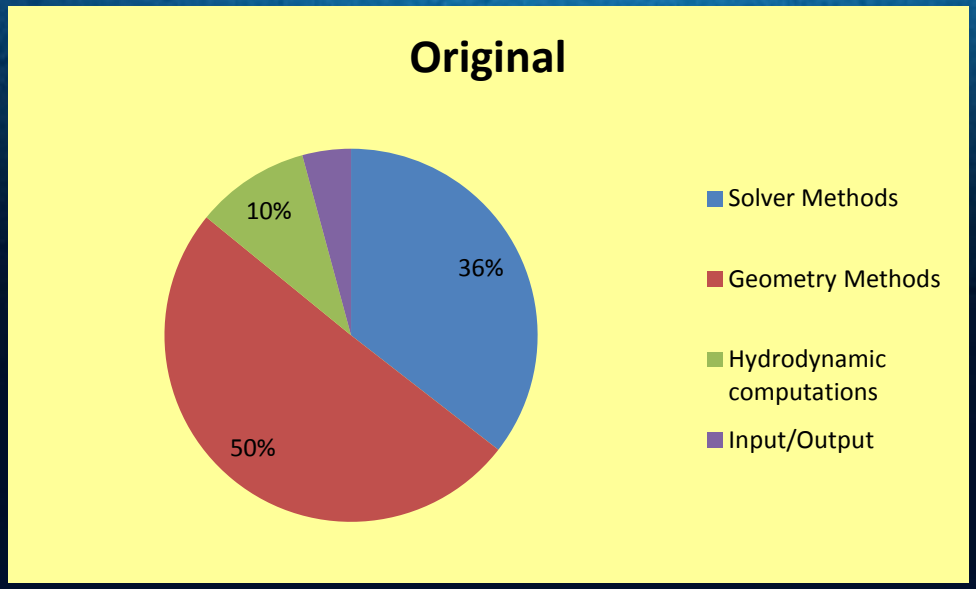
- Hydro (slowest) by a factor of 4 w.r.t Qual
- Qual
- PTM

CodeAnalyst

- A hammer in need of a nail…
- Tool from AMD (Intel's only competitor)
- Profiles code to method and line level

Initial state

- Geometry calculations 50% runtime !
- Solver 33% runtime
- Hydrodynamic calculations 10% runtime

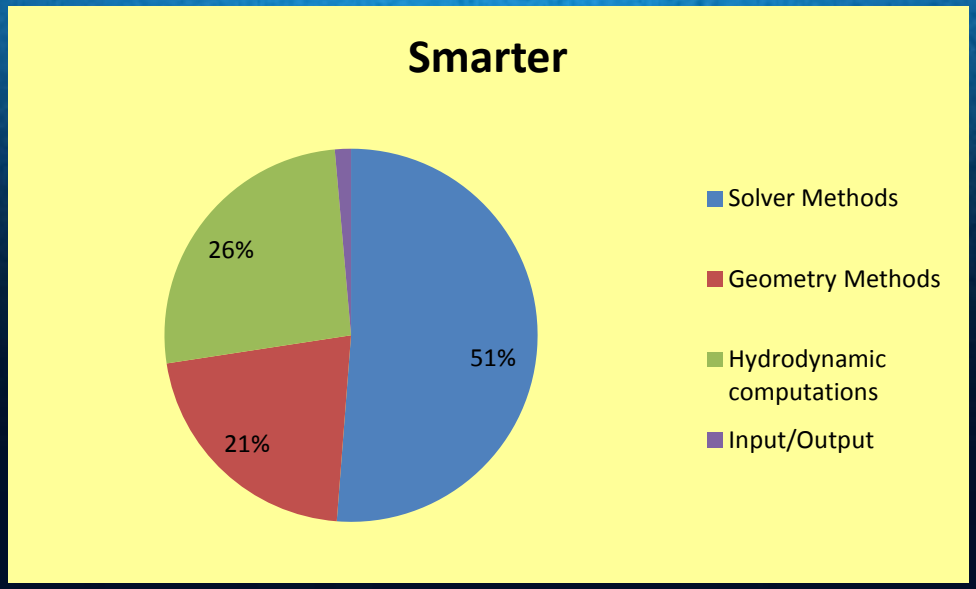


Modifying for performance

- Eliminate use of power function
  - Use more efficient cuberoot instead
  - Reorganize conveyance calculations to evaluate cuberoot only once
- Eliminate repeated redundant calls to geometry calculations
  - Faster
  - More accurate (no loss due to round off)

Performance snapshot

- Speed up of approximately x2
- Solver now dominating runtime > 50%



Research on solvers

- Saved a number of sample matrices from a sample run
- Fed to multiple sparse solvers
  - Sparse 1.3 (current Berkeley solver from Kundert)
  - PARADISO (Fastest parallel sparse solver)
  - KLU (Solver from Tim Davis, has replaced Sparse in circuit simulation solver packages)

…and the fastest of them all is

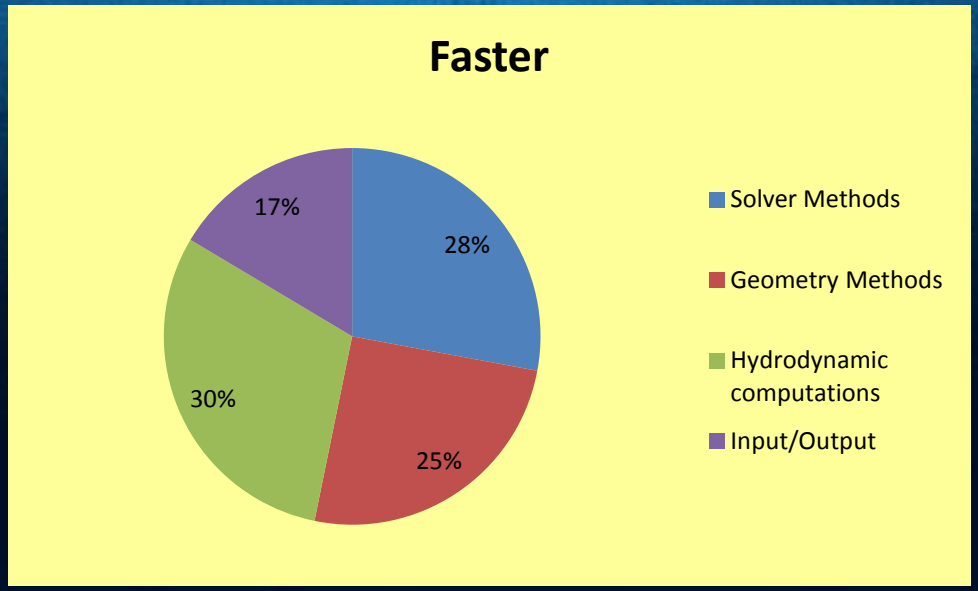
- KLU
  - 3x times faster than Sparse
  - 6x times faster than PARADISO
- Notes
  - Our problem is small (2500x2500 size)
  - Parallelization doesn't make sense as overhead is large. That is why PARADISO is not the fastest

Scaling

- Iterative multiple solves converges in about 3 iterations
- Scaling the matrix slightly differently leads to faster convergence
- Converges in about 2 iterations/time step
- Total number of solver calls reduced by 30%
- Lower number of max iteration hits as well

Final performance snapshot

- Faster than original by x4
- Equitable distribution of calculation times between different parts of the computations



What this all means

- Hydro is now faster than Qual and 4 times faster than before performance changes
- Results are more accurate (cube root vs power and superior solver). Change is less than 0.001%
- A 16 year combined hydro and qual run which used to take nearly 4 hours now runs in less than an hour.
- Speed kills…so drive carefully ☺

Appendix: Profile from CodeAnalyst

Original		Eliminating redundant calculations		Faster solver	
Symbol + Offset	Timer samples	Symbol + Offset	Timer samples	Symbol + Offset	Timer samples
RealRowColElimination	14.15	spOrderAndFactor	23.16	KLU_FORTRAN_REFACTOR	12.4
FIND_LAYER_INDEX.	14.12	spSolve	8.13	DYNAMICWAVEEQ	9.21
__libm_sse2_pow	11.4	CALCULATECHANNELGEOMETRYASPECTS	7.94	CALCULATECHANNELGEOMETRYASPECTS	8.43
DCONVEYANCE	6.52	DYNAMICWAVEEQ	7.93	_aulldvrm	4.99
spSolve	5.43	spClear	7.68	cbrt.J	4.39
CONVEYANCE	5.05	spScale	6.52	for_index	3.61
spClear	4.48	cbrt.J	4.11	FIND_LAYER_INDEX.	3.54
spScale	4.23	FIND_LAYER_INDEX.	3.4	SOLVEFOURPT	2.99
DYNAMICWAVEEQ	4	for_index	3.02	szip_compress_memory	2.98
CXAREA	3.87	UPDATENETWORK	2.97	UPDATENETWORK	2.96
FindLargestInCol	3.04	SOLVEFOURPT	2.52	klu_usolve.	2.84
CHANNELWIDTH	2.28	for_cpstr	2.16	KLU_mp_SCALE_COO	2.59
spOrderAndFactor	1.47	_intel_fast_memcmp	1.41	KLU_mp_ADD_TO_MATRIX	2.15
SOLVEFOURPT	1.4	NINDXR	1.31	klu_lsolve.	2.12
_aulldvrm	1.36	READTVD	1.29	READTVD	2.04
for_index	1.25	SFADD4EQUATION	1.02	for_cpstr	1.37
szip_compress_memory	1.97	FORCESUMSTREAMFLOW	0.94	H5S_hyper_denormalize_offset	1.34
CHECKCHANNELCOMPLOCATIONRANGE	0.79	SFADD1REAL	0.91	_aulldiv	1.21
NETWORKCLOSURE	0.68	__intel_memset	0.83	__intel_memset	1.19
SFADD4EQUATION	0.64	FORCEEQUALSTREAMSURFACE	0.58	WRITEHYDROTOTIDEFILE	1.16
READTVD	0.52	STOREATLOCATION	0.57	FORCESUMSTREAMFLOW	1.08
FORCESUMSTREAMFLOW	0.52	SETSTREAMSURFACEELEVATION	0.53	_intel_fast_memcmp	0.99
BTMELEV	0.5	CHECK_DATAQUAL.	0.44	FORCEEQUALSTREAMSURFACE	0.77
SFADD1REAL	0.43	UPSTREAMPOINTER	0.41	H5S_select_iterate	0.66
for_cpstr	0.43	STREAMDISTANCE	0.39	H5V_chunk_index	0.58
STREAMFLOW	0.4	STREAMFLOW	0.39	H5V_array_calc	0.58
SETSTREAMSURFACEELEVATION	0.38	STREAMENDNODE	0.37	H5V_array_offset_pre	0.57
_aulldiv	0.35	OPENCHANNEL	0.36	CHECK_DATAQUAL	0.56
UPDATECHANNEL	0.34	STREAMSURFACEELEVATION	0.33	SETSTREAMSURFACEELEVATION	0.55
WRITEHYDROTOTIDEFILE	0.34	GET_INP_DATA	0.33	_allmul	0.51
H5S_hyper_denormalize_offset	0.33	STREAMDEPTH	0.32	klu_solve.	0.5
STREAMSURFACEELEVATION	0.33	FORWARDELIM	0.26	GET_INP_DATA	0.49
STREAMDISTANCE	0.33	INCREMENTALQ	0.25	STREAMDISTANCE	0.49
STREAMDEPTH	0.32	ADDATROW	0.23	NINDXR	0.49
UPDATENETWORK	0.29	SETBOUNDARYVALUESFROMDATA	0.2	H5V_hyper_copy	0.48
FORCEEQUALSTREAMSURFACE	0.27	GLOBALSTREAMSURFACEELEVATION	0.2	STOREATLOCATION	0.48