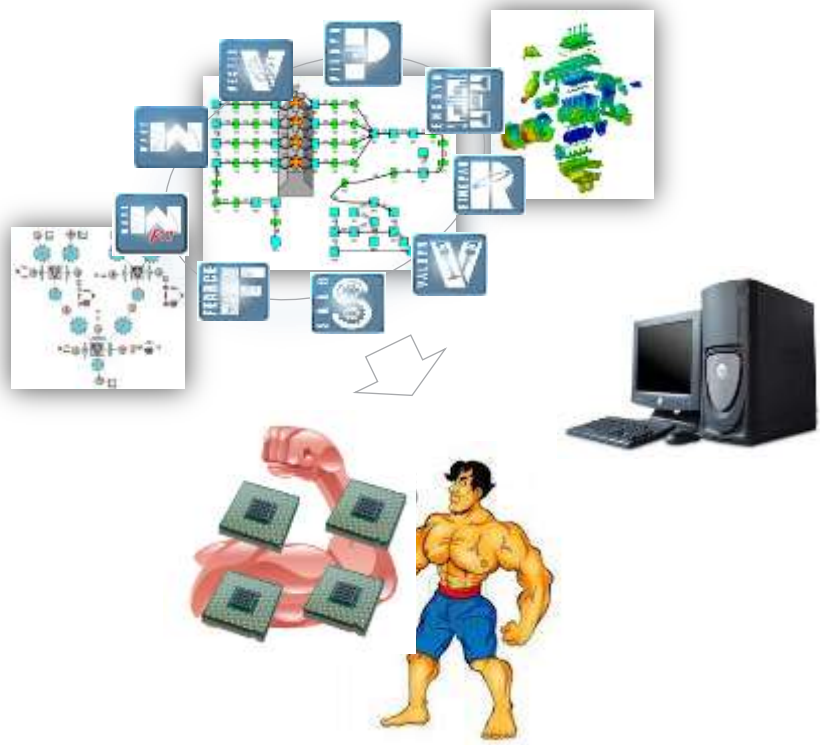
Distributed Running (RDM) and Parallelization with Ricardo Software

This short presentation presents advantages of distributed running together with parallelization for various Ricardo software codes leading for significant analyses speed up useful for both common analyses and DoE studies



Date **15th January 2013**

Report by **Jiri Navratil, Nick Tiney, Jonathan Plail, Patrick Niven, David Bell, Iain Robertson and Dave Higbie**

Summary

- Aim:
  - The main aim of this presentation is to show the huge speed up potential by using distributed (RDM) and parallel running for common analyses and DoE studies and understanding the how it works
  - The second aim is to show the expected speed up for chosen sample cases which can be used by the commercial team for recommending the proper licencing packages customers
- Method:
  - Distributed running: RDM_4, RDM_8, RDM_16, RDM_32, RDM_Unlimited (WAVE and VALDYN)
  - Parallel running: (VECTIS, WAVE, ENGDYN and FEARCE)
- Conclusions:
  - Distributed run presents very efficient way for analysis run time speed up (WAVE, VALDYN)
    - Up to **7** times faster run time for WAVE performance analysis and VALDYN with 15 cases RDM_16
    - Up To **14** times faster run time for WAVE DoE with 4 variables (typically 256 cases) with RDM_16
  - Parallel run (VECTIS, WAVE and ENGDYN)
    - Essential for VECTIS where it is used for mesh dividing into multiple CPUs with **4-11** time faster calculation time reduction when using 16 cores
    - Efficient for special WAVE cases e.g. transient simulations or acoustic where running at maximum **4** CPUs can lead to **2** x faster run time
    - Up to **2** x faster run with 2 CPUs running FEARCE for chosen analysis
    - Up to **1.8** speed factor runtime with 2 CPUs running ENGDYN (dependent on model/solution type)

Agenda

- **Introduction**
- Distributed Running (RDM)
- Parallel Running
- Examples & Recommendations
- Conclusions

There are 2 basic possibilities for analysis speed up – Distributed and Parallel Run

- It is important to understand how both approaches work to select the correct one for chosen code and analysis type
- Distributed run is efficient for analyses with many calculation cases
  - Typically one case presents one engine speed
  - The model is split up into cases and calculated in one CPU
  - Recommended for WAVE, VALDYN
- Parallel Run is efficient for analyses with one case where the calculation takes long time
  - Calculation case is distributed or divided into more CPUs
  - Recommended for VECTIS, WAVE, ENGDYN and FEARCE
  - No benefit for PISDYN
  - No parallelization implemented within RINGPAK

Agenda

- Introduction
- **Distributed Running (RDM)**
- Parallel Running
- Examples & Recommendations
- Conclusions

RDM has several licensing options and can be divided into local and cluster distribution

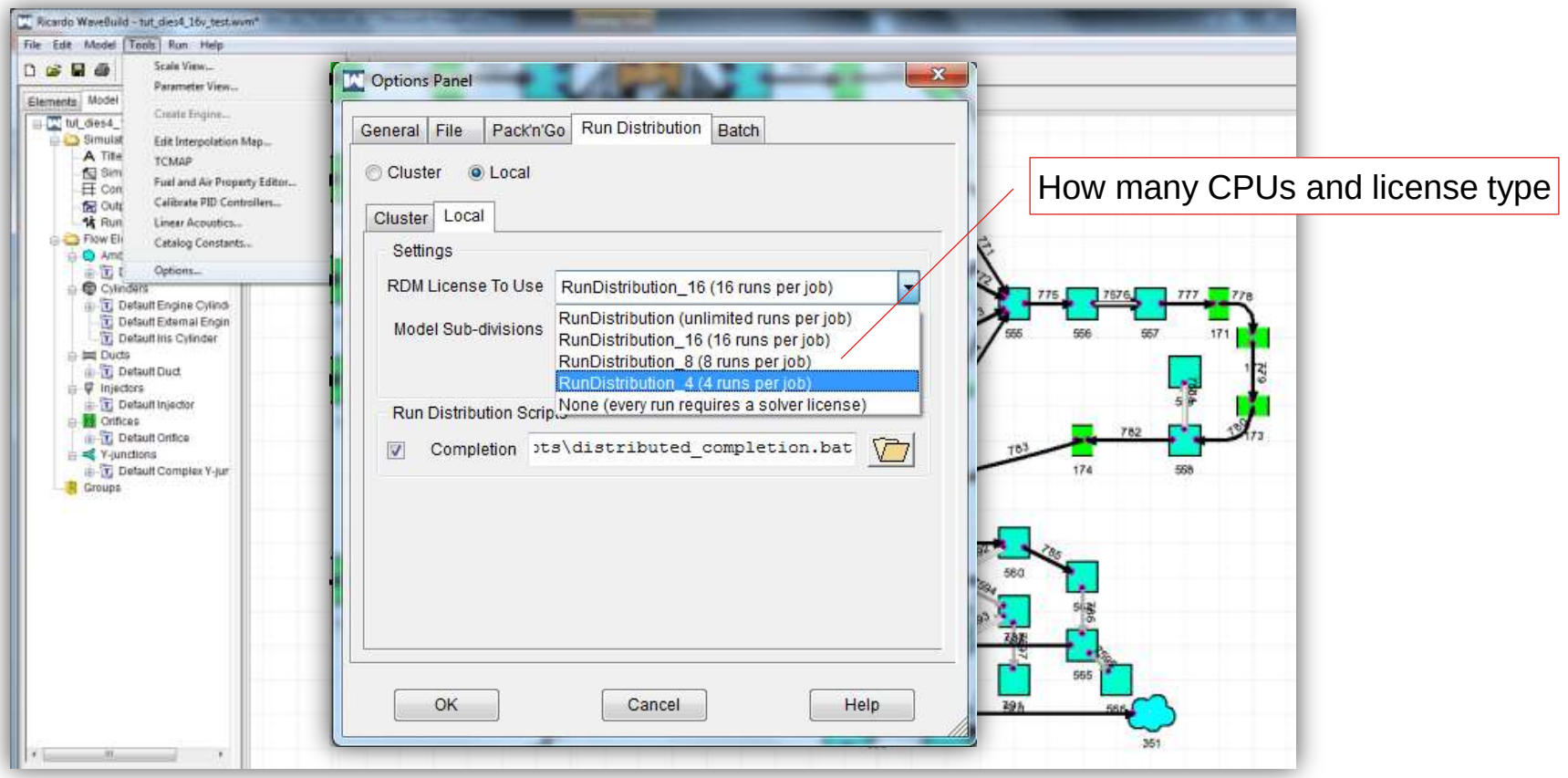
- RDM licensing options:
  - RDM_4: Model is divided into 4 “submodels” and prepared for 4 CPUs distributed run
  - RDM_8: Model is prepared to be run at 8 CPUs
  - RDM_16: Model is prepared for 16 CPUs run
  - RDM_32: Model is prepared for 32 CPUs run
  - RDM_Unlimited: Model is prepared for user selected number CPUs

- Local Run Distribution enables running the cases locally, on a multi-core machine

- Cluster Run Distribution enables sending the cases to a queuing system so they can be submitted to very large clusters with many CPUs / computers
  - Assumes user submitting the divided WAVE model cases to a queuing system
  - Ricardo Software’s ISIS queuing solution can be used “out-of-the-box”, but WAVE Run Distribution can be configured to submit jobs to any commercial queuing system

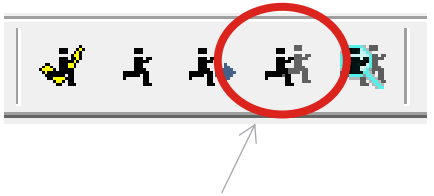
This Example shows the WAVE setting of how the Local distribution is set up within Tools > Options > Run Distribution > Local

- The user can select how many simultaneous CPUs will be used (~ license type)
- WAVE Run Distribution includes 5 types of license options
- Example Completion script provided by Ricardo
  - The user can use this “as is”, edit it to add more functionality, or make their own

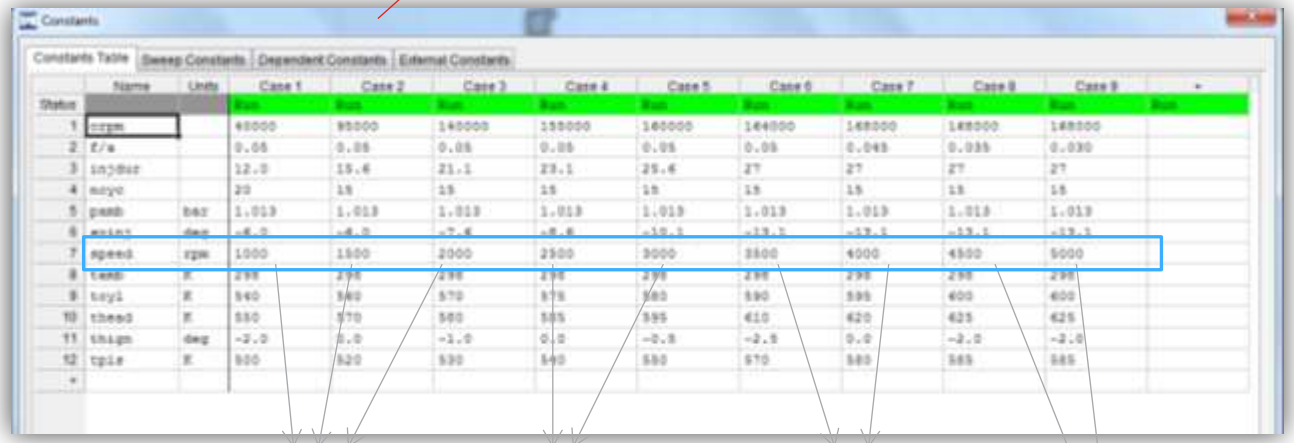


This example shows how the WAVE model with 9 speed cases is divided and run locally on 4 CPUs

- WAVE Run Distribution enables to split model into specific cases and run them in selected number of CPUs (machines)
  - Local Run Distribution starts the jobs directly on the local machine, so doesn't require pre-submit, submit, and post-submit scripts



RDM_4 license: **60%** shorter computation time



Example:
PC with **4 CPUs**

Cases 1, 2 and 3 are run on the 1st CPU

Cases 4 and 5 are run on the 2nd CPU

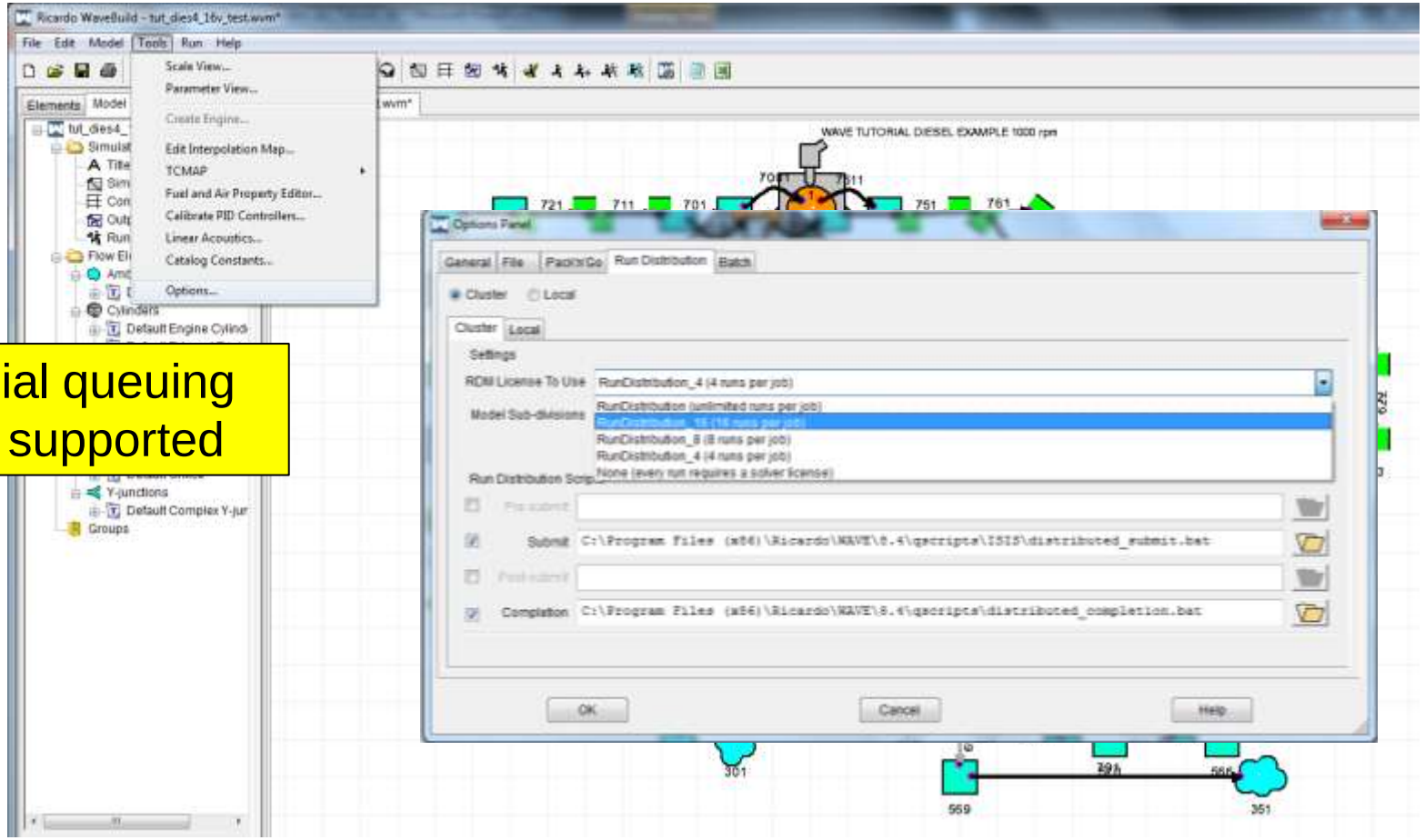
Cases 6 and 7 are run on the 3rd CPU

Cases 8 and 9 are run on the 4th CPU

Cluster distribution is set up within Tools > Options > Run Distribution > Cluster

- User can select how many CPUs can be used (~ license type)
- Wave Distributed includes 5 types of license options
- Batch file for Submit and Completion provided by Ricardo
- ISIS needed

All commercial queuing systems are supported

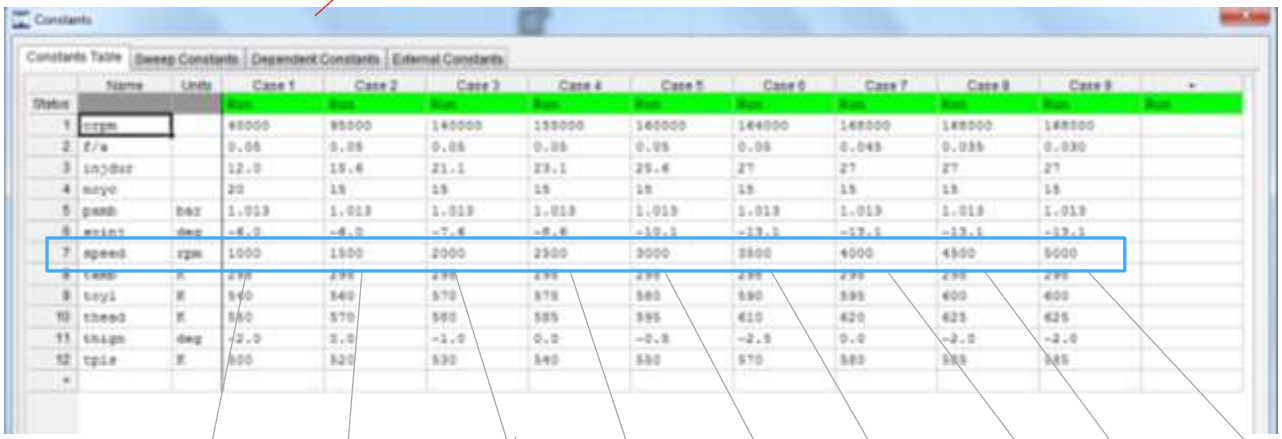


WAVE Cluster Distribution assumes a queuing system is used

- Wave Distribution enables to slit model into specific cases and run them at selected number of CPUs (machines)
  - Cluster Option enables to send cases to cluster with lot of CPUs or more computers
    - This cluster run requires ISIS installation

CTRL + D

RDM_16 license: **85%** shorter computation time



Example:
Cluster with **10 CPUs**

Each case runs separately on one CPU

FREE

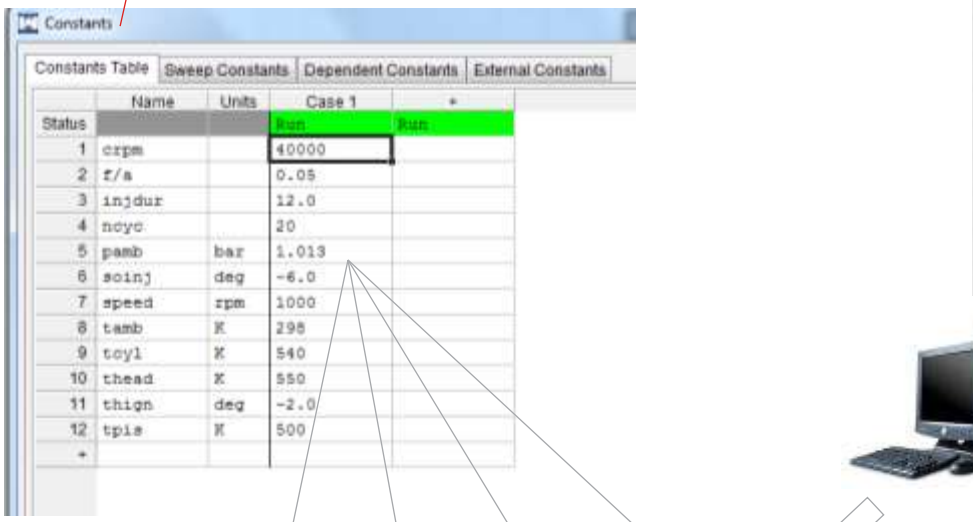
Agenda

- Introduction
- Distributed Running (RDM)
- **Parallel Running**
- Examples & Recommendations
- Conclusions

WAVE Parallel Running provides effective speed up especially for transient runs

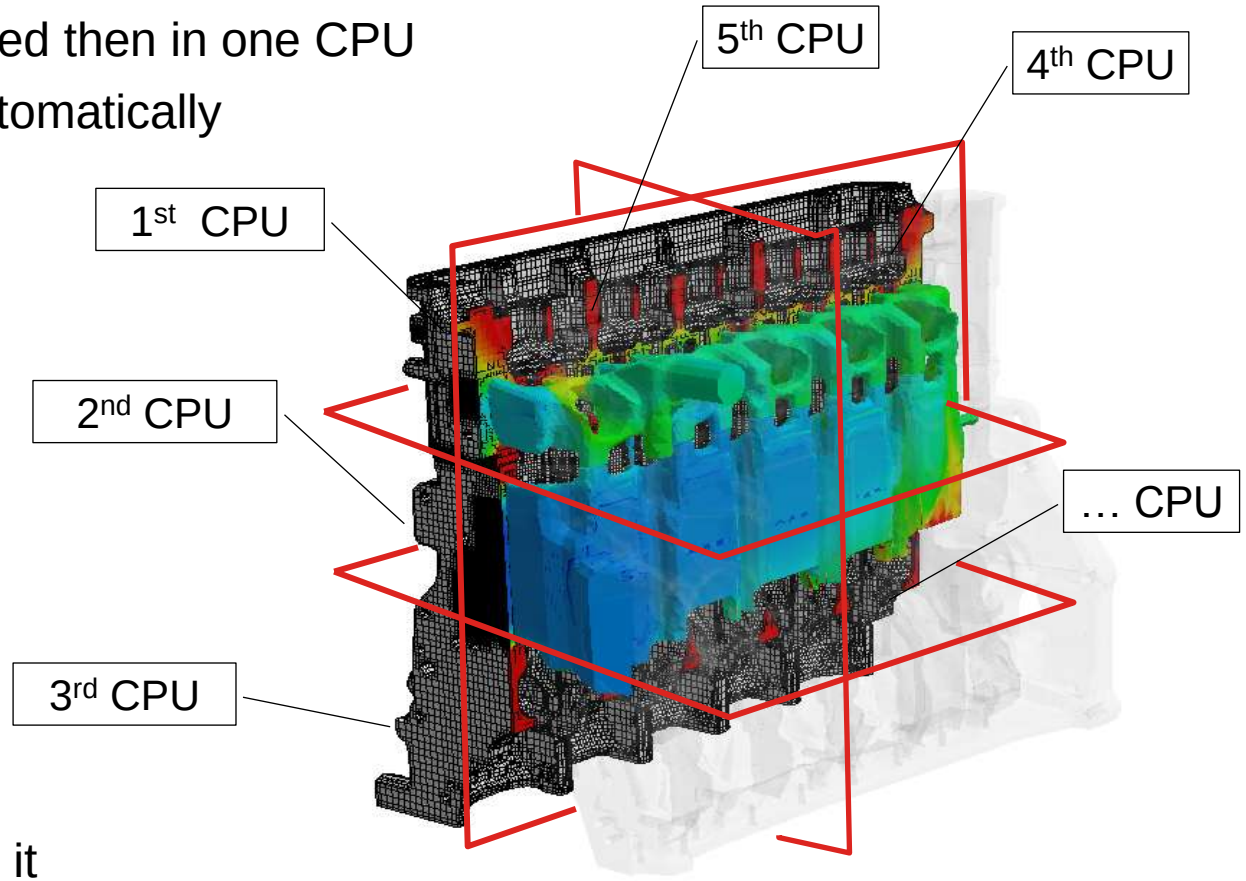
- Parallel running uses more CPUs simultaneously to speed up each case
  - It is beneficial only up to 4 cores !!!
  - It can speed up the calculation significantly
  - Recommend for long transient runs

Parallel License: **50%** shorter computation time



VECTIS is designed / desired to run (parallel) on multiple CPUs

- The model is divided during meshing into several domains
- Every mesh domain is calculated then in one CPU
  - Mesh is divided to CPUs automatically
  - User can influence the split into domains
- Mesh can be fined at selected areas
- The simulation speed up is driven by the conditions at specific domain
  - E.g. if a domain includes small cells or e.g. fuel droplets the other domain calculations have to wait for it since it is the slowest one



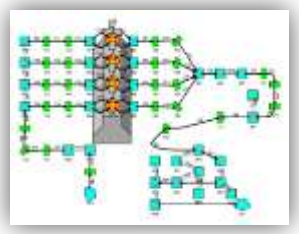
12 Calculation Domains Example

Agenda

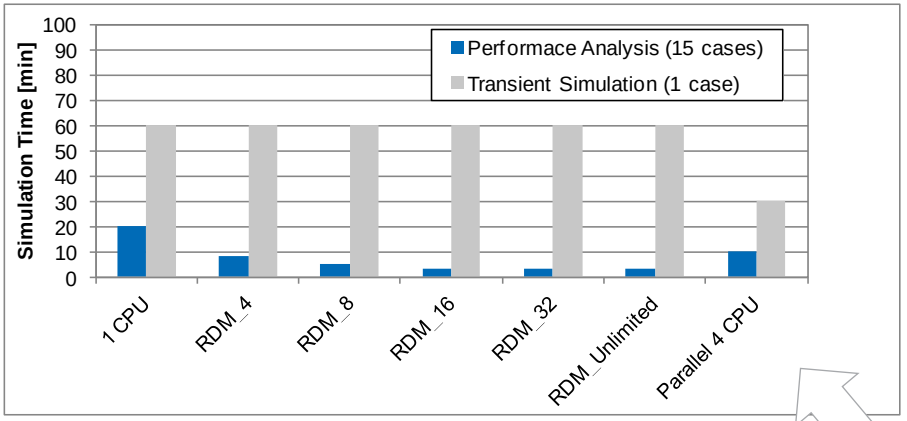
- Introduction
- Distributed Running (RDM)
- Parallel Running
- **Examples & Recommendations**
  - **WAVE**
  - VALDYN
  - VECTIS
  - ENGDYN
  - FEARCE
  - PISDYN/RINGPAK
- Conclusions

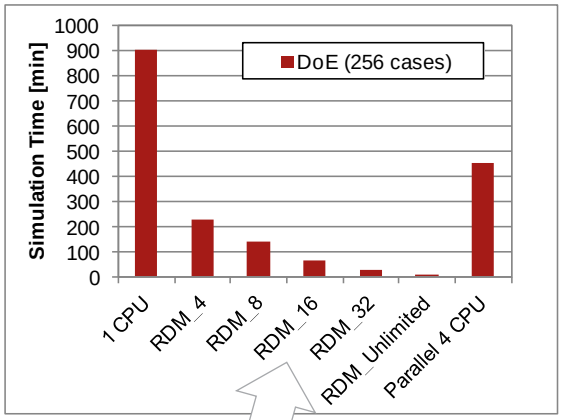
Both RDM and Parallel is supported by WAVE

- This example shows approximate time speed up factors using various RDMs licenses and parallelization



Spark Ignited Turbocharged or Diesel Light Duty Automotive Engines:

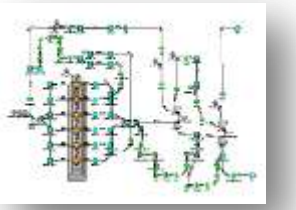




$$Time\ Factor = \frac{20}{3}$$

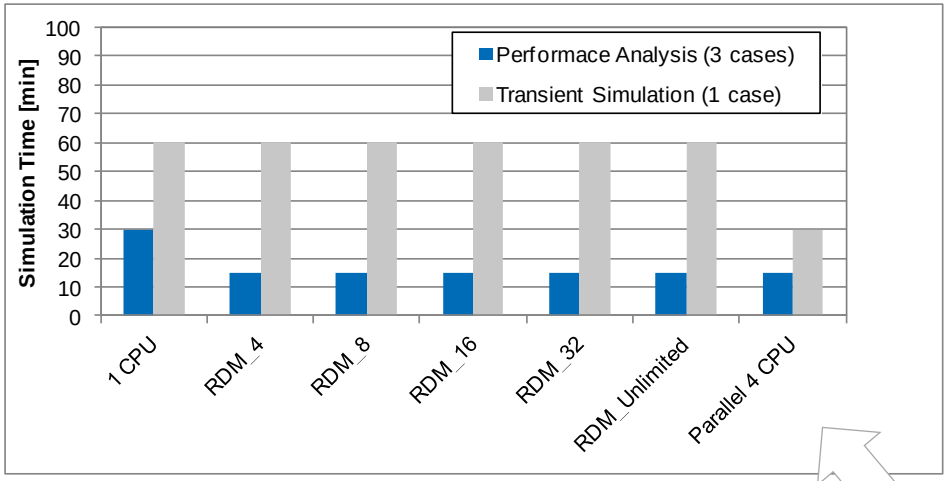
	Performace Analysis (15 cases)	Transient Simulation (1 case)	DoE (256 cases)
Example Baseline 1CPU Run Time	20min	1h	15h
RDM_4	2.5	1	4
RDM_8	4	1	7
RDM_16	6.7 (3min)	1	14
RDM_32	6.7	1	33
RDM_Unlimited	6.7	1	150 (3-5min)
Parallel 4 CPU	2 (10min)	2 (10min)	2

Both RDM and Parallel is supported by WAVE

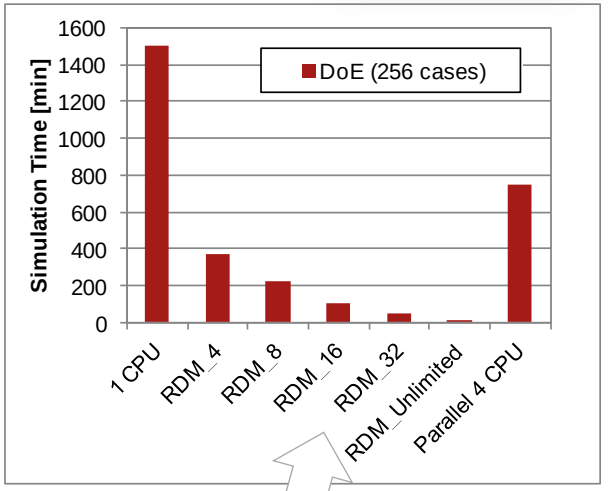


- This example shows approximate time speed up factors using various RDMs licenses and parallelization

Heavy Duty Diesel / Gas Engine or Gensets:



$$Time\ Factor = \frac{30}{15}$$



	Performace Analysis (3 cases)	Transient Simulation (1 case)	DoE (256 cases)
Example Baseline 1CPU Run Time	30	1h	25h
RDM_4	2 (15min)	1	4
RDM_8	2	1	7
RDM_16	2	1	14
RDM_32	2	1	33
RDM_Unlimited	2	1	100 (15min)
Parallel 4 CPU	2 (15min)	2 (15min)	2

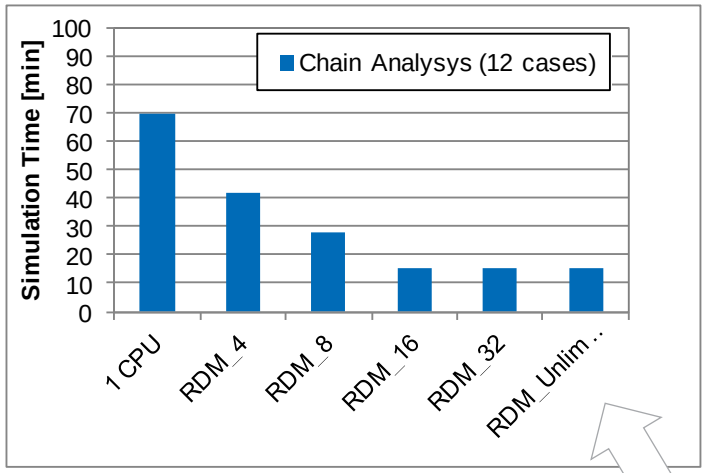
Agenda

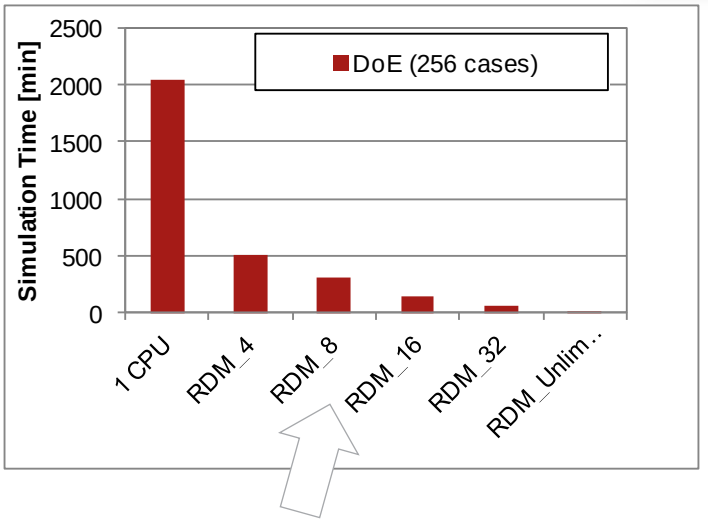
- Introduction
- Distributed Running (RDM)
- Parallel Running
- **Examples & Recommendations**
  - WAVE
  - **VALDYN**
  - VECTIS
  - ENGDYN
  - FEARCE
  - PISDYN/RINGPAK
- Conclusions

RDM is supported by VALDYN

- This table shows approximate time speed up factors using various RDMs licenses used for VALDYN chain analysis

VALDYN Chain Analysis:





	Chain Analysys (12 cases)	DoE (256 cases)
Example Baseline 1CPU Run Time	70min	34h
RDM_4	1.7	4
RDM_8	1.5	7
RDM_16	5 (15min)	14
RDM_32	5	33
RDM_Unlimited	5	136 (10-15min)

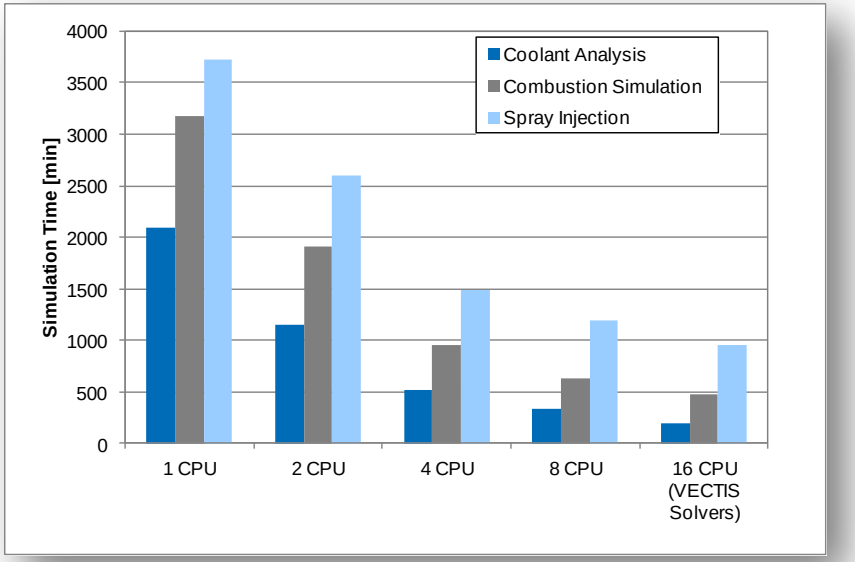
$$Time\ Factor = \frac{34 * 60}{15}$$

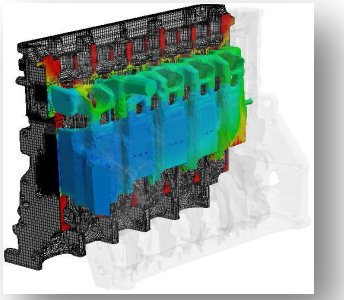
Agenda

- Introduction
- Distributed Running (RDM)
- Parallel Running
- **Examples & Recommendations**
  - WAVE
  - VALDYN
  - **VECTIS**
  - ENGDYN
  - FEARCE
  - PISDYN/RINGPAK
- Conclusions

The parallelization of VECTIS simulation provides typically excellent speed up

- Table below shows approximate time speed up factors using several CPUs in parallel for chosen anaylisys





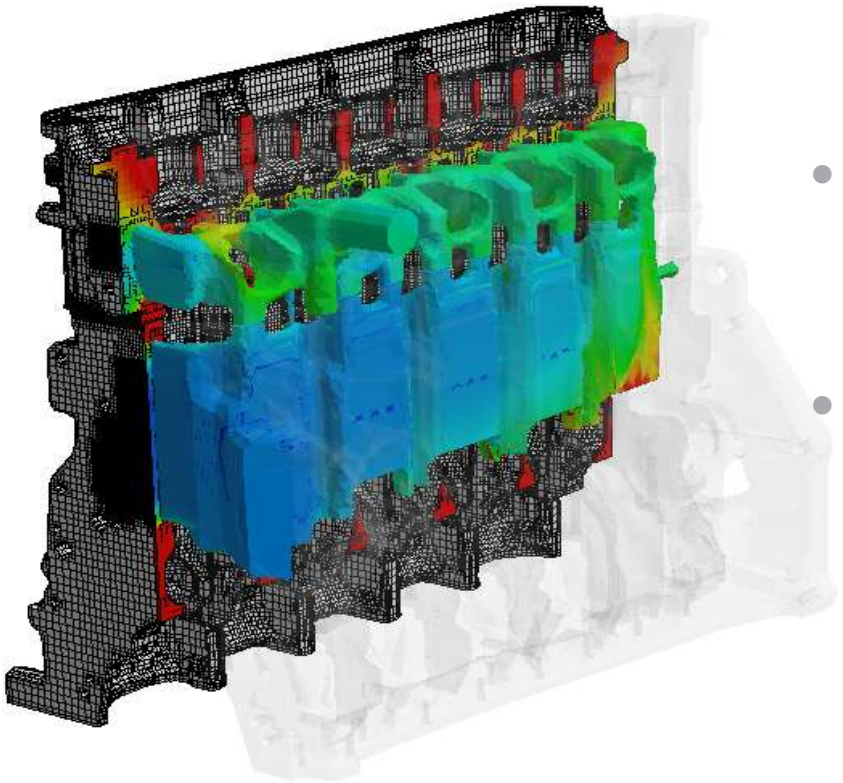
$$Time\ Factor = \frac{53}{8}$$

Frequent re-mapping, fine mesh

GDI, intake & compression

	Coolant Analysis	Combustion Simulation	Spray Injection
Example Baseline 1CPU Run Time	35h	53h	62h
2 CPU	1.8	1.7	1.4
4 CPU	4	3.3	2.5
8 CPU	6	5	3
16 CPU (VECTIS Solvers)	11 (3.5h)	7 (8h)	4 (16h)

VECTIS Parallel scalability



- There are dependencies on the mesh size
  - For example, a small mesh will lose scalability as the size of each domain reduces
  - There is a trade off between the time take for the actually solution and the time taken to pass data between the domains
- Larger models, such as underhood calculations which typically have mesh sizes between 10-15million cells will see good scalability up 64 CPU's and still significant improvement with 128cpu's+
- Table below shows approximate calculation speed up time factor using various number of processor used for VECTIS coolant analysis
  - The coolant analysis is "ideal" case for the parallelization
  - The time saving of another analyses are highly dependent

    on the specific model

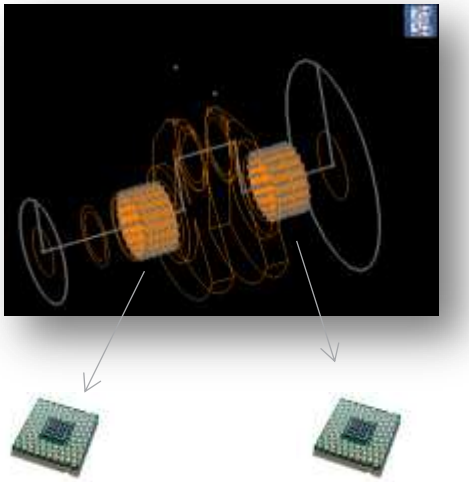
Agenda

- Introduction
- Distributed Running (RDM)
- Parallel Running
- **Examples & Recommendations**
  - WAVE
  - VALDYN
  - VECTIS
  - **ENGDYN**
  - FEARCE
  - PISDYN/RINGPAK
- Conclusions

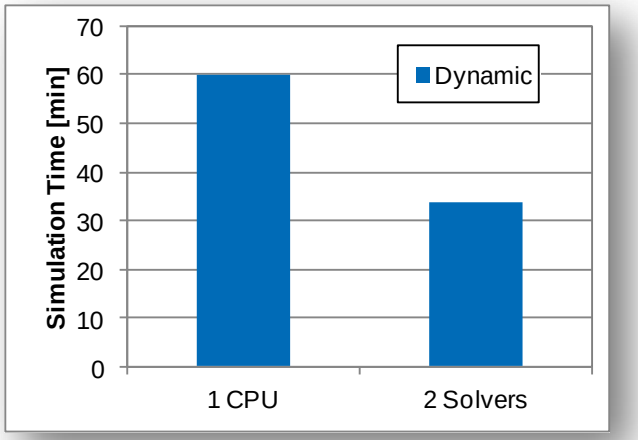
ENGDYN effectively uses parallelization in 2013.1

- Table shows approximate time speed up factors using ENGDYN over multiple processors to run a dynamic calculation with EHD solution at bearings.
  - ENGDYN benefits in one of 2 ways
    - General use of INTEL Math Kernel Library
      - Bearing independent, benefit dependent on model matrix size and solution type
    - OMP, processor per bearing
      - Benefit limited to number of bearings
      - Benefit greatest if the bearings are EHL (Dynamic and Determinate solutions)
  - Extra timings required for say an IL6 with 1 thru to 6 bearings
- 2014 Release will benefit from use of RDM

	Dynamic
Example Baseline 1CPU Run Time	60min
2 Solvers	1.8 (34min)



1 CPU => 1 Bearing

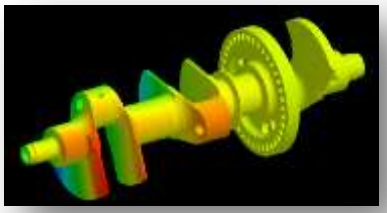


Agenda

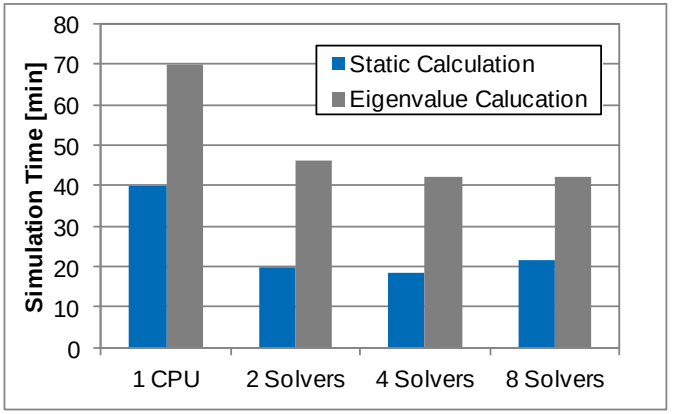
- Introduction
- Distributed Running (RDM)
- Parallel Running
- **Examples & Recommendations**
  - WAVE
  - VALDYN
  - VECTIS
  - ENGDYN
  - **FEARCE**
  - PISDYN/RINGPAK
- Conclusions

FEARCE effectively uses parallelization

- Table below shows approximate calculation time reduction using FEARCE over multiple processors to perform a static and eigenvalue solutions
  - Static solution : Displacement analysis, 62 load cases
  - Eigenvalue solution : 50 modes



$$Time\ Factor = \frac{40}{18}$$



	Static Calculation	Eigenvalue Calucation
40mins	40mins	70min
2 Solvers	2	1.5
4 Solvers	2.2 (18min)	1.7 (42min)
8 Solvers	1.9	1.7

- Need to investigate for 2013.1 release why no benefit > 2 CPU's, and fix
- As with all FE solvers solution speed can be greatly influenced by I/O disk performance
- 2013.1 will include speed improvements for FEARCE/VECTIS mapping (without parallelization)
  - Anecdotal evidence (via consulting) suggests we are already quicker than competitors
  - Published timings for say IL4 mapping would be advantageous with 2013.1 release

Agenda

- Introduction
- Distributed Running (RDM)
- Parallel Running
- **Examples & Recommendations**
  - WAVE
  - VALDYN
  - VECTIS
  - ENGDYN
  - FEARCE
  - **PISDYN/RINGPAK**
- Conclusions

PISDYN/RINGPAK doesn't effectively use parallelization

- PISDYN
  - Although the program supports multiple CPU's there is no value in enabling this feature
    - Utilises Math Kernel Library (MKL), but this is of little benefit in PISDYN
    - Future enhancement required to use OMP in a similar way to ENGDYN 2013.1
      - Processor per sub-model (skirt, piston bearing, S/E bearing etc)
      - Benefit will be limited to number of sub-models
      - Benefit will be greater once EHL of piston bearings is included
  - 2014 Release will allow multiple cases, and therefore use of RDM
    - This may be more effective than OMP
- RINGPAK
  - No parallelization enabled
  - Run times circa 5 minutes per case (for current RINGPAK)
  - 2014 Release (RINGPAK-3D) will benefit from parallelization
    - Likelihood multiple slices will use OMP
    - Multiple cases will be solved with RDM (as with other products)

Agenda

- Introduction
- Distributed Running (RDM)
- Parallel Running
- Examples & Recommendations
- **Conclusions**

Conclusions

- This presentation shows the speed up possibilities using distributed (RDM) and parallel running for common analyses and DoE studies
- Distributed running possibilities RDM_4, RDM_8, RDM_16, RDM_32, RDM_Unlimited and Parallel running were introduced
- It is important to understand how both approaches work to select the correct one for chosen code and analysis type
  - Distributed run presents very efficient way for analysis run time speed up (WAVE, VALDYN)
    - Up to **7** times faster run time for WAVE performance analysis and VALDYN with 15 cases RDM_16
    - Up To **14** times faster run time for WAVE DoE with 4 variables (typically 256 cases) with RDM_16
  - Parallel run (VECTIS, WAVE and ENGDYN)
    - Essential for VECTIS where it is used for mesh dividing into multiple CPUs with **4-11** time faster calculation time reduction when using 16 cores
    - Efficient for special WAVE cases e.g. transient simulations or acoustic where running at maximum **4** CPUs can lead to **2** x faster run time
    - Up to **2** x faster run with 2 CPUs running FEARCE for chosen analysis
    - Up to **1.8** speed factor runtime with 2 CPUs running ENGDYN (dependent on model/solution type)

Appendix 1: RDM License Packages

- Below table shows available licensing packages
- Please note that the parallel run can be run independently from distribution run

	RDM_4	RDM_8	RDM_16	RDM_32	RDM_Unlimited
Package	4 CPU	8 CPU	16 CPU	32 CPU	Unlimited CPU
Public Comments	Parallel processor keys for Run Distribution on 4 CPUs and 1 parallel user	Parallel processor keys for Run Distribution on 8 CPUs and 2 parallel users	Parallel processor keys for Run Distribution on 16 CPUs and 4 parallel users	Parallel processor keys for Run Distribution on 32 CPUs and 8 parallel users	Parallel processor keys for Run Distribution on unlimited CPUs and unlimited
Private Comments	ENGDYN Base plus stress and acoustics				
Keys	RunDistribution_4 ISIS_Queue RS_Parallel	RunDistribution_8 2 x ISIS_Queue 2 x RS_Parallel	RunDistribution_16 4 x ISIS_Queue 4 x RS_Parallel	RunDistribution_32 8 x ISIS_Queue 8 x RS_Parallel	RunDistribution u x ISIS_Queue u x RS_Parallel

Appendix 2: Example of Complex Acoustic DoE study using RDM