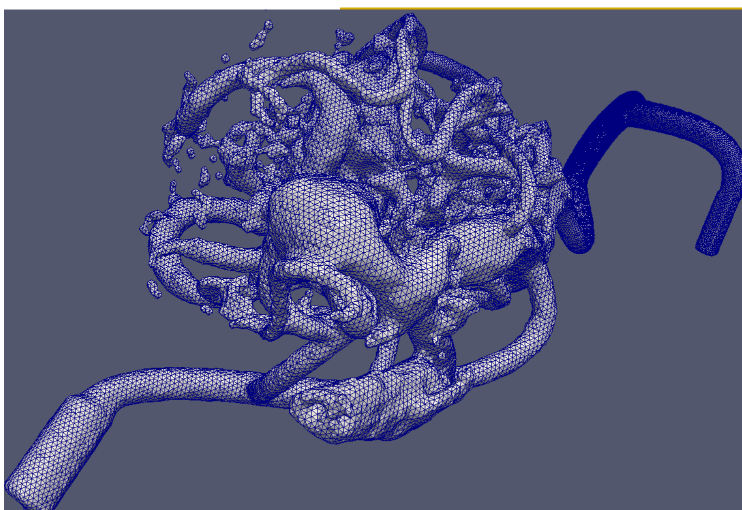
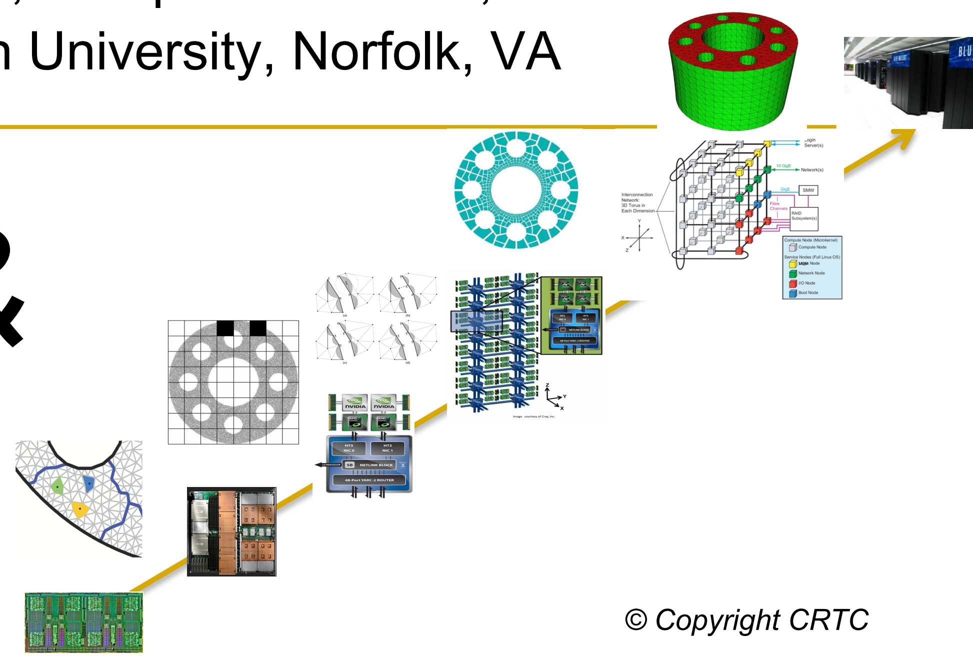
Center for Real-Time Computing: On going efforts in parallel mesh generation and medical image computing

Nikos Chrisochoides
CRTC Lab, Computer Science,
Old Dominion University, Norfolk, VA

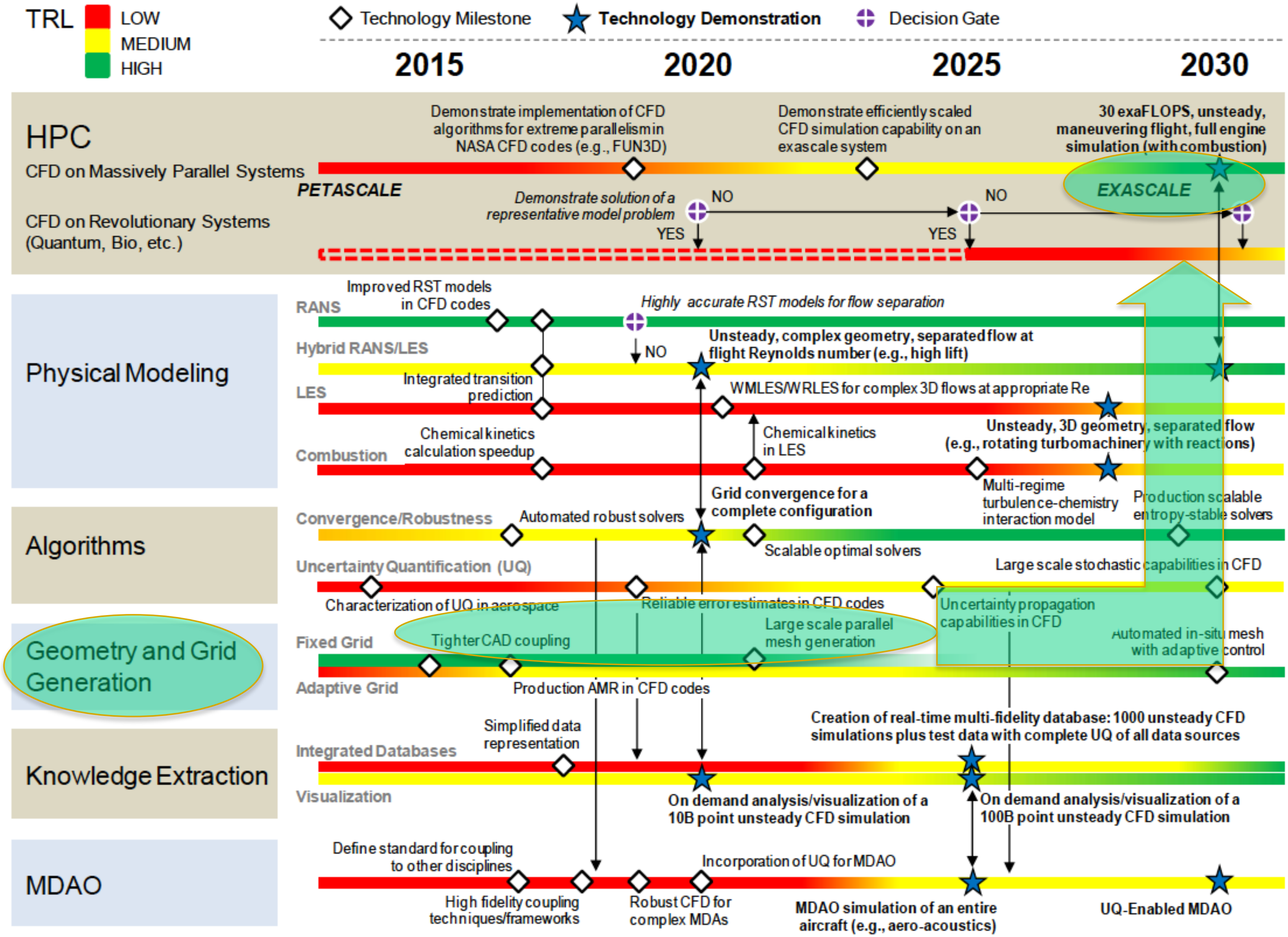


&

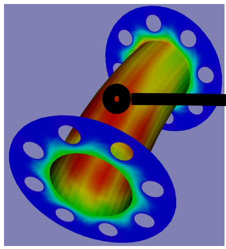


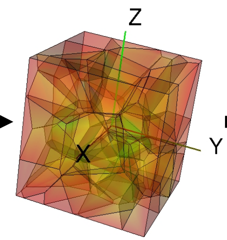
Supported in part by NSF (CCF-1439079 & -0916526) /NIH/NASA grants, the John Simon Guggenheim Foundation and Richard T. Cheng Endowment

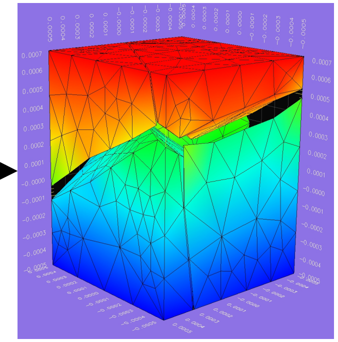
CFD Vision 2030 Study: A Path to Revolutionary Computational Aero-sciences



The 90's Approach to Parallel FEM







Late 90', Cornell TC, IBM SP2 16 PEs with parallel I/O:

Mesh Size	Time (sec)	I/O	CSR Tran.	ParMetis	Data Movem.	Total (//mesh)
1M	255.9	217.8	24.28	5.46	85	588.44 (38.6) **x 7**
2M	461.8	471	50.58	8.58	173.3	1162.2 (85.1) **x 8**

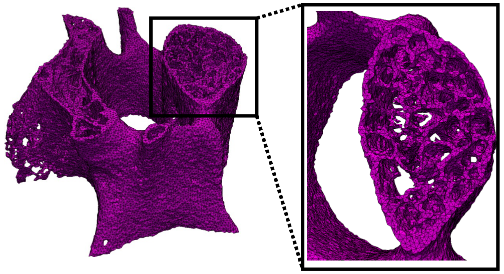
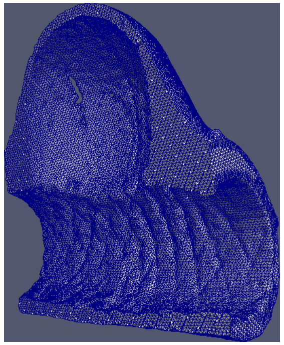
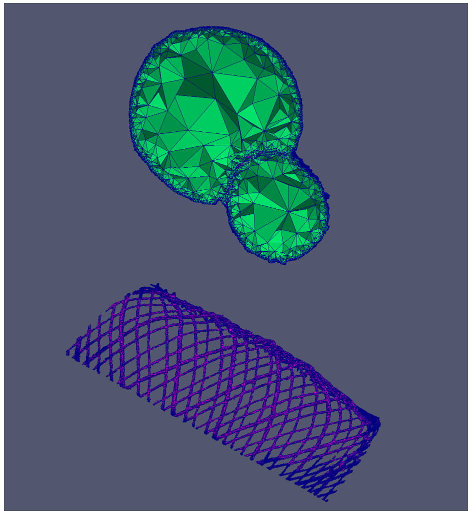
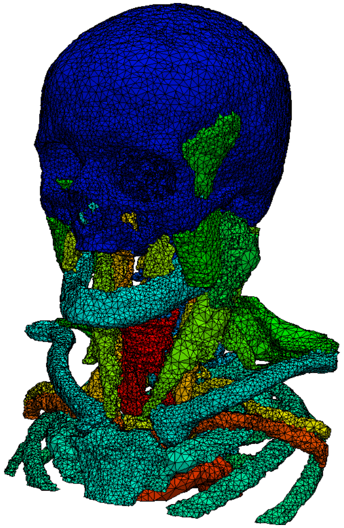
Traditional			SMGP		SMGP + ParMetis	
M_{Size}	$S_{Sep.}$	$D_{Imb.}$	$S_{sep.}$	D_{Imb}	$S_{sep.}$	D_{Imb}
1M T	0.042	1298	0.064	3058	0.044	883
2M T	0.038	2009	0.061	6578	0.037	1529

I/O & data movement > 50%

Simultaneous Mesh Generation and Partitioning for 3D Domains D. Nave and N. Chrisochoides.
In Proceedings of 8th International Meshing Roundtable, Lake Tahoe, CA, pp 55-66, Sandia National Labs Publications, ***1999.***

Parallel FEM Simulation of Crack Propagation *- Challenges, Status and Perspectives (Bruce Carter, Chuin-Shan Chen, L. Paul Chew, N. Chrisochoides, Guang R. Gao, Gerd Heber, Antony R. Ingraffea, R. Krause, Chris Myers, Demian Nave, Keshav Pingali, Paul Stodghill, Stephen S. Vavasis, P. Wawrzynek. In 7th International Workshop for Solving Irregularly Structured Problems in Parallel, Cancun, Mexico, Lecture Notes in Computer Science Vol. 1800, pp 443-449,* ***2000.***

The gap will only increase: I/O and Complexity & scales of models

Traditionally approach today:

#Tets	Meshing	CSR	ParMetis	I/O	Total
14M	149s(1)	0.2s	7s	67s	**223s**

30% I/O overhead W/o the cost for Data movement and memory contention

Parallel Mesh Generation today: mesh is already in place

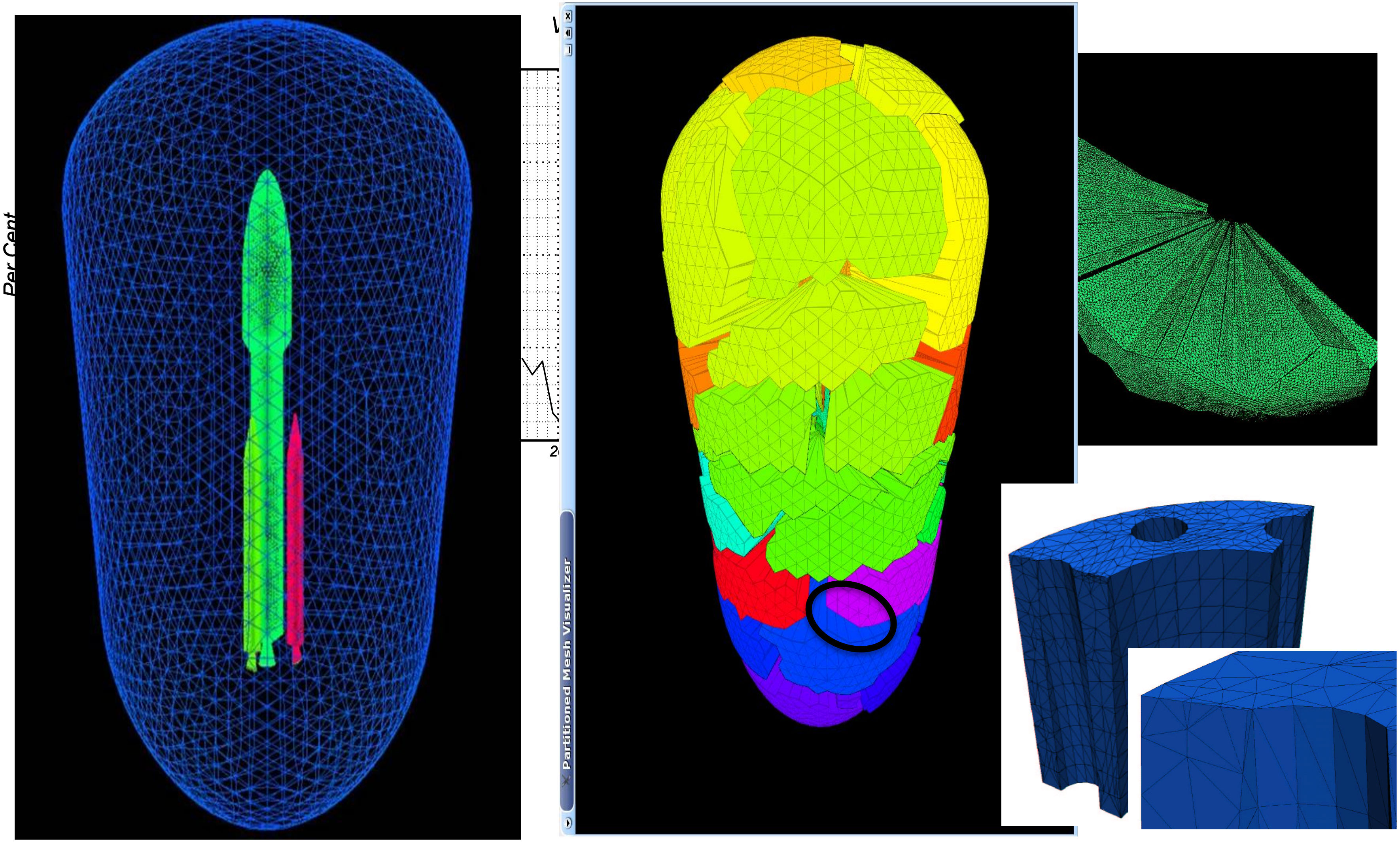
#Tets	// Meshing	CSR	ParMetis	IO	Total
14M	**13s(12)**	0.2s	**7s***	**0s**	**20.2s**

X11

*Dominique Lasalle, George Karypis: Multi-threaded Graph Partitioning, IPDPS, 2013

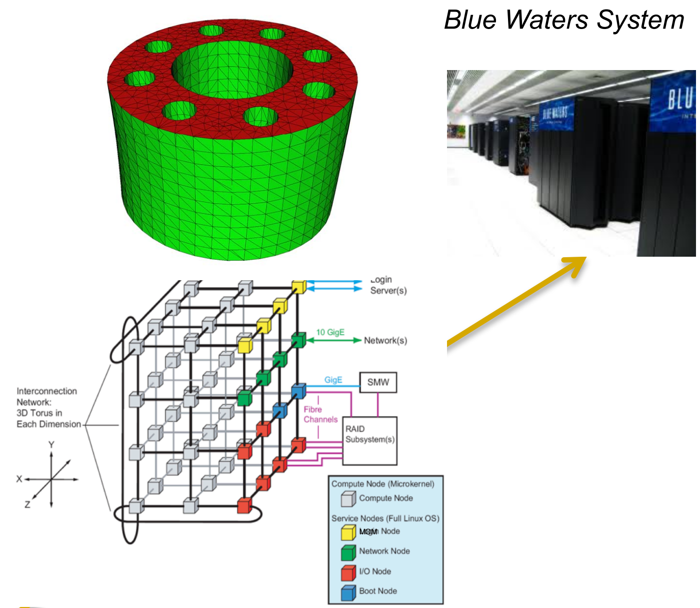
For large scale platforms the difference is even higher!

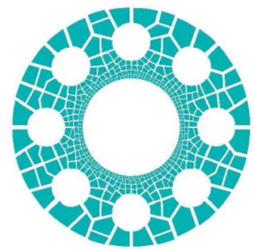
Traditional Approach to Parallel Meshing



Parallel Mesh Generation Using an Advancing Front Method Ywith Yasushi Ito, Alan Shih, Anil Erukala, Bharat Soni, Andrey Chernikov, Nikos Chrisochoides and Kazuhiro Nakahashi. Mathematics and Computers in Simulation, Publisher Elsevier, Volume 75, No. 5, pages 200 -- 209, September, 2007

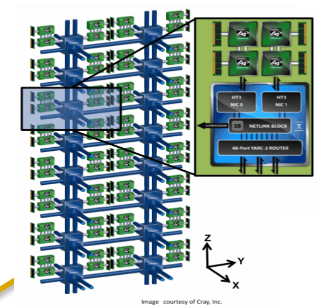
Telescopic Approach for exascale-era computing



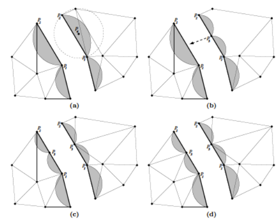


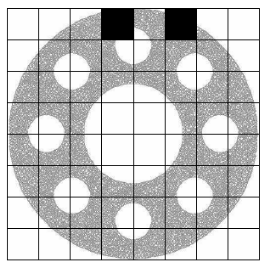
PDD

Rack
(24 Geminis)



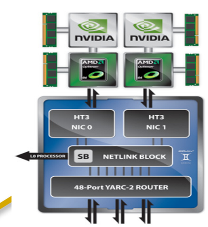
Parallel Mesh Generation





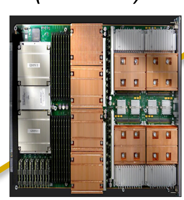
PCDM

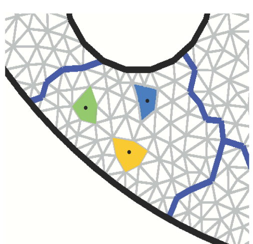
Gemini
(2 nodes)



PDR

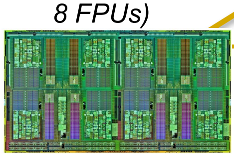
SMP node
(2 CPUs)

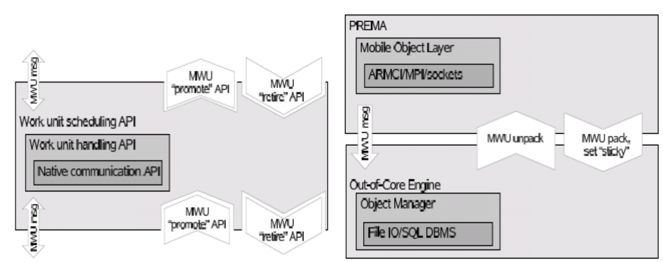




PODM

AMD Interlagos
(16 cores /
8 FPUs)



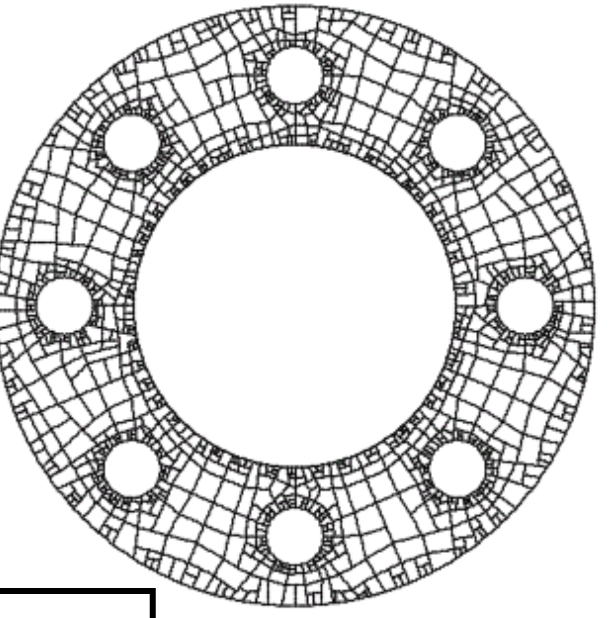


Parallel Runtime System & Out-of-Core Support

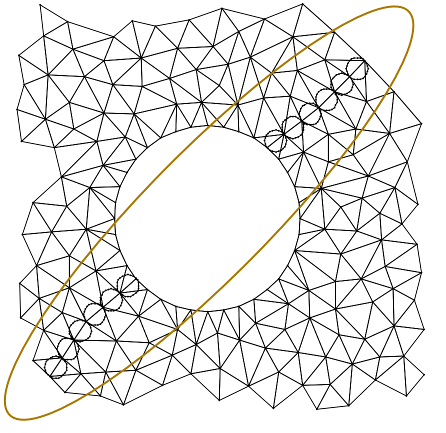
Towards Exascale Parallel Delaunay Mesh Generation
Nikos Chrisochoides, Andrey Chernikov, Andriy Fedorov, Andriy Kot, Leonidas Linardakis and Panagiotis Foteinos.
18th International Meshing Roundtable, pages 319 -- 336, Salt Lake City, Utah, October, 2009

Definitions & Taxonomy

- Given a domain
  overdecompose the geometry into N >> P **subdomains**
- In parallel mesh the subdomains



Coupling Method	Delaunay Method	Advancing Front	Edge Subdiv.
Tightly	**PODM**	Lohner99	PCTE
Partially	**PCDM** **GPDR**	Lohner89 DeCougny99	Castanos97 Jones94
Decoupled	Galtier97 **PD^3**	PAFT	DeCougny99 PTE



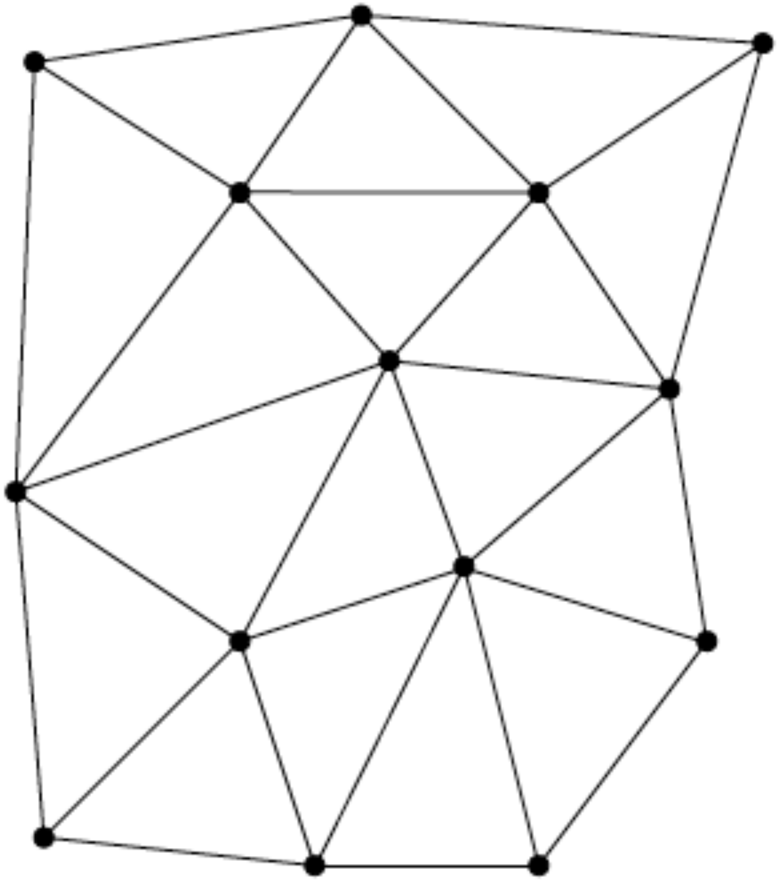
Parallel Mesh Generation. Nikos Chrisochoides, In Numerical Solution of Partial Differential Equations on Parallel Computers, (Eds. Are Magnus Bruaset, Petter), Springer-Verlag, 2005.

Delaunay Mesh Generation

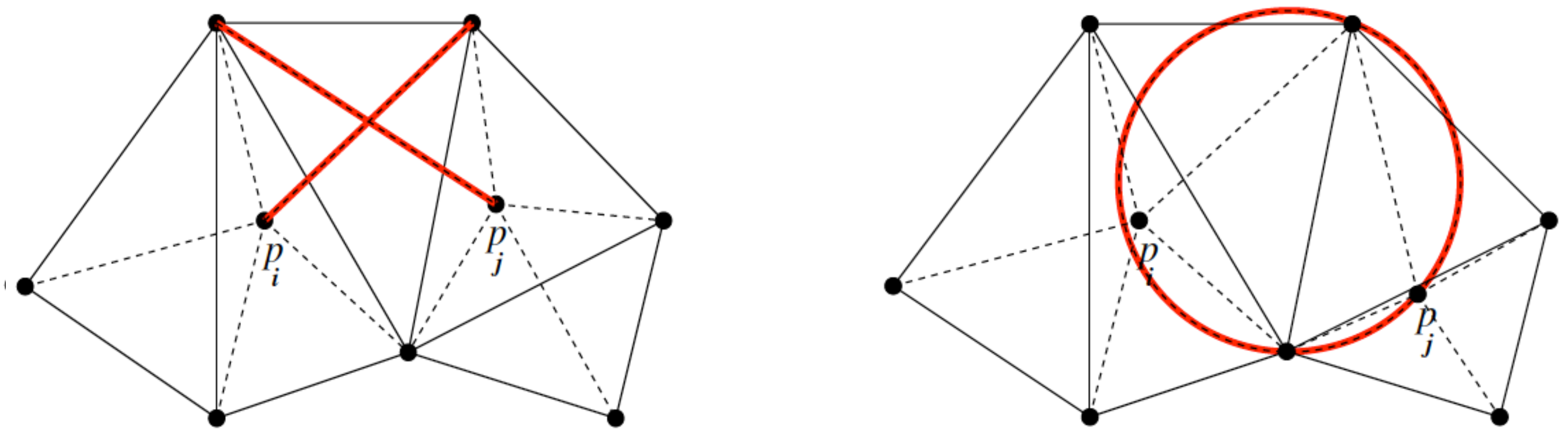
- Bowyer-Watson kernel
  - Marked element
  - Insert its circumcenter
  - Compute its cavity
  - Triangulate its cavity

$$M_{i+1} = M_i - C_i + B_i$$

- before and after BW kernel the mesh is:
  - conformal
  - Delaunay



Challenges and Requirements



Stability: *the elements of the parallel generated mesh should retain the* ***same quality*** *as the elements of sequentially generated mesh;*

Code re-use: *meshing is labor intensive.*
Parallel code development still very difficult, …

Efficiency: *Single core and node performance/power*

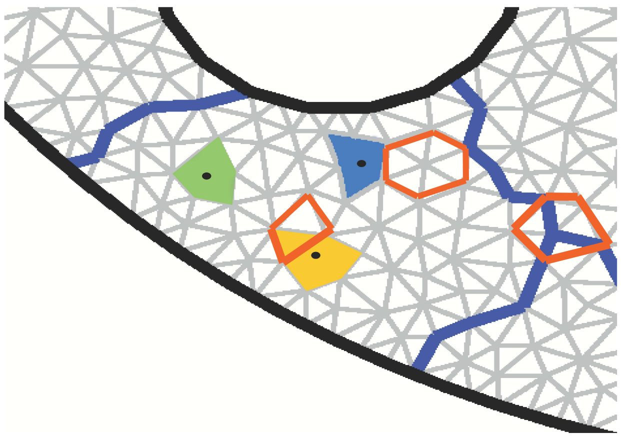
Maintenance and Correctness: *Functionality, Algorithm/Software (termin.)*

Code Re-use

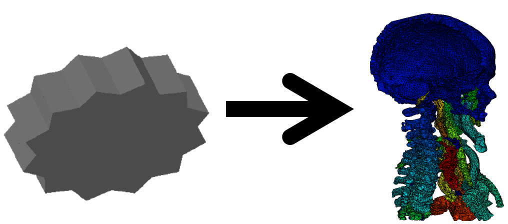
leverage the ever evolving basic meshing algorithms/software

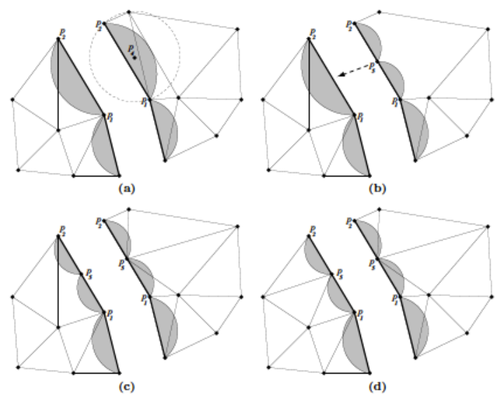
Low ↔ *High*

PODM | *PCDM* | *PDR*

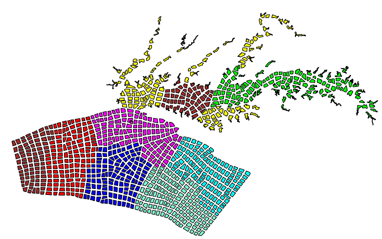


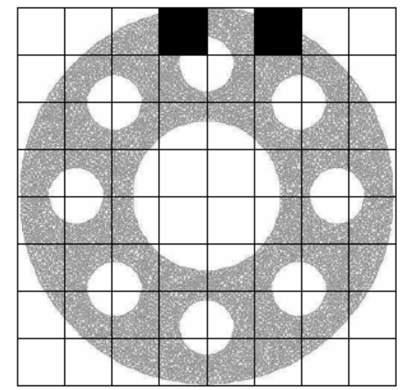
ACM SoCG02, CGTA04, ISVD10, IMR12
ACM ICS13, CGTA14, JPDC14



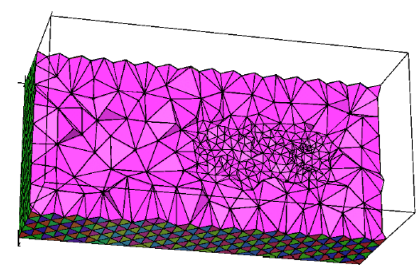


ACM TOMS08 [870,872]

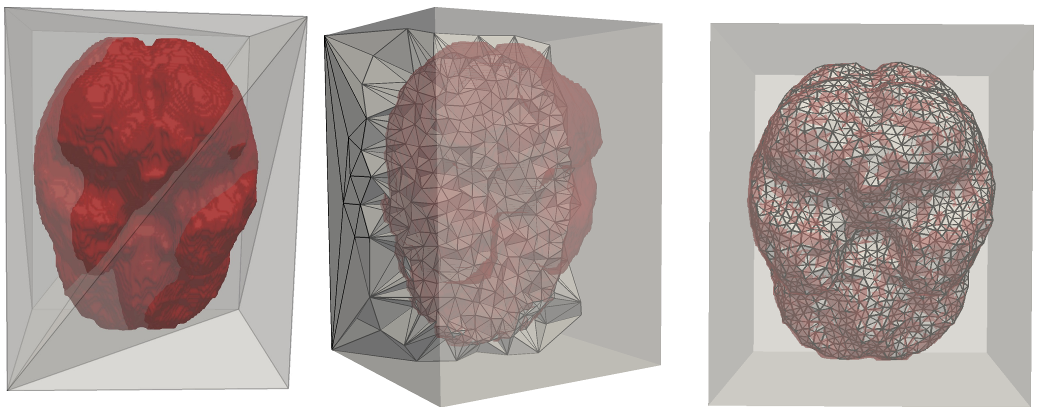


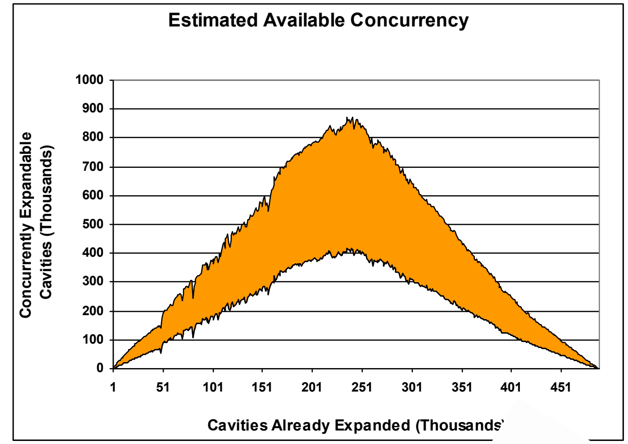


ACM ICS04 & ICS08, FEAD10,
SIC10, SISC11, SISC12



Parallel Optimistic Delaunay Mesh

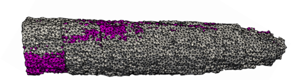




Cavity level (i.e., medium grain for 2D, 3D and 4D: ~5, 15, 150 elems)

3D Mesh

#threads	1	32	64	128	144	160	176
#Tets	10.7M	3.49M	7.44M	0.132B	0.151B	0.167B	0.185B
Average Time(secs)	**90.37**	87.50	99.23	93.00	**103.26**	150.03	181.10
Speedup	1.00	**33.71**	63.33	119.56	123.67	94.10	86.36
efficiency	1.00	1.05	0.99	0.93	**0.86**	0.59	0.49



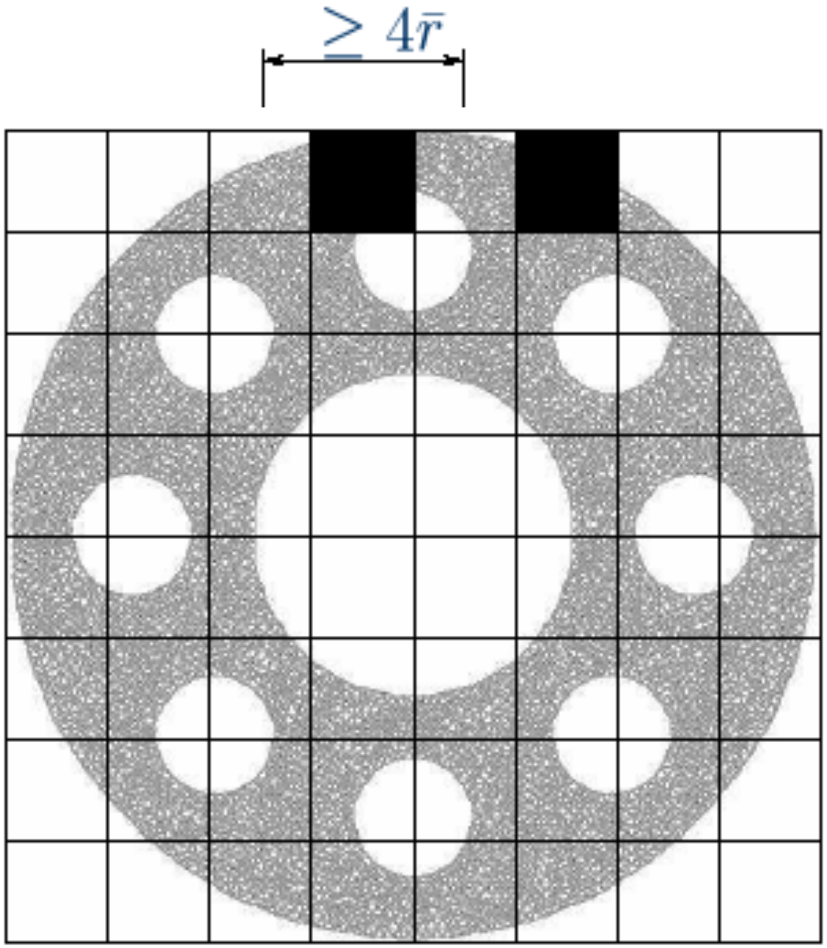
4D Mesh

1	12
301.4K	2.837M
1679s	2489s
1	6.35
Overhead	871s

MRI Myocardium

Real-Time Image-to-Mesh conversion for Medical Images, *(with P. Foteinos): ISVD10, IMR11, MICCAI11, ACM ICS13, CGTA14, JPDC14, EwC14*

Parallel Delaunay Refinement (PDR)



Sufficient condition for point Independence

Given two points p_i and p_j we can show that they can be inserted concurrently if

$$\|p_i - p_j\| \geq 4\bar{r}$$

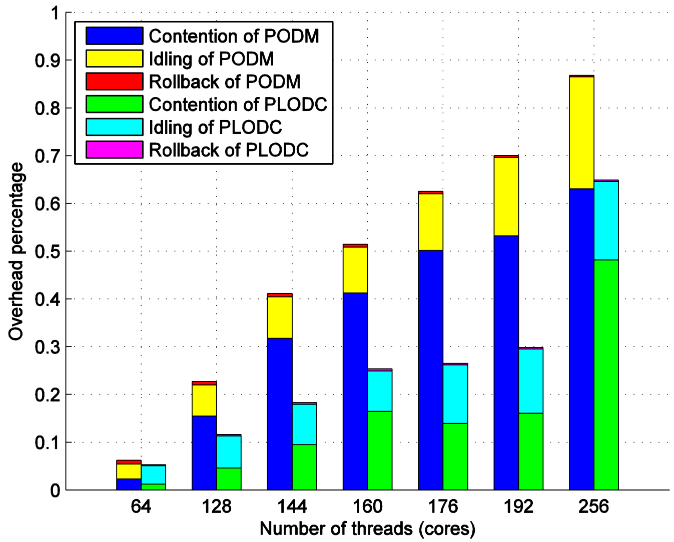
Then given a block decomposition we can concurrently refine selective blocks at a time

Practical and Efficient Point Insertion Scheduling Method for Parallel Guaranteed Quality Delaunay Refinement, (with A. Chernikov and N. Chrisochoides), In the proceedings of ACM 18th International Conference on Supercomputing, 2004.

Putting Together: PODM & PDR (cont.)

PODM

cores	1	128	144
No. elms	3.1e+7	3.74e+9	4.2e+9
T_{PODM} (s)	27.78	35.97	47.24
Speedup	1	94.5	80.8
Efficiency	1	0.74	0.56
T_{GOSD} (s)	27.78	31.46	34.04
Speedup	1	109	112
Efficiency	1	0.85	0.78

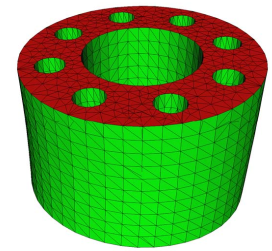


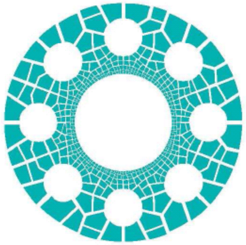
PODM & PDR

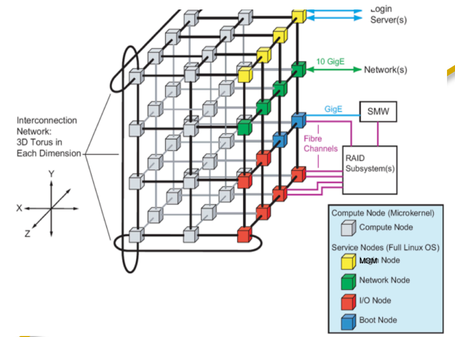
cores	1	128	144	160	176	192
No. elms	3.1e+7	3.74e+9	4.2e+9	4.67e+9	5.14e+9	5.59e+9
T_{PODM}(s)	27.78	35.97	47.24	57.24	74.20	92.77
Speedup	1	94.5	80.8	74	64	55
Efficiency	1	0.74	0.56	**0.46**	**0.36**	**0.29**
T_{GOSD}(s)	27.78	31.46	34.04	37.24	37.82	39.63
Speedup	1	109	112	114	124	128.2
Efficiency	1	0.85	0.78	**0.71**	**0.7**	0.67

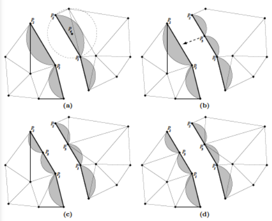
Domain Decomposition

Blue Waters System



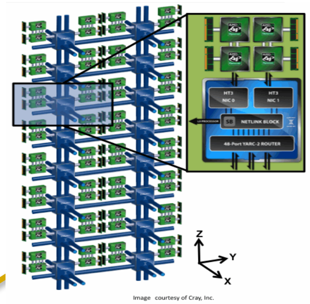




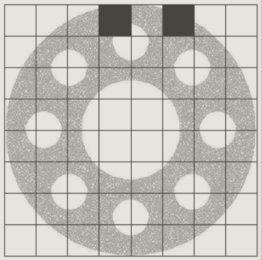


PDD

Rack
(24 Geminis)

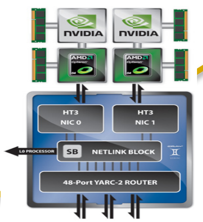


Data Decomposition



PCDM

Gemini
(2 nodes)

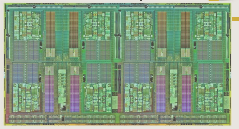


PDR

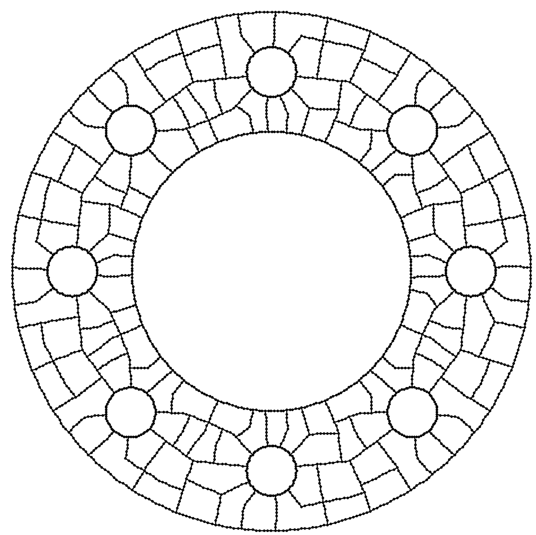
SMP node
(2 CPUs)

PODM

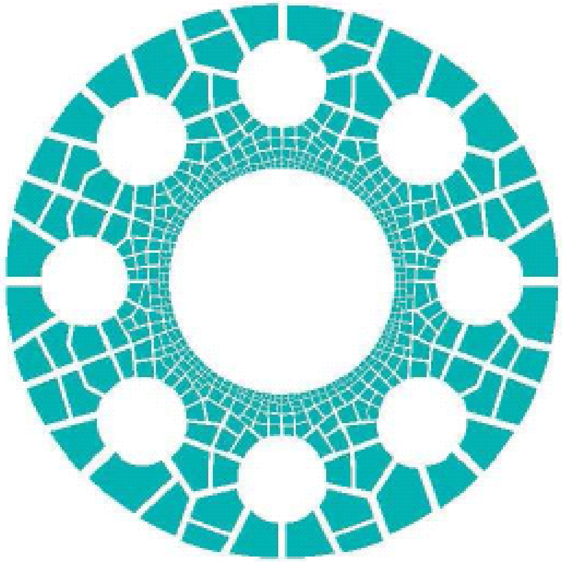
AMD Interlagos
(16 cores /
8 FPUs)

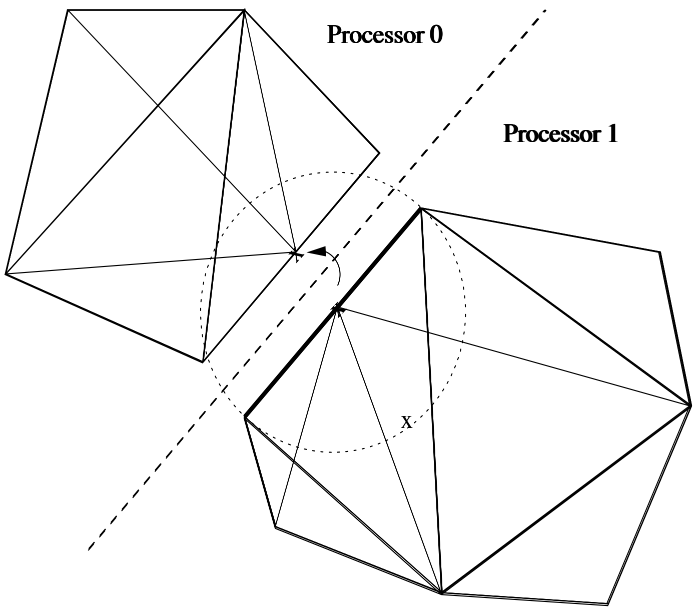


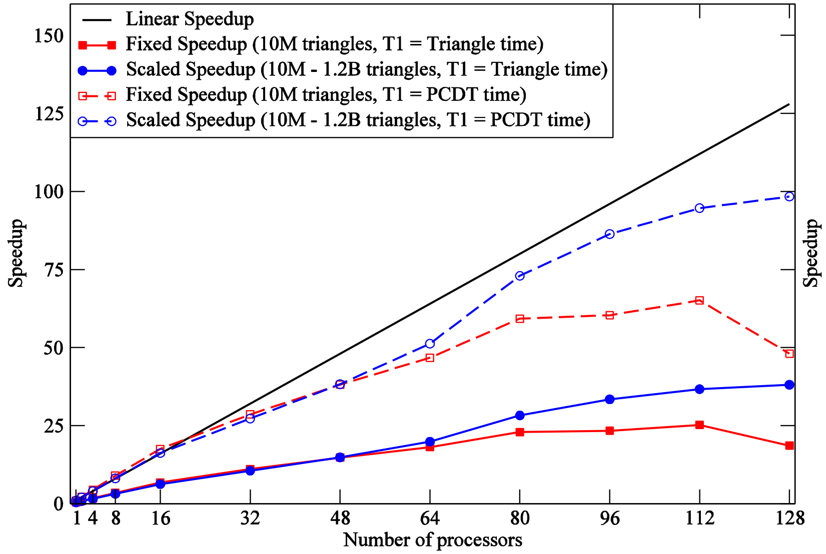
Concurrency
2×10^2 (.9 ef.)



Parallel Constrained Delaunay Mesh





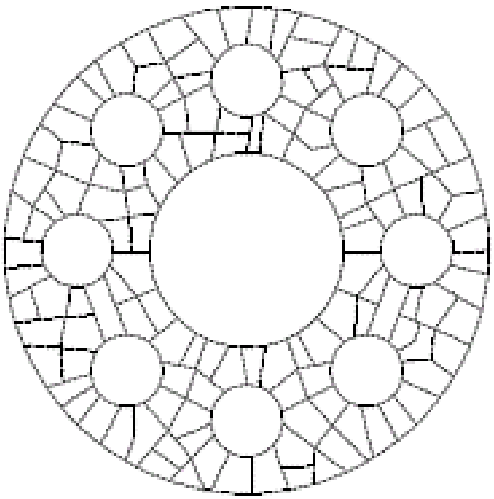


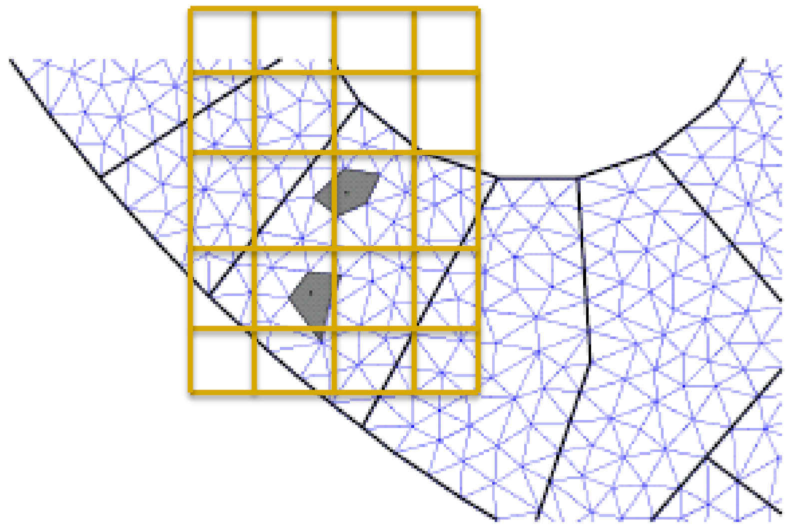
	Computation			Communication		
	min	avr	max	min	avr	max
PCDM	0.134	0.141	0.147	0.0001	0.004	0.005
PDR	0.006	0.606	0.907	0.2654	0.5662	1.169

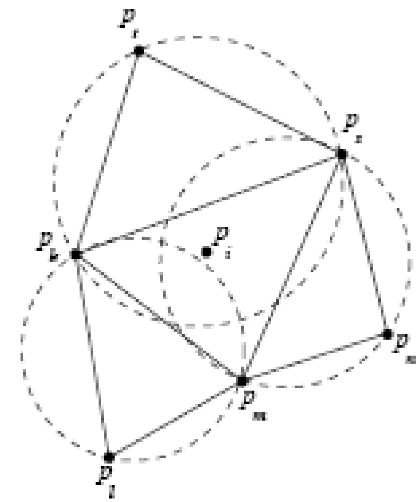
Pipe, 64 processors, time in secs per 10^6 elems

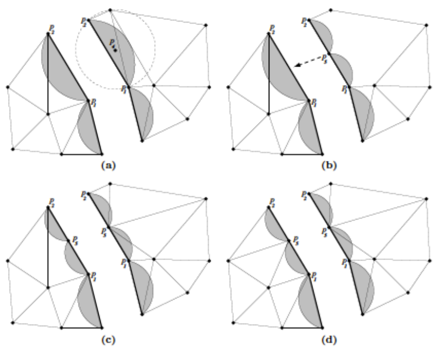
Parallel 2D Constrained Delaunay Mesh Generation (with A. Chernikov and N. Chrisochoides) ACM Transactions of Mathematical Software, Vol 34(1), February, 2008.

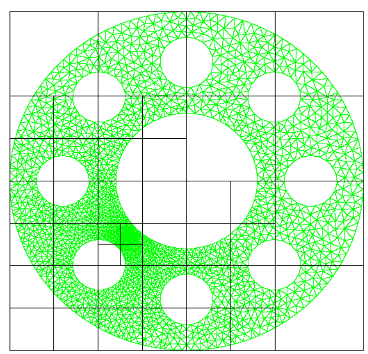
Telescopic Approach: PCDM + LAPD (PODM + PDR)











For 4D only

PCDM & ***GOST = PODM + PDR for 2D/3D/4D***

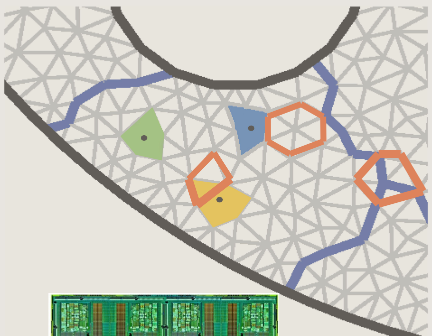
Generalized Two-Dimensional Delaunay Mesh Refinement Andrey Chernikov and Nikos Chrisochoides. SIAM Journal on Scientific Computing, Volume 31, No. 5, pages 3387 -- 3403, 2009
Generalized Insertion Region Guides for Delaunay Mesh Refinement Andrey Chernikov and Nikos Chrisochoides. SIAM Journal on Scientific Computing, Voll. 34, No. 3, pp A1333 -- A1350, 2012

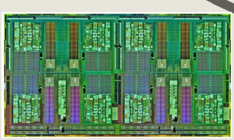
Putting It All Together

- **Code re-use:** leverage the ever evolving basic meshing algorithms/**software**
- **Communication/Synchronization:** leverage network/memory hierarchy

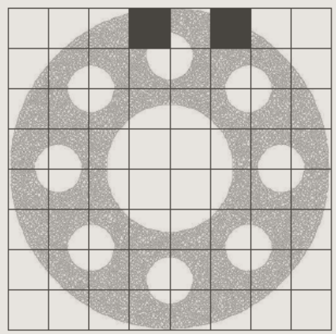
Data Decomposition *Domain Decomposition*

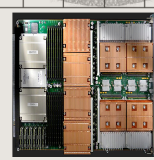
PODM



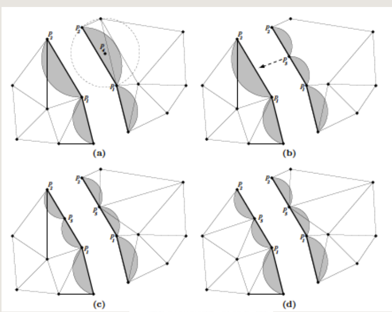


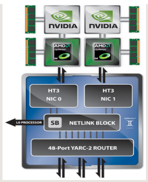
PDR



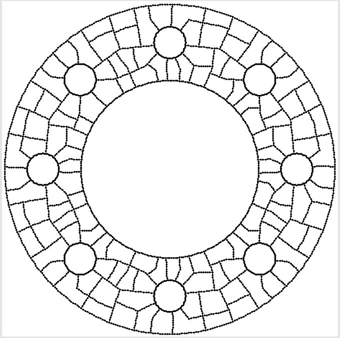


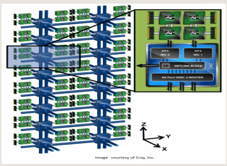
PCDM

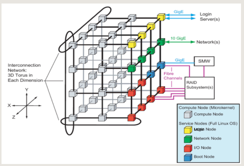




PD^3







Concurrency:

Today: 2D	*$2x10^2$ (0.7 ef)*	*x*	*10^2 (0.9 ef)*	*x*	*10^4 = 10^8*	
3D	*$2x10^2$ (0.7 ef)*	*x*	*10^3 (0.9 ef)*	*x*	*10^4 = 10^9*	*(2015-17)*
4D	*$2x10^2$ (0.5 ef) (2017)*		*TBD*		*TBD*	

Tightly Coupled *Partially Coupled* *Decoupled*

ACM SoCG02, CGTA04, ISVD10, IMR12, ACM ICS13, CGTA14, JPDC14a

ACM ICS04 & ICS08, FEAD10, SIAM SISC11, SISC12 JPDC14b

ACM TOMS08 [870,872] SIAM SISC10

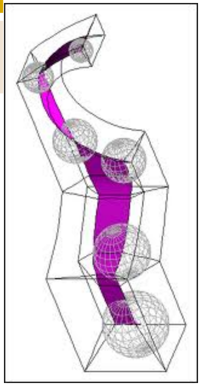
SIAM SISC06 SISC08

Summary

- Presented four different methods for 2D, 3D and 4D (3D+time) objects/ images and a RTS for their parallel implementation
- Map these methods according to their communication/synchronization requirements in current architectures with deep network hierarchies
- Quality is the same and sometimes better and performance is much better to known methods:

	CGAL	Tetgen	GSH3D	PODM	PODM+PDR
Tets/sec	37K(1)	143K(1)	49K(1)	**12.3M(144)**	**14.12(192)**

and guarantee quality!



- A plan to get to a billion-way concurrency
  - Today: **2×10^2 for 3D images** and 10^8 for 2D geometries
  - By 2017 we should have **10^6 concurrency** (0.7 ef) for 3D images!
  - By 2020 we should be close to **billion-way concurrency**!