International Symposium on Physical Design

sigda

ISPD 2020 Special Session: Wafer Scale Engine Placement Contest

June 18, 2020, 6pm PT

https://zoom.us/j/94879259570

Marvin Tom
Michael James
Vladimir Kibardin
Robby Fry
Patrick Groeneveld

cerebras

Program

- 6:00pm Welcome, ISPD's Benchmark Contest History
  - Patrick
- Overview CS-1 Waferscale Engine Hardware
  - Marvin
- 6:10pm The wafer scale placement problem, formats, benchmarks
  - Vladimir
- 6:20pm Teams & Result Measurement
  - Patrick
- 6:20pm Top-5 Teams: short 8-min presentations
  - Cupid, Gigaplacer, Ripplewafer, NTU428,I ISPD_AI_Team
- 7:15pm Results, Reveal of the winners
  - Patrick
- 7:30pm closing


Gigaplacer
Xidian University
PosPlace
Pohang Univ.
Deep Learning is a Scam
Binghamton Univ. NY
Cerebras Systems
Silicon Valley, CA
Sizeisnotaproblem
UT Dallas, TX
ISPD_AI_Team
Fuzhou & Nanjing
NTU428
Nat. Taiwan Univ.
TCLab
Nat. Tsing Hua Univ.
vdalab
Nat. Chiao Tung Univ.
Rippleplacer
CUHK
Cupid
CUHK
China
South Korea
Taiwan
GANSU
NINGXIA
SHANXI
SHANDONG
HENAN
JIANGSU
ANHUI
ZHEJIANG
FUJIAN
GUANGXI
GUANGDONG
Yellow Sea
East China Sea
United States

Benchmark Contest History

- 1988 Macro & Standard Cell Placement
- 2005&2006 Standard Cell Placement
- 2007&2008 Global Routing
- 2009&2010 Clock Tree Synthesis
- 2011 Congestion Driven Placement
- 2012&2013 Discrete Gate Sizing
- 2014 Routing-aware placement
- 2015 Blockage&Routing-aware placement
- 2016 Routability-driven FPGA placement
- 2017 Clock-aware FPGA Placement
- 2018&2019 Detailed Routing
- **2020 Wafer Scale Placement for ML**

Bill Swartz
ISPD2020 chair

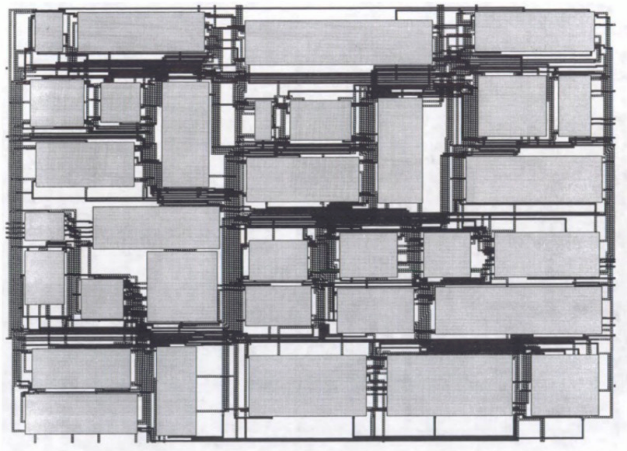
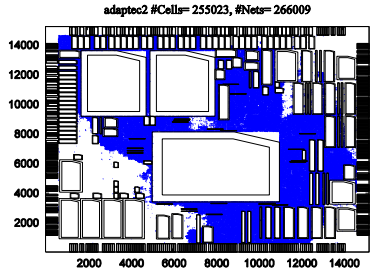
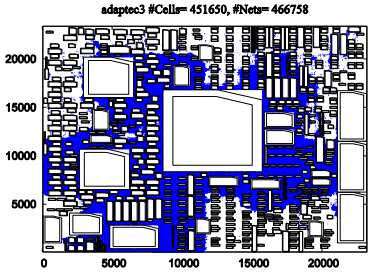


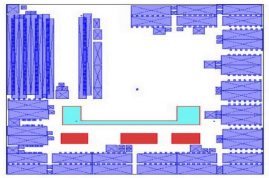
Figure 2.33. The benchmark circuit AMI33.

Table 2.3. 1988 MCNC Benchmark Results

	AMI33		XEROX	
	Area	# Vias	Area	# Vias
DELFT	2.60	967	26.57	925
SEATTLE S.	2.73	862	28.63	1235
VITAL	3.21	763	31.71	1029
MOSAICO	3.10	885	29.01	1173
BEAR	3.05	1092	28.47	1101







Cerebras Systems

- Founded in 2016
- 200 people team
- Building systems to drastically change the landscape of compute for AI

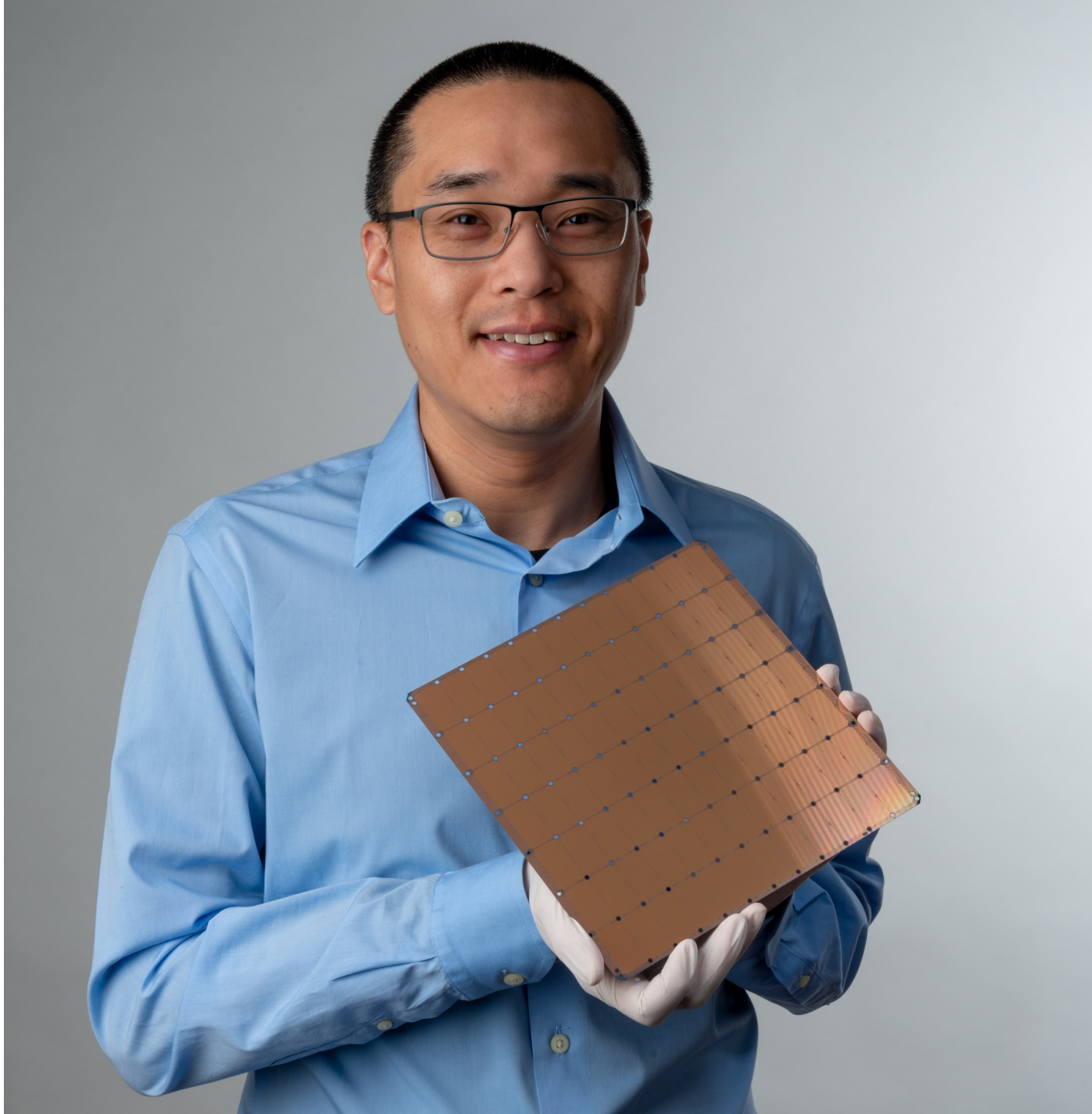
BENCHMARK | foundation capital | ECLIPSE

coatue | VY Capital | ALTIMETER

cerebras

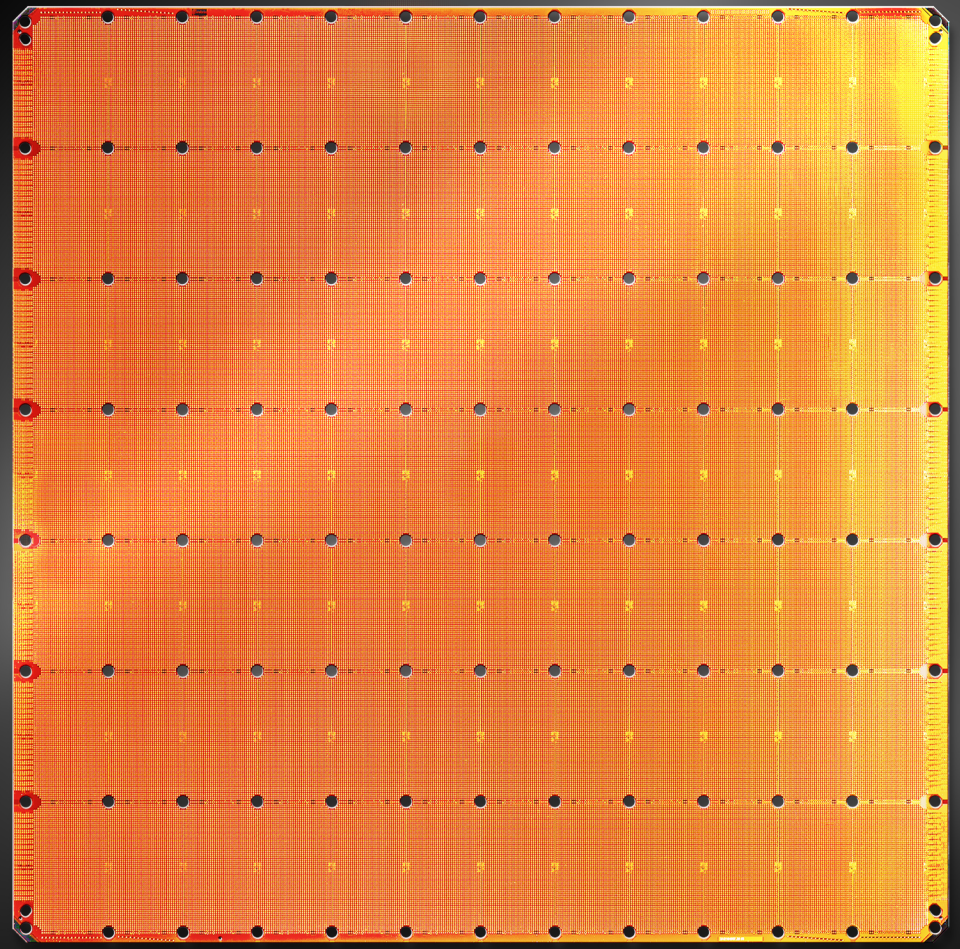
Largest Chip Ever Built

- 46,225 mm^2 silicon
- 1.2 trillion transistors
- 400,000 AI optimized cores
- 18 Gigabytes of On-chip Memory
- 9 PByte/s memory bandwidth
- 100 Pbit/s fabric bandwidth



cerebras

Cerebras Wafer Scale Engine



Cerebras WSE

1.2 Trillion Transistors

46,225 mm² Silicon

Largest GPU

21.1 Billion Transistors

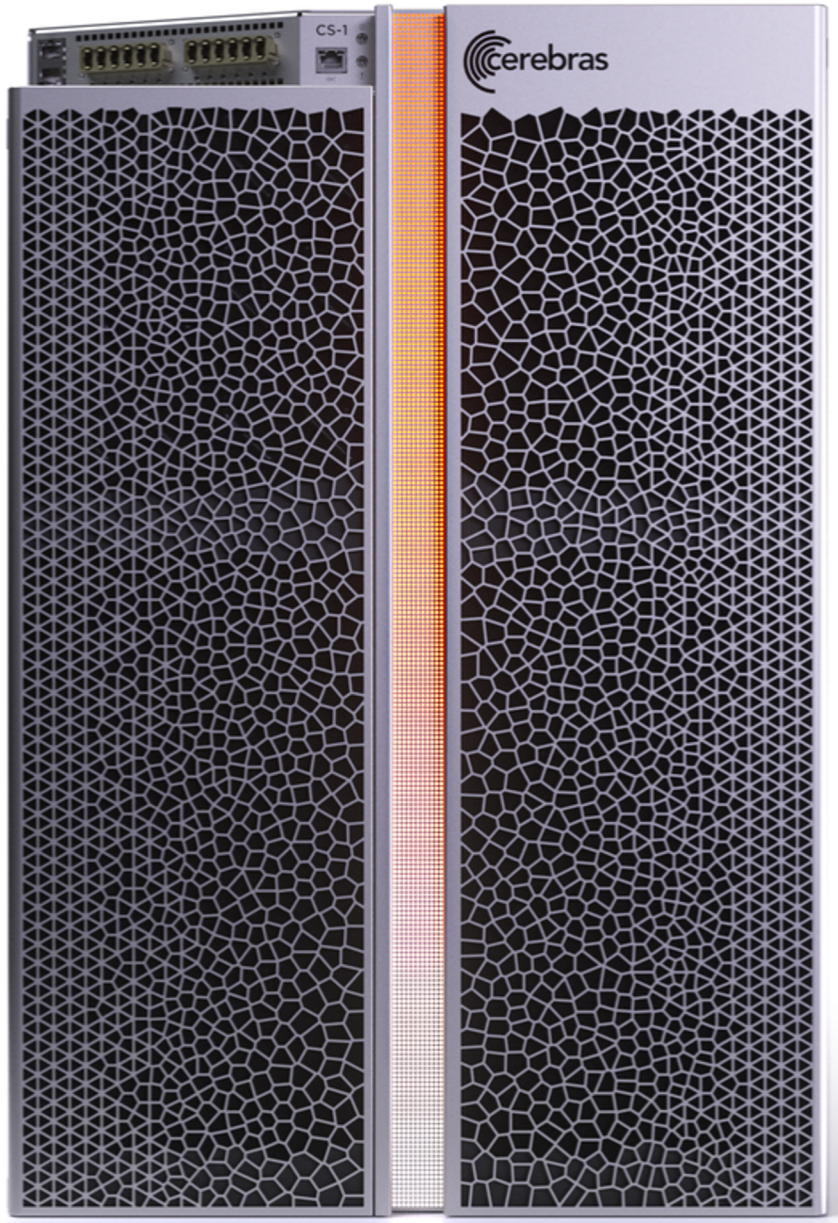
815 mm² Silicon

The Cerebras CS-1

A **full solution** in a single system:

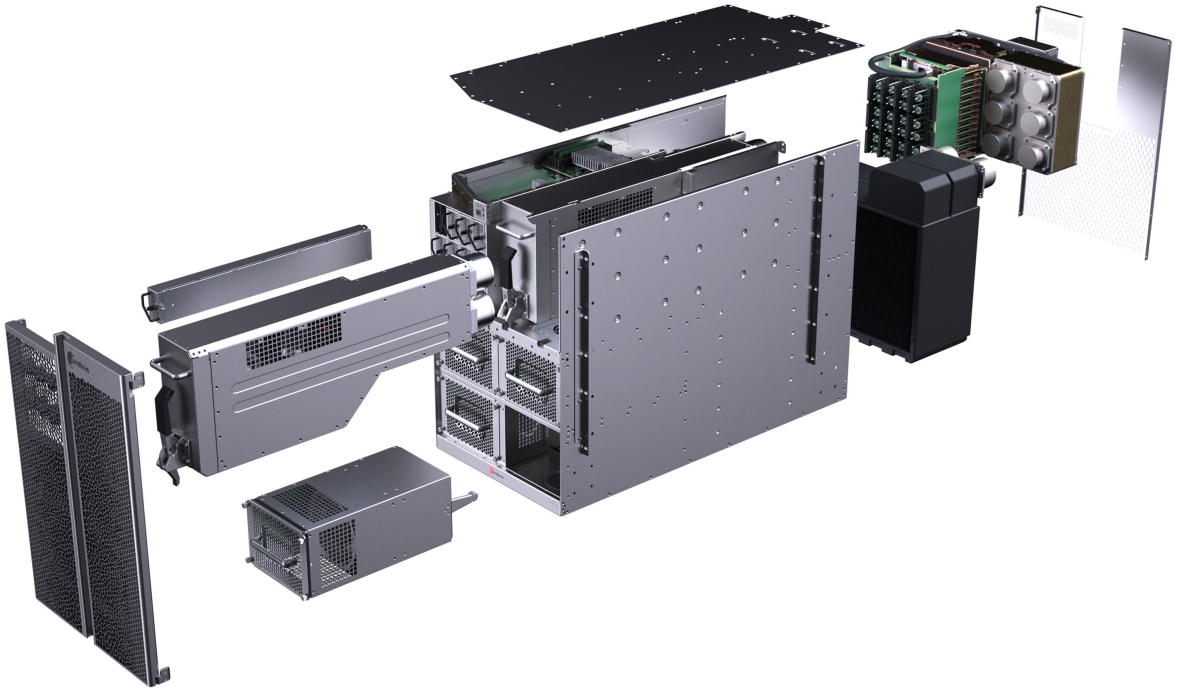
- Powered by the WSE
- Program with TF and other frameworks
- Installs in standard datacenter rack

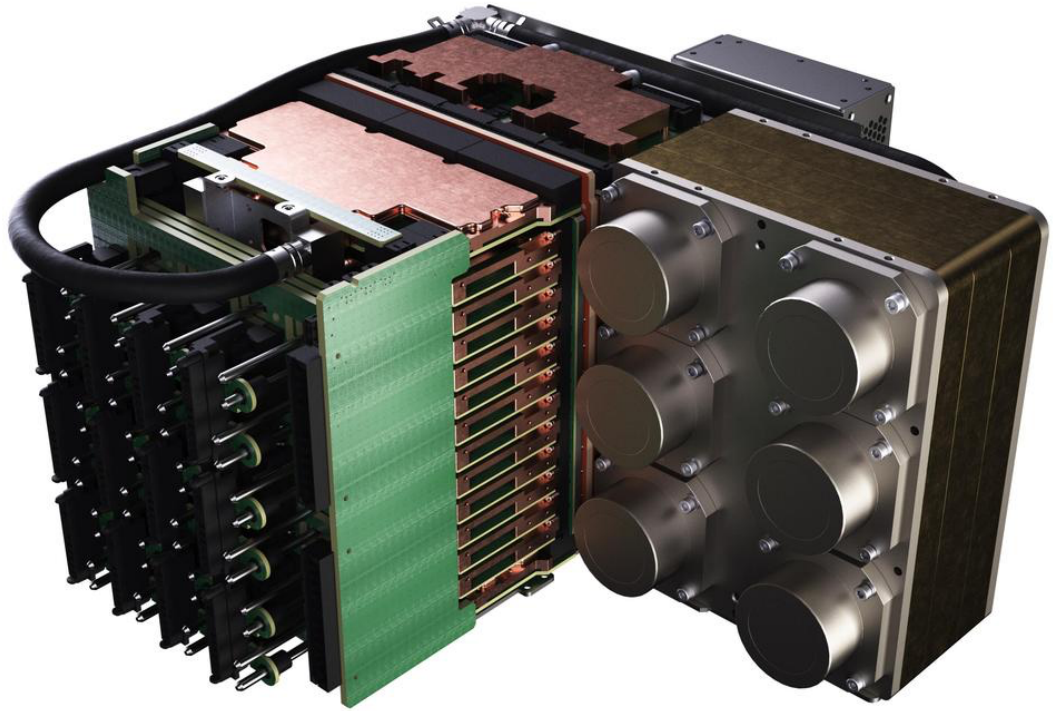
The world's most powerful AI computer



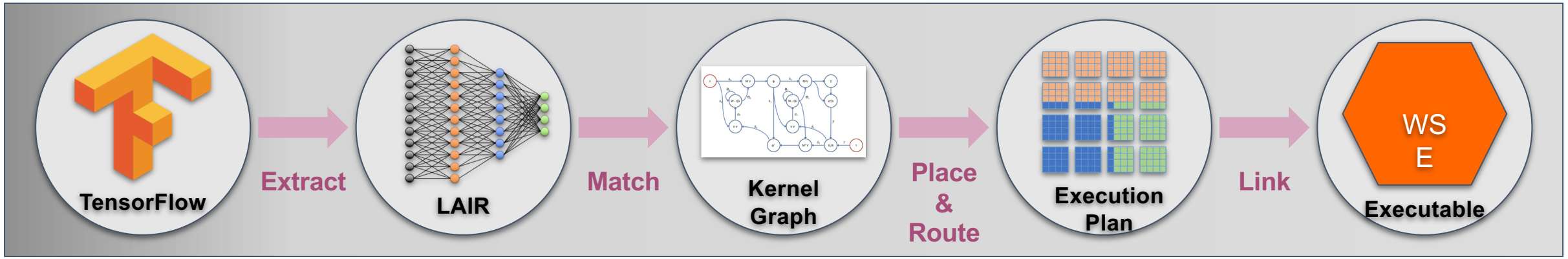
cerebras

CS-1 Supercomputer Hardware





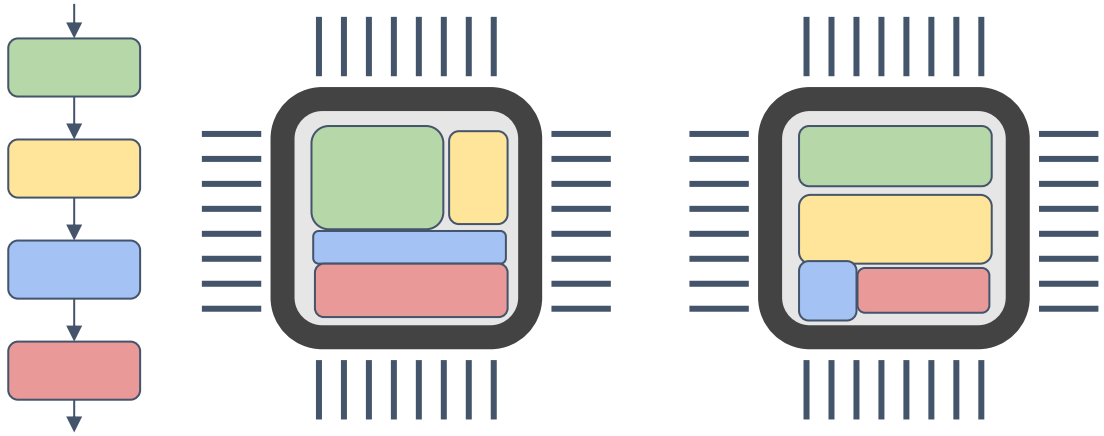
Cerebras Software Stack



- **Extract** - Convert TF via XLA to Cerebras graph IR
- **Match** - Obtain graph covering of Cerebras graph with Cerebras kernels
- **Place & Route** - Assign kernels to WSE regions and connect them up
- **Link** - Create executable output

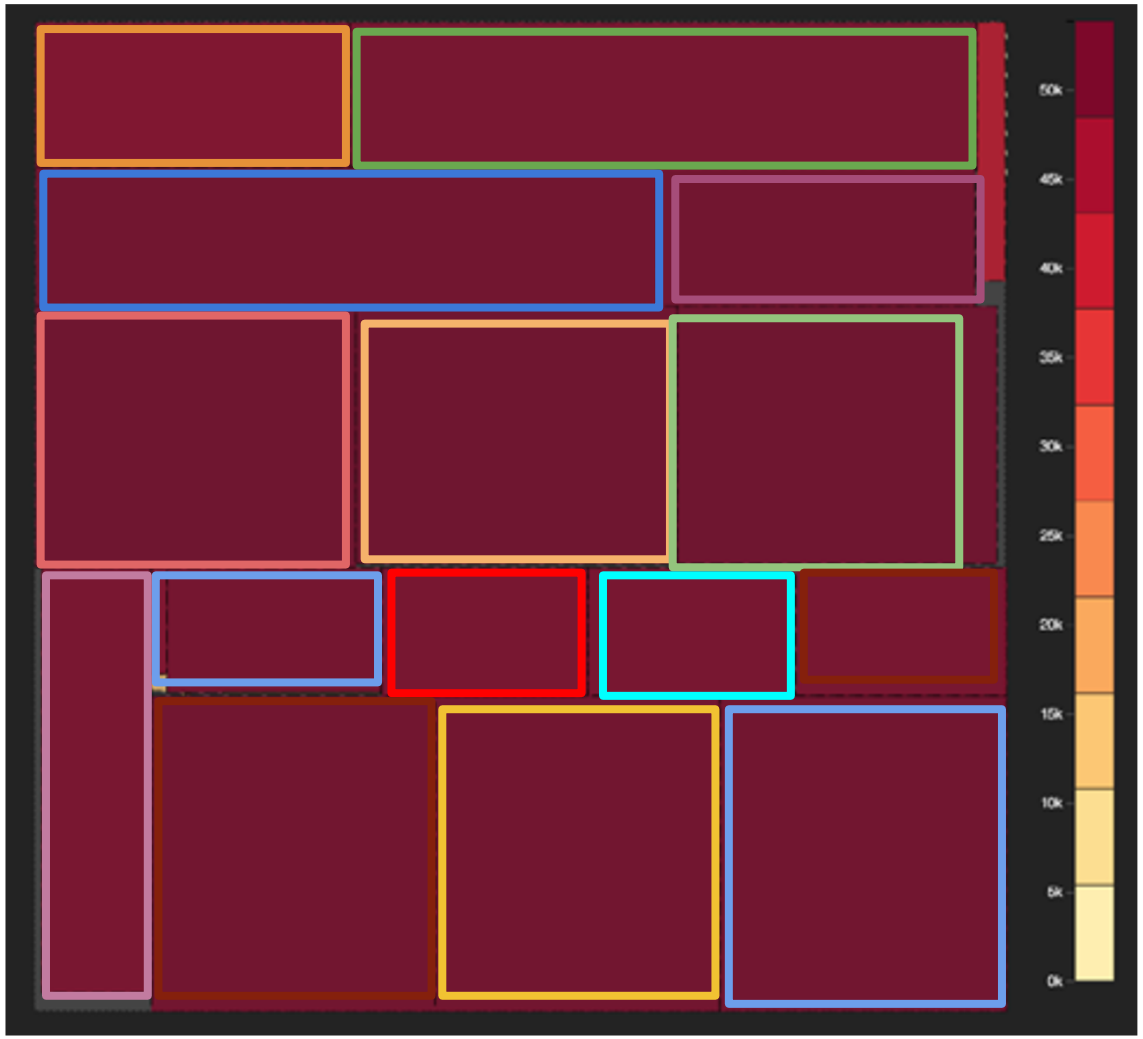
Size, Place, and Route

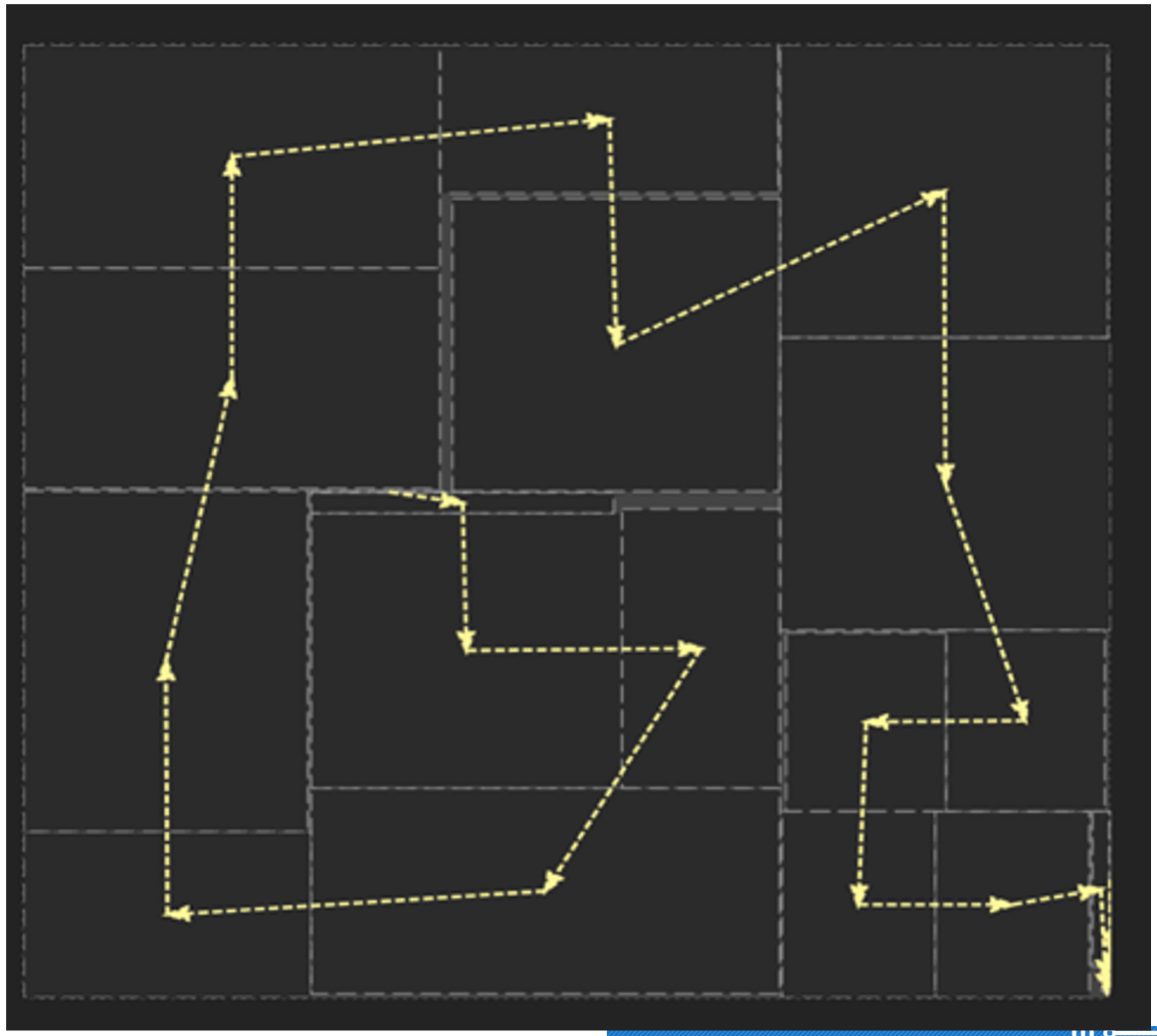
- Cerebras WSE - Spatial compute engine
- Freedom to pick and choose size (# cores) and location
- Challenge: Lot of free variables!



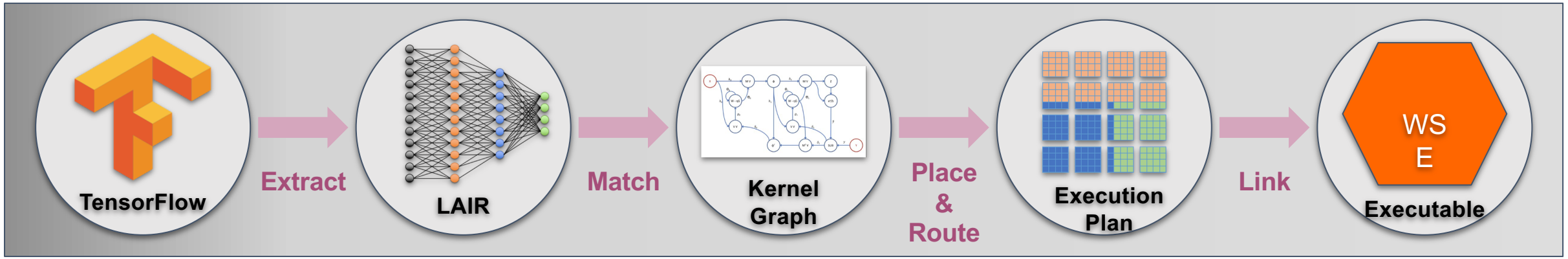
- Global optimization problem
- Variables
  - Area & loc on WSE for each kernel
- Constraints
  - Fit compute within WSE cores
  - Fit memory within WSE RAMs
  - Reduce communication
- Objective function
  - Maximize throughput, utilization
  - Minimize latency

Size, Place and Route





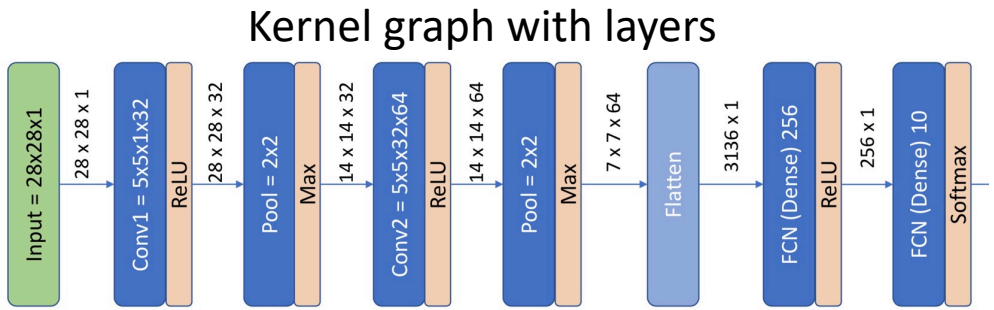
Cerebras Place&Route and EDA

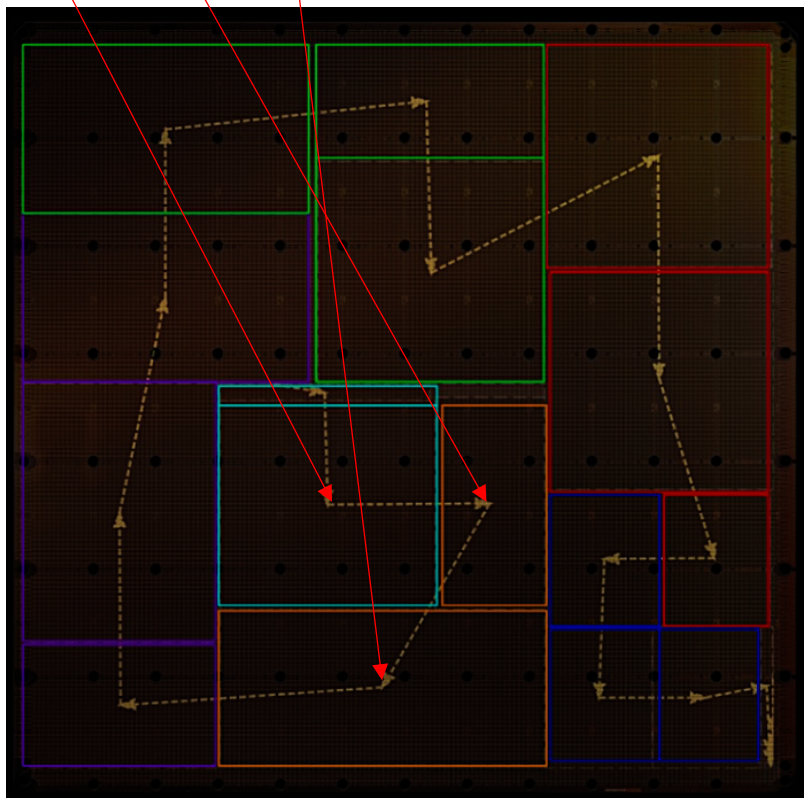


- Cerebras place and route looks and feels like the traditional ASIC/FPGA mapping problem
- But it is fundamentally different
  - Custom programmable HW fabric for data center training/inference
  - New source languages (TensorFlow, Pytorch)
  - New optimization objectives of throughput and latency
  - Compile in seconds to compete with CPU/GPUs
- Need to apply our collective knowledge to solve an *"EDA"* mapping problem that has never been solved before

Formalizing a Neural Networks as an EDA Problem

- Each layer is a program with nested loops.
- Parallelization maps loops to 2D core mesh.
- Choose mapping strategy for each layer.
- Strategy determines
  - layer size
  - communication pattern
  - memory usage
  - I/O protocol
- Network is a graph communicating layers.





Layers mapped on Wafer Scale Engine

The Performance of a Single Layer

Each layer is placed as a rectangle has **free variables**: splits

Splits define:

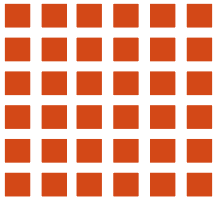
- Rectangle's dimensions (w, h)
- Performance (Δt, the smaller the faster)
- Memory consumption (should be within the core's limits)
- I/O Protocol

Overall Δt is the max of the layer's Δt

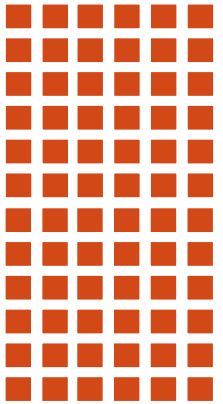
```
convperf(H, W, R, S, C, K; h', w', c', k') = {
  height = h'*w'*(k'+1)
  width  = 3*c'
  time   = ceil(H/h') * ceil(W/w') * ceil(C/c') * ceil(K/k') * R * S
  memory = C/c' * K/k' * R * S + (W+S-1)/w' * (H+R-1)/h' * K/k'
  xproto = tuple(h' w' c')
  yproto = tuple(k' w' min(c' k'))
 }
```

6x3 split:
~2X Δt (2X slower)
~2X memory

6x6 split

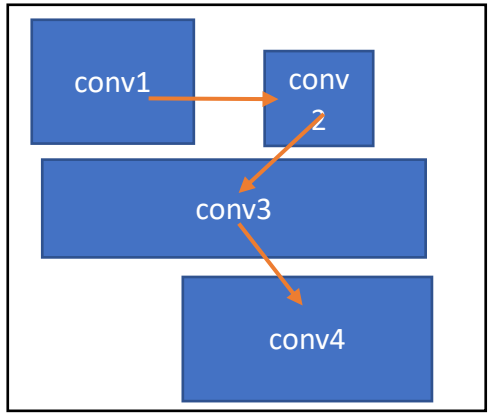


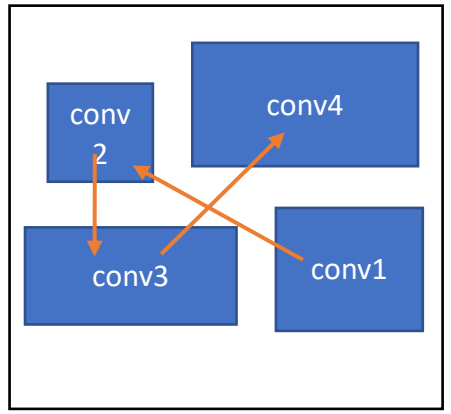
6x12 split:
½ Δt (2X faster)
~1/2 memory



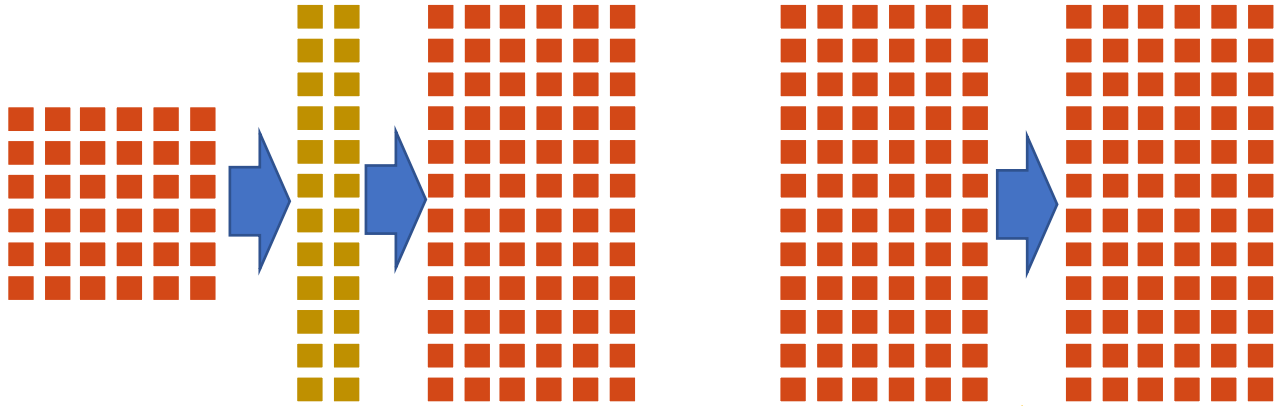
Layer Interactions: congestion and I/O mismatch

- Communication busses between layers are routed over the Network-on-Chip.
  - May case congestion
- We approximate it with the distance between kernels (wire length)





- I/O protocol mismatch changes the shape of the kernels
- We abstracted it into a split mismatch penalty



Adapter needed
To adjust h=6 to h=12

No adapter needed

cerebras

Optimization Objective

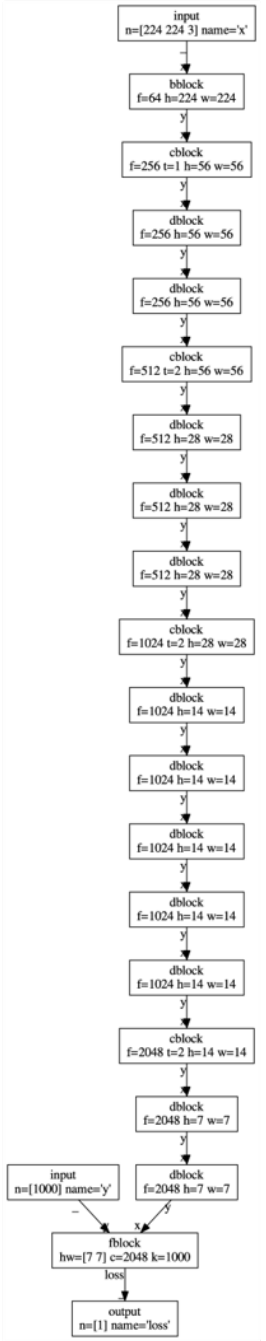
- Competition uses a simplified version of our commercial objective function:

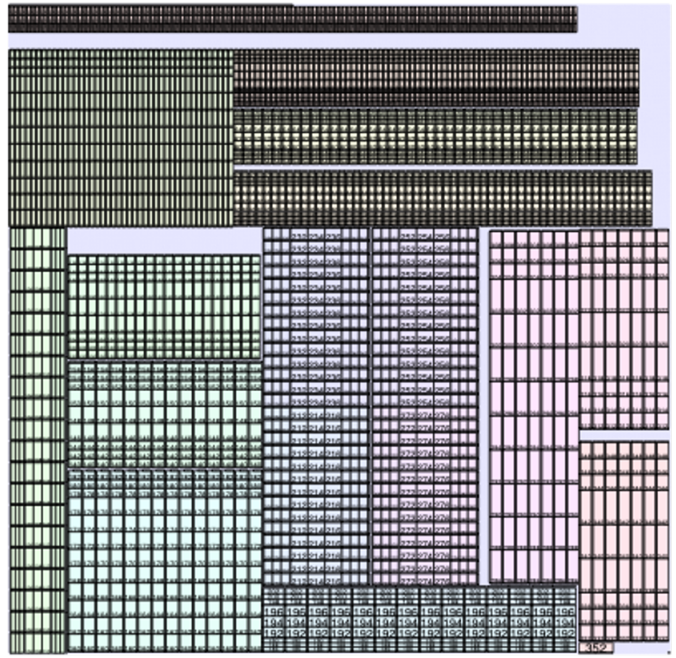
- Minimize a weighted sum of:
  - Pipeline Time
  - Routing Distance (L1 distance between layers)
  - Protocol Compatibility
  - Optimizer Runtime

- Subject to:
  - Memory footprint
  - Non-overlapping rectangles
  - Total processor size

Benchmark networks

- Convolution networks for image recognition (like resnet)
- Scientific and biomedical networks (like U-net)
- Networks used in language modelling
- Multiple corners of the cost function (variable weights)

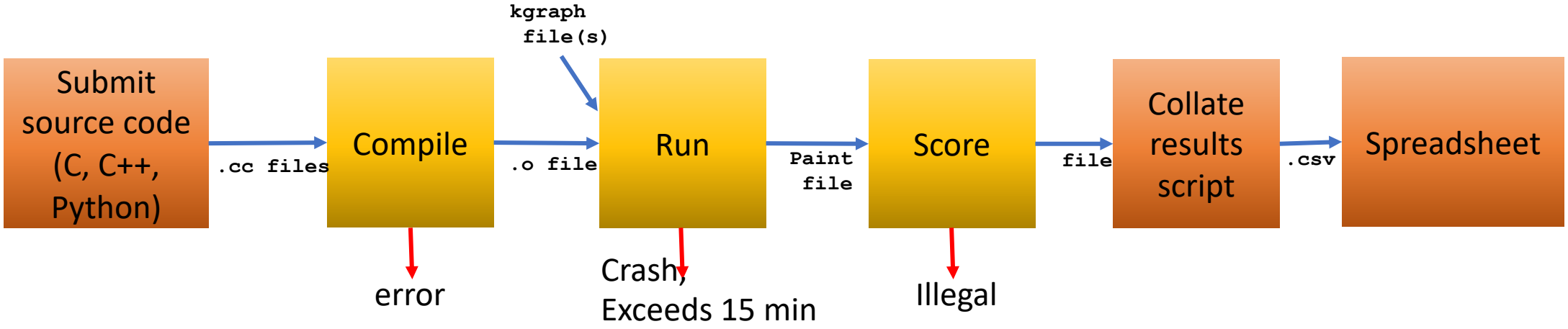




Meet the Teams in the 2020 ISPD Contest!

Team	Affiliation	Country	Members
CUPID	The Chinese University of Hong Kong	Hong Kong	Zhuolun He, Peiyu Liao, Ran Chen, Siting Liu, Yuzhe Ma, Bei Yu
Deep Learning is a Scam	Binghamton University, NY	USA	Mohammad Khasawneh, Patrick Madden
GigaPlacer	Xidian University, Giga-DA	China	Benzheng Li, Qi Du, Dingcheng Liu, Jingchong Zhang, Likang Tao, Nan Li, Gengjie Chen
ISPD_AI_Team	Fuzhou University, Nanjing Normal University	China	Long Sun, Xiwen Chang, Huihong Peng, Longkun Guo, Xiaoyan Zhang
NTU428	National Taiwan University	Taiwan	Yu-Neng Wang, Pei-Wei Chen, Yu-An Shih, Yu-Ching Huang, Yao-Wen Chang
PosPlace	Pohang University of Science and Technology	Korea	Sunghye Park, Jongho Yoon, Daeyeon Kim, Sung-Yun Lee, Minhyuk Kweon
RippleWafer	The Chinese University of Hong Kong	Hong Kong	Bentian Jiang, Jingsong Chen, Jinwei Liu, Xiaopeng Zhang, Fangzhou Wang, Lixin Liu, Evangeline F.Y. Young.
Sizeisnotaproblem	University of Texas at Dallas, TX	USA	Dinesh Bhatia, Mark Ripley Sears, P. Goswami, Masoud Shahshahani
TClab	National Tsing Hua University	Taiwan	Po-Yu Chou, Ting-Yu Chen, Tzu-Chuan Lin, Meng-Chien LinTing-Chi Wang
vdalab	National Chiao Tung University	Taiwan	Cheng-Yu Chiang, Po-Yang Chen, Yang Lu, Wing-Yee Lau, Chu-Wen Ho, Li-Yu Lin, Jun-Wei Huang

Benchmarking Method and Rules



- Teams submit via web-interface of the DOMJudge tool, running on AWS
  - Mixed bag of experiences
- Benchmarks:
  - 1 pipecleaner design
  - 8 public benchmarks
  - +12 secret benchmarks
- Every design twice:
  - Once with DeltaT:wirelength:Adaptercost of 1:1:1, which mainly values DeltaT
  - Once with ‘balanced costs’ 1:10:100 that accounts better for wirelength and adapter cost

20 benchmark kgraphs

		Team 1	Team 2	Team 3	Team 4	Team 5	Team 6	Team 7	Team 8	Team 9	Team 10	Best of 10
compact-balanced-router	A	8953	9117	10429	12012	12443	16595	121748	887214	14251	14252	8953
2x-compact-balanced-router	B	16536	16791	18214	20573	22077	31389	196510	887361	25722	24235	16536
2x-full-balanced-router	C	16738	16746	17659	21760	19844	35349	61838	1148192	17841	25346	16738
full-balanced-router	D	9108	8949	9586	11197	11274	16657	79259	1147467	14554	13466	8949
compact-adapter-router-limited	E	11503	11541	13670	17664	15614	20893	129335	889575	15847	15757	11503
2x-compact-adapter-router-limited	F	20320	20680	23775	29701	26984	48580	223854	890945	31798	26832	20320
2x-full-adapter-router-limited	G	24876	26143	28047	51538	30503	72042	121610	1164423	32699	36819	24876
full-adapter-router-limited	F	13850	13926	16493	28532	17634	38536	99886	1155678	24857	20033	13850
resnet75-totem-deltat	G	13650	13639	15004	18234	18935	24126	92645	888181	18128	19144	13639
resnet75-conv-deltat	H	15042	14342	16781	23918	26674	33755	118703	1149819	22234	23956	14342
resnet50-lean-totem-deltat	I	348	486	609	655	2031	5533	4936	20331	562		348
resnet50-lean-conv-deltat	J	495	852	891	4680	2912	3814	34459	56934	1320		495
unet-conv-deltat-t	K	556570	569147	594753	578909	1624738	1072026	4102534	6639181			556570
fcnet-lite-deltat	L	847	849	932	1401	1614	3419	13229	6701	930		847
resnet75-totem-balanced	M	20494	21730	23296	43300	43738	50610	168084	898620	35948	25442	20494
resnet75-conv-balanced	N	33988	33877	43388	112260	101942	150684	498726	1197300	68768	44022	33877
resnet50-lean-totem-balanced	O	2470	4993	5427	6464	18442	9290	43122	23470	6030		2470
resnet50-lean-conv-balanced	P	6646	9053	19782	44654	28307	25194	281086	63668	25933		6646
unet-conv-balanced	Q	793876	739040	794940	993876	3232136	1335840	5158504	7031620			739040
fcnet-lite-balanced	R	3495	5582	5087	7659	13162	9299	88143	13068	4892		3495
												0
Average score (absolute)		78490	76874	82938	101449	263550	150182	581910	1307987	20128	24109	20128

cerebras

Download graphs at: https://drive.google.com/open?id=1MSRn754fdBAX98atF6KMm7J45Ouf87gl

Normalized Results on the Designs

	Team 1	Team 2	Team 3	Team 4	Team 5	Team 6	Team 7	Team 8	Team 9	Team 10
compact-balanced-router	1.0000	1.0183	1.1648	1.3417	1.3898	1.8536	13.5985	99.0968	1.5917	1.5918
2x-compact-balanced-router	1.0000	1.0155	1.1015	1.2442	1.3351	1.8983	11.8840	53.6632	1.5555	1.4656
2x-full-balanced-router	1.0000	1.0005	1.0550	1.3000	1.1856	2.1119	3.6945	68.5990	1.0659	1.5143
full-balanced-router	1.0178	1.0000	1.0712	1.2512	1.2598	1.8614	8.8570	128.2265	1.6264	1.5048
compact-adapter-router-limited	1.0000	1.0033	1.1883	1.5356	1.3573	1.8163	11.2436	77.3342	1.3776	1.3698
2x-compact-adapter-router-limited	1.0000	1.0177	1.1701	1.4617	1.3280	2.3908	11.0167	43.8468	1.5649	1.3205
2x-full-adapter-router-limited	1.0000	1.0510	1.1275	2.0718	1.2262	2.8961	4.8887	46.8100	1.3145	1.4801
full-adapter-router-limited	1.0000	1.0055	1.1908	2.0600	1.2732	2.7824	7.2120	83.4424	1.7947	1.4464
resnet75-totem-deltat	1.0008	1.0000	1.1001	1.3369	1.3883	1.7689	6.7927	65.1207	1.3291	1.4036
resnet75-conv-deltat	1.0488	1.0000	1.1701	1.6677	1.8599	2.3536	8.2766	80.1715	1.5503	1.6703
resnet50-lean-totem-deltat	1.0000	1.3966	1.7500	1.8822	5.8362	15.8994	14.1839	58.4224	1.6149	
resnet50-lean-conv-deltat	1.0000	1.7212	1.8000	9.4545	5.8828	7.7051	69.6141	115.0182	2.6667	
unet-conv-deltat-t	1.0000	1.0226	1.0686	1.0401	2.9192	1.9261	7.3711	11.9287		
fcnet-lite-deltat	1.0000	1.0024	1.1004	1.6541	1.9055	4.0366	15.6187	7.9115	1.0980	
resnet75-totem-balanced	1.0000	1.0603	1.1367	2.1128	2.1342	2.4695	8.2016	43.8480	1.7541	1.2414
resnet75-conv-balanced	1.0033	1.0000	1.2808	3.3138	3.0092	4.4480	14.7217	35.3426	2.0299	1.2995
resnet50-lean-totem-balanced	1.0000	2.0215	2.1972	2.6170	7.4664	3.7611	17.4583	9.5020	2.4413	
resnet50-lean-conv-balanced	1.0000	1.3622	2.9765	6.7189	4.2593	3.7909	42.2940	9.5799	3.9020	
unet-conv-balanced	1.0742	1.0000	1.0756	1.3448	4.3734	1.8075	6.9800	9.5145		
fcnet-lite-balanced	1.0000	1.5971	1.4555	2.1914	3.7660	2.6607	25.2197	3.7391	1.3997	
Average 20 designs	**1.0072**	**1.1648**	**1.3590**	**2.3800**	**2.7578**	**3.5119**	**15.4564**	**52.5559**	**1.7599**	**1.4423**
average % worse vs the best	**1%**	**16%**	**36%**	**138%**	**176%**	**251%**	**1446%**	**5156%**	**76%**	**44%**
Ranking	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**

Team Presentations (Alphabetical order)

- Cupid
- Gigaplacer
- ISPD_AI_Team
- NTU428
- Ripplewafer

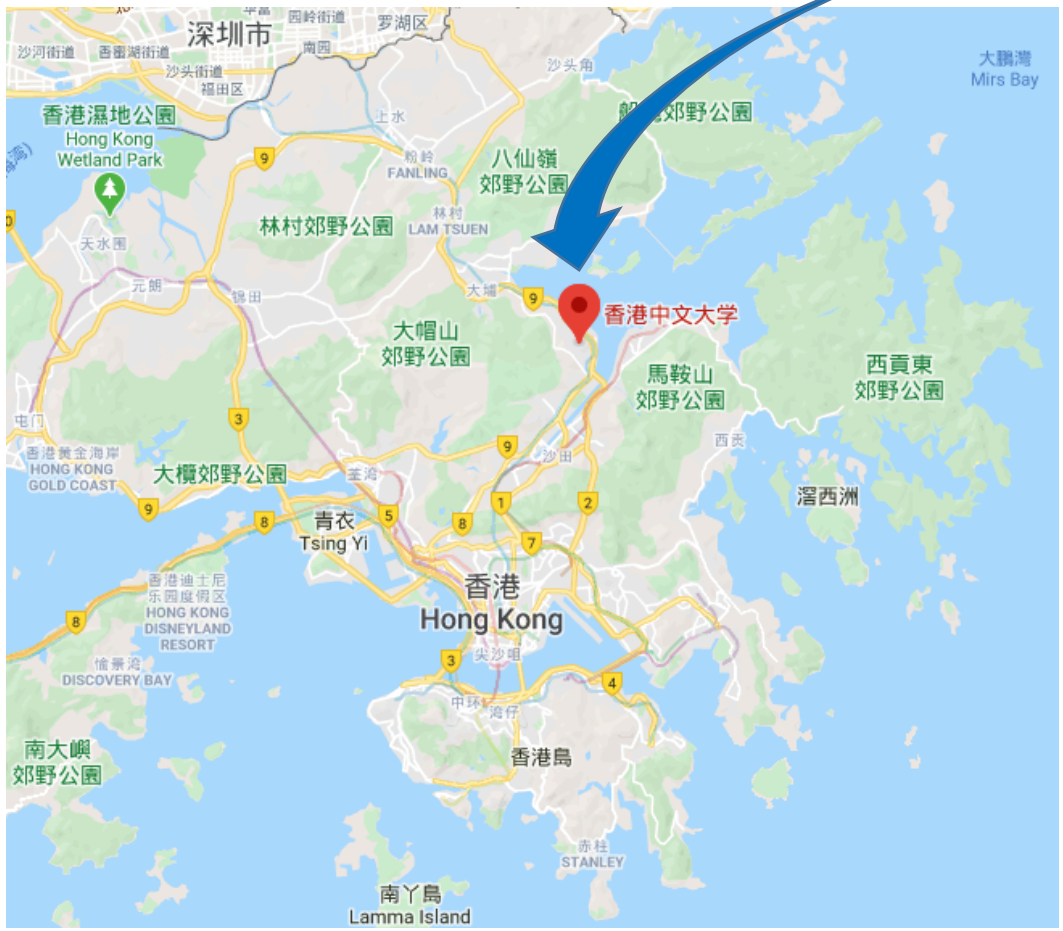
NTU428

National Taiwan University

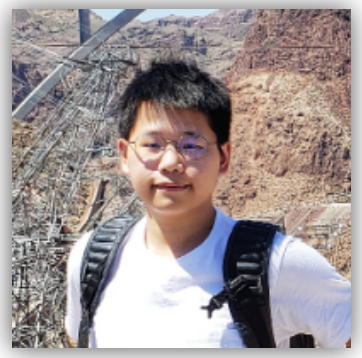
2020 ISPD Contest
Team: RippleWafer
The Chinese University of Hong Kong

Our School

RippeWafer



Teammates



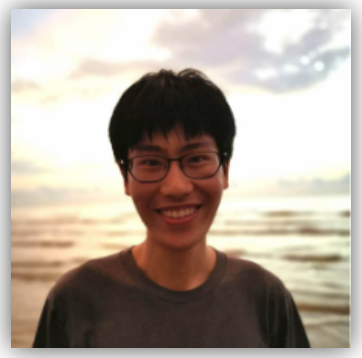
Bentian Jiang
Year-3 PhD student,
CSE Dept.

Jingsong Chen
Year-3 PhD student,
CSE Dept.

Jinwei Liu
Year-2 PhD student,
CSE Dept.

Xiaopeng Zhang
Year-2 PhD student,
CSE Dept.

Lixin Liu
Year-1 PhD student,
CSE Dept.



Fangzhou Wang
Year-1 PhD student,
CSE Dept.

Supervisor

Evangeline F.Y. Young
Professor, CSE Dept.

Objectives

The objective of this problem is to minimize the weighted summation of:

- Maximum compute time
- Total wire length
- Total adapter cost

Subject to all legality constraints

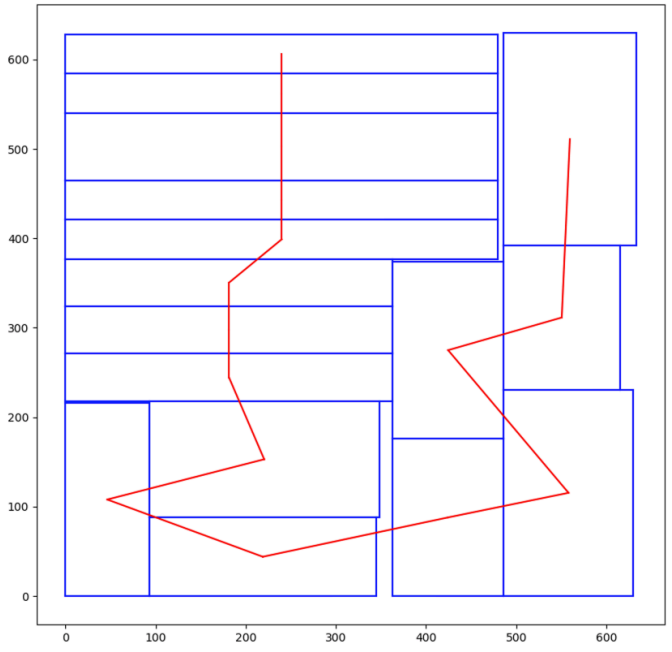
Some Insights

- Conventional floorplanning methods may not be suitable
  - Simulated annealing, slicing floorplan, mosaic floorplan
- How to determine best kernel shapes efficiently is the key for all approaches
- Customization is essential
  - Searching efficiency should be considered
  - The topology of kernel graph should be honored
  - Optimize adapter cost in a unified way
- What's next?
  - Collaborations with predecessor/successor stages
  - Simultaneously kernel graph generation and placement…

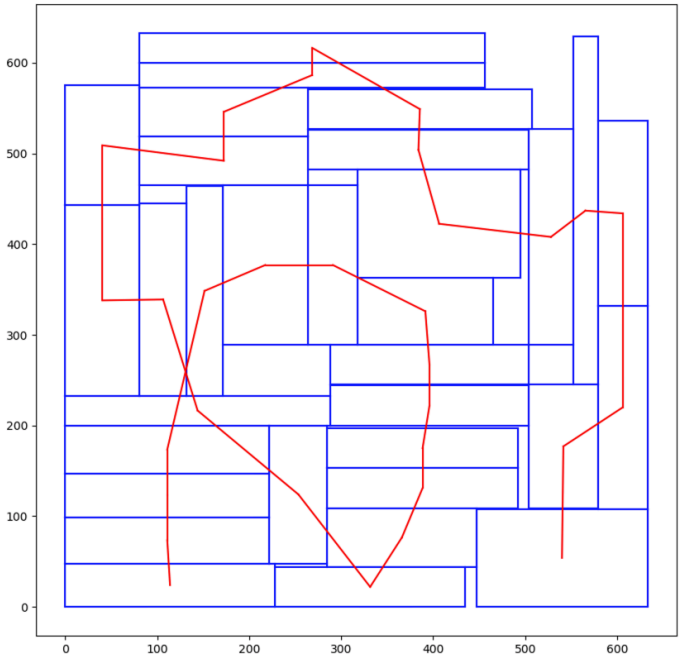
Methodology

The first method we tried:

Simulated Annealing + Deliciated Modifications



Kgraph E

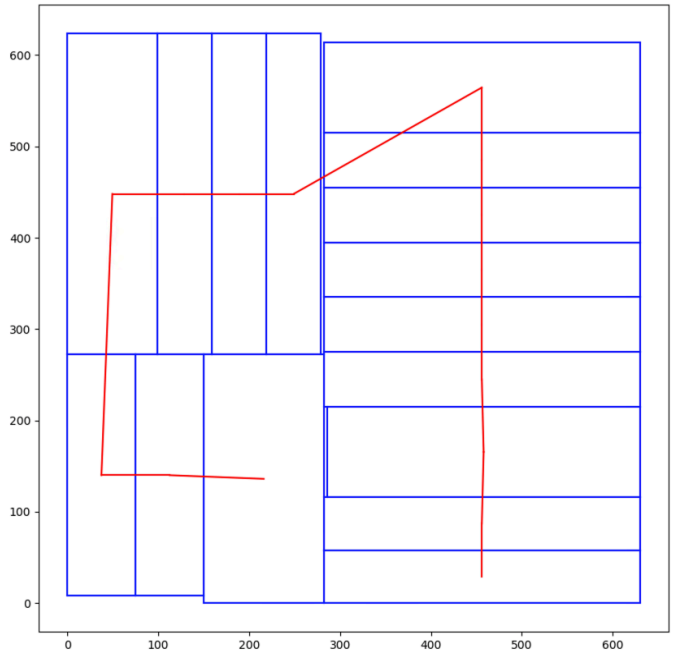


Kgraph F

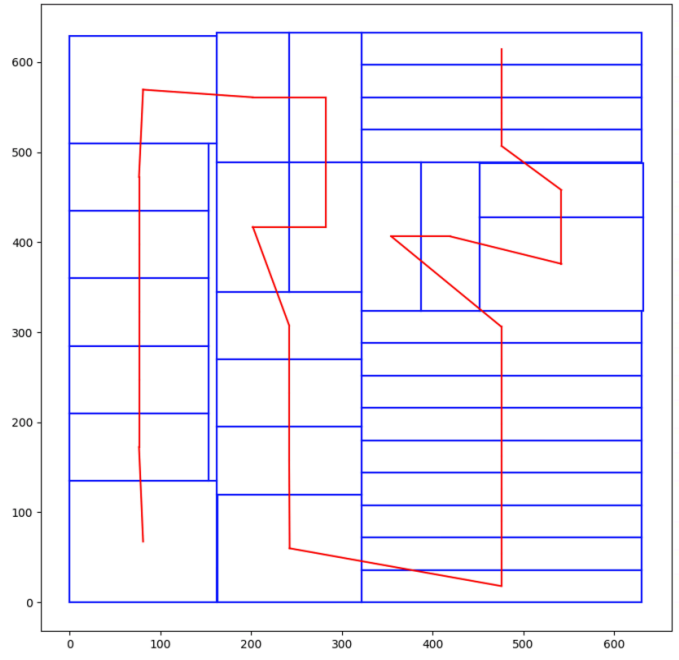
Methodology

Then we tried:

Slicing Floorplan + Deliciated Modifications



Kgraph E



Kgraph F

Some Insights

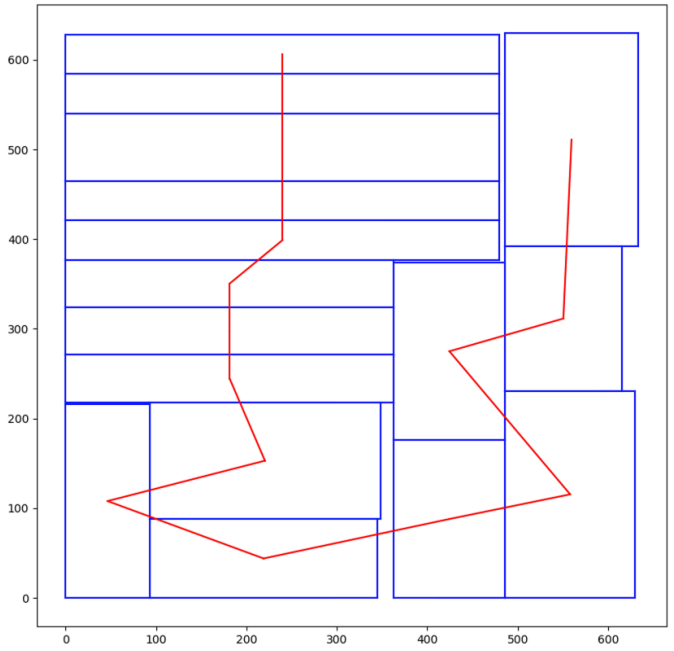
- Conventional methods for general floorplanning may not be suitable
  - Simulated annealing, slicing floorplan, mosaic floorplan
- How to determine best kernel shapes efficiently is the key for all approaches
- Customization is essential
  - Searching efficiency should be considered
  - The topology of kernel graph should be honored
  - Optimize adapter cost in a unified way
- What's next?
  - Collaborations with predecessor/successor stages
  - Simultaneously kernel graph generation and placement…

Methodology

Our final method:

- Adopt a two-level (coarse-grained and fine-grained) search scheme to find a good maximum compute time.
- For each traversed maximum compute time T:
  - Generate optimal legal candidate shapes with compute time not exceeding T for each kernel.
  - Row-based placement according to kgraph topology to minimize the wire length.
- Universal post-refinement process to improve the adapter cost.

Comparison – Kgraph E



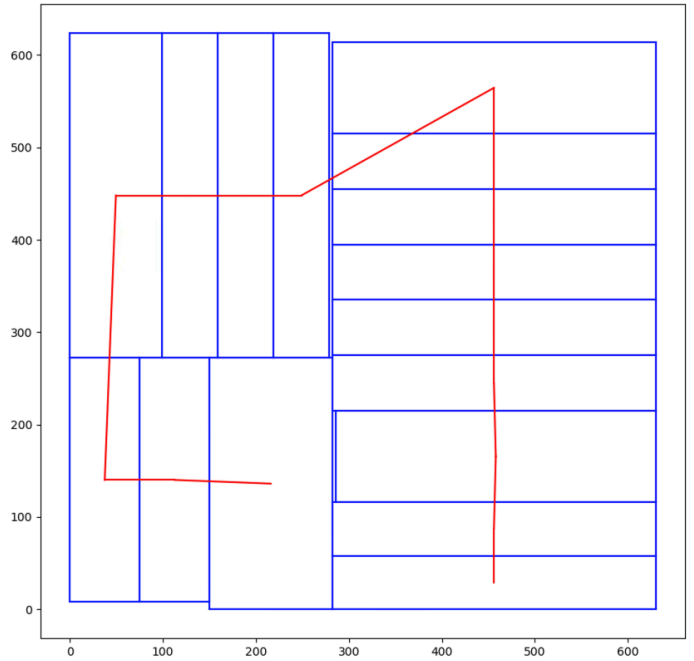
SA Placement:

Max_time: 36288

Wire_length: 2080.5

Adapter_cost: 7

Score: 57793



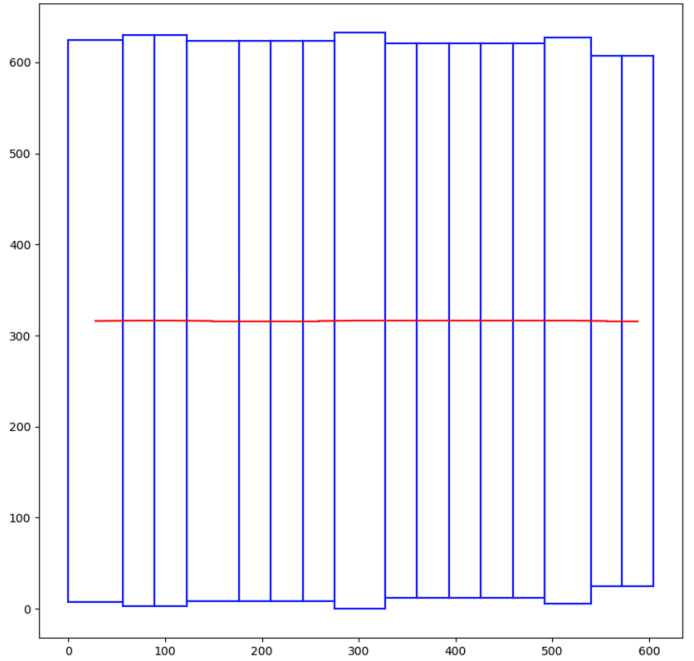
Slicing Placement:

Max_time: 35280

Wire_length: 1565

Adapter_cost: 12

Score: 52130



Our Final Method:

Max_time: 39312

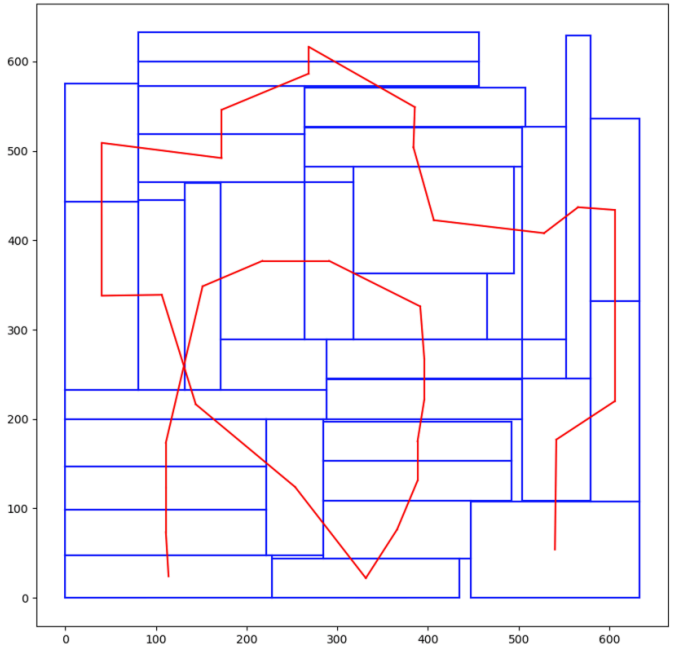
Wire_length: 562

Adapter_cost: 11

Score: 46032

Comparison – Kgraph F

RippeWafer



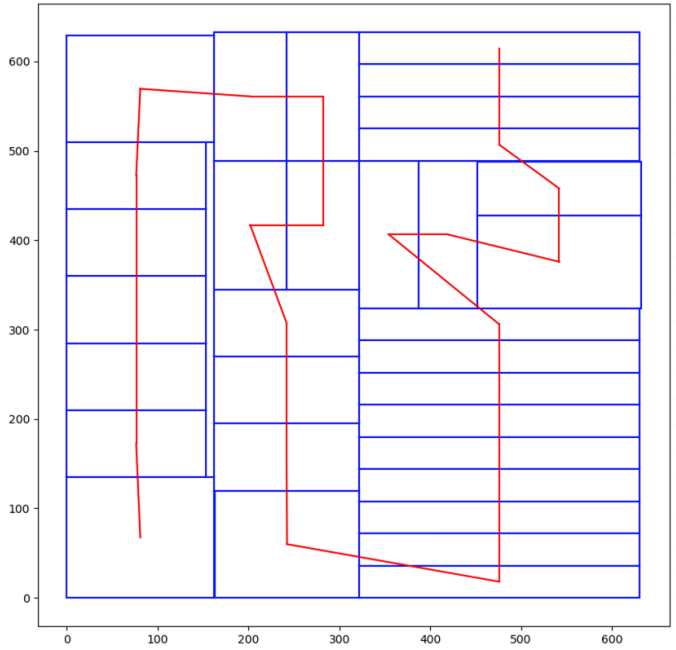
SA Placement:

Max_time: 76698

Wire_length: 3237

Adapter_cost: 15

Score: 110478



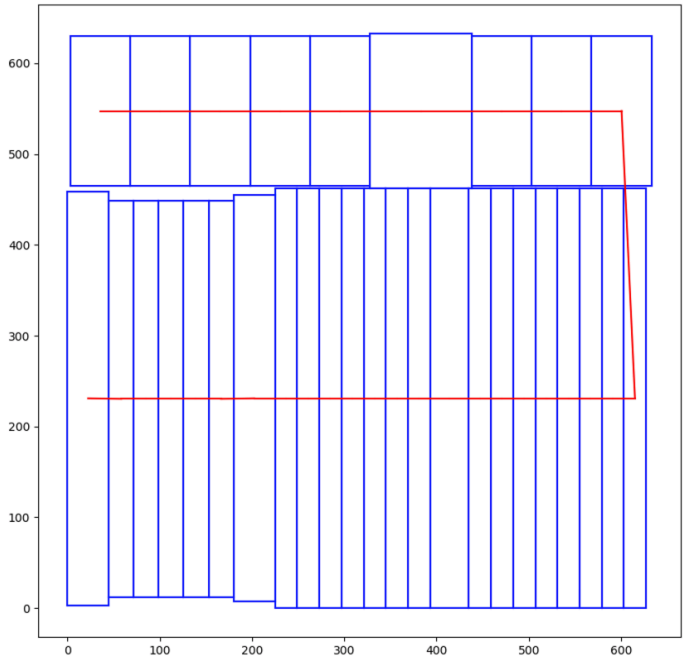
Slicing Placement:

Max_time: 65016

Wire_length: 2650.5

Adapter_cost: 18

Score: 93321



Our Final Method:

Max_time: 65170

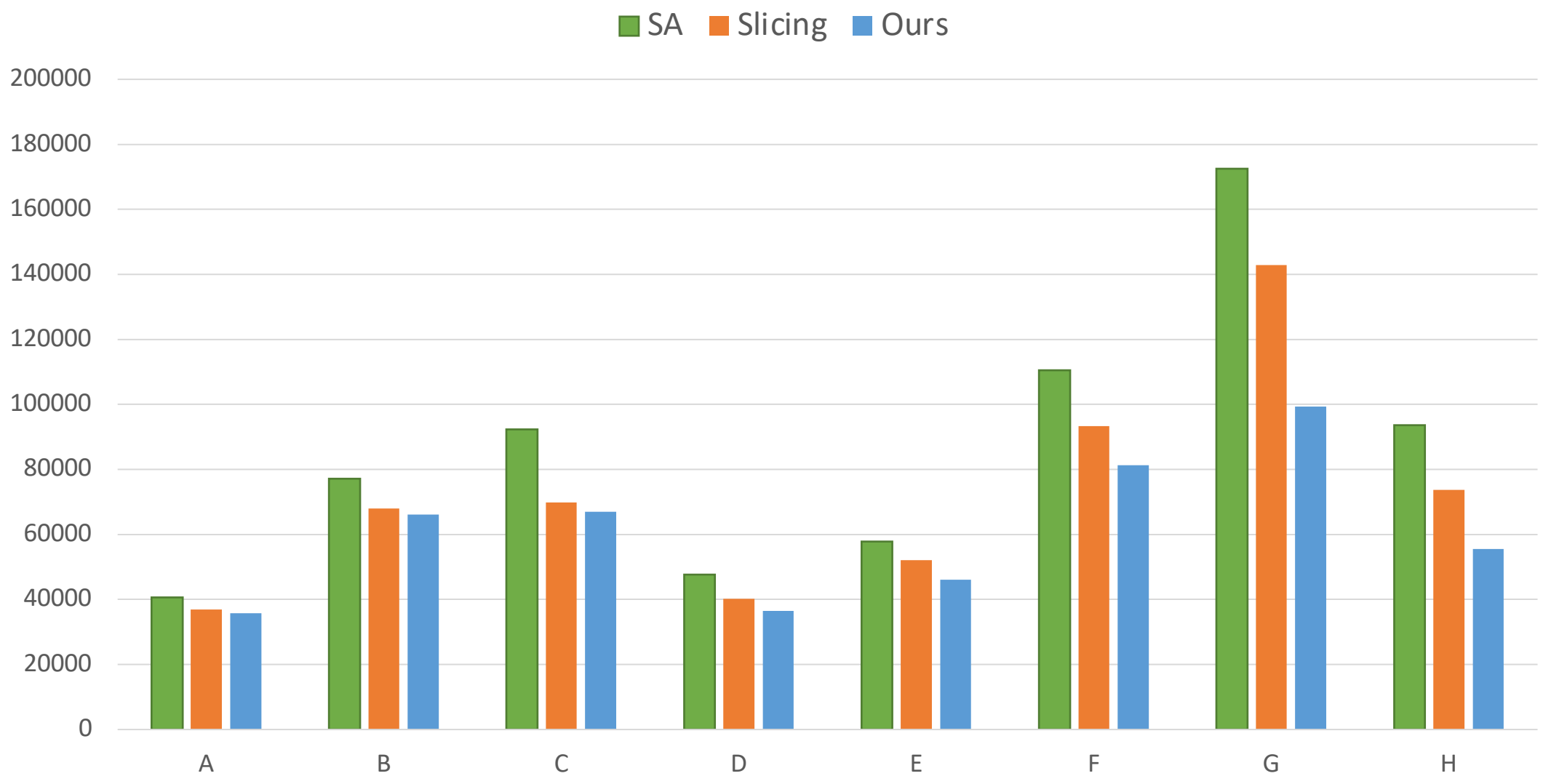
Wire_length: 1489.5

Adapter_cost: 12

Score: 81265

Case Study – Public Benchmarks

RippeWafer



Some Insights

- Conventional methods for general floorplanning may not be suitable
  - Simulated annealing, slicing floorplan, mosaic floorplan
- How to determine best kernel shapes efficiently is the key for all approaches
- Customization is essential
  - Searching efficiency should be considered
  - The topology of kernel graph should be honored
  - Optimize adapter cost in a unified way
- What's next?
  - Collaborations with predecessor/successor stages
  - Simultaneously kernel graph generation and placement…

End

Thank you!

And now
The Winners!!

2020 ACM International Symposium on Physical Design

Wafer-Scale Deep Learning Accelerator Placement Contest

THIRD PLACE

Team CUPID

Zhuolun He, Peiyu Liao, Ran Chen, Siting Liu,
Yuzhe Ma, Bei Yu
The Chinese University of Hong Kong

William Swartz
General Chair
ISPD2020

Jens Lienig
Technical Program Chair
ISPD2020

Marvin Tom, Michael James,
Vladimir Kibardin, Robby Fry,
Patrick Groeneveld,
Cerebras

2020 ACM International Symposium on Physical Design

Wafer-Scale Deep Learning Accelerator Placement Contest

SECOND PLACE

Team Gigaplacer

Benzheng Li, Qi Du, Dingcheng Liu, Jingchong Zhang
Xidian University
Gengjie Chen
Giga Design Automation

William Swartz
General Chair
ISPD2020

Jens Lienig
Technical Program Chair
ISPD2020

Marvin Tom, Michael James,
Vladimir Kibardin, Robby Fry,
Patrick Groeneveld,
Cerebras

2020 ACM International Symposium on Physical Design

Wafer-Scale Deep Learning Accelerator Placement Contest

FIRST PLACE

Team Ripplewafer

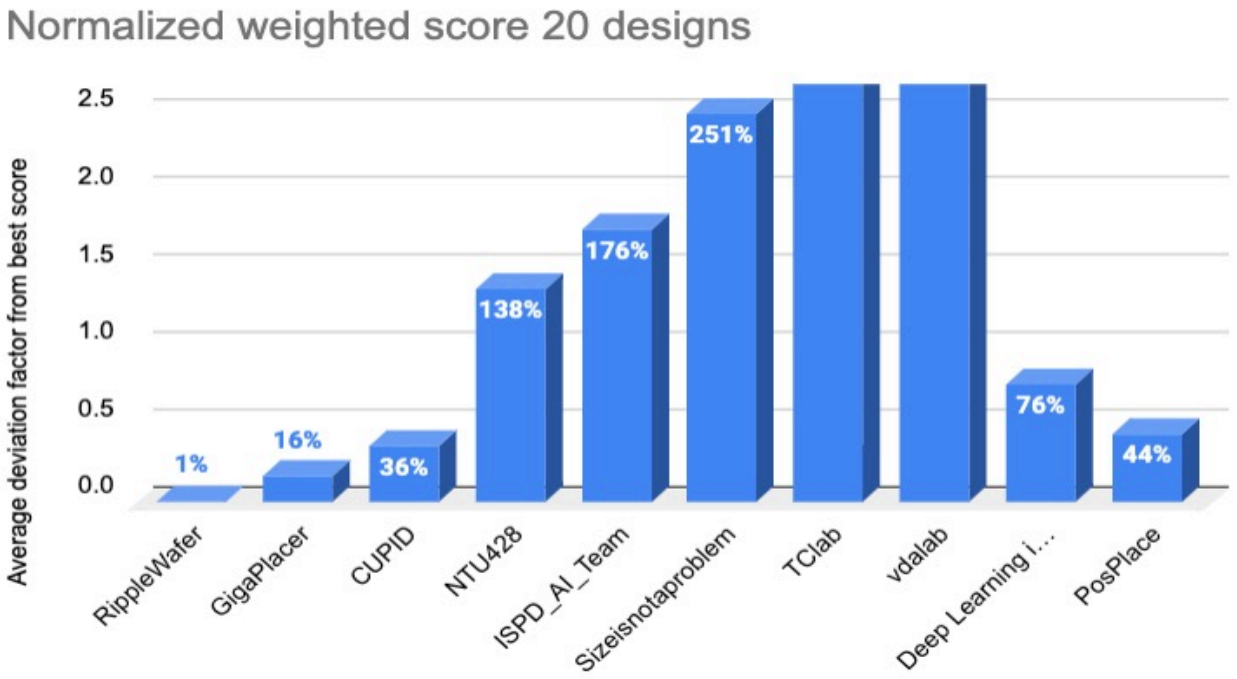
Bentian Jiang, Jingsong Chen, Jinwei Liu, Xiaopeng Zhang,
Fangzhou Wang, Lixin Liu, Evangeline F.Y. Young
The Chinese University of Hong Kong

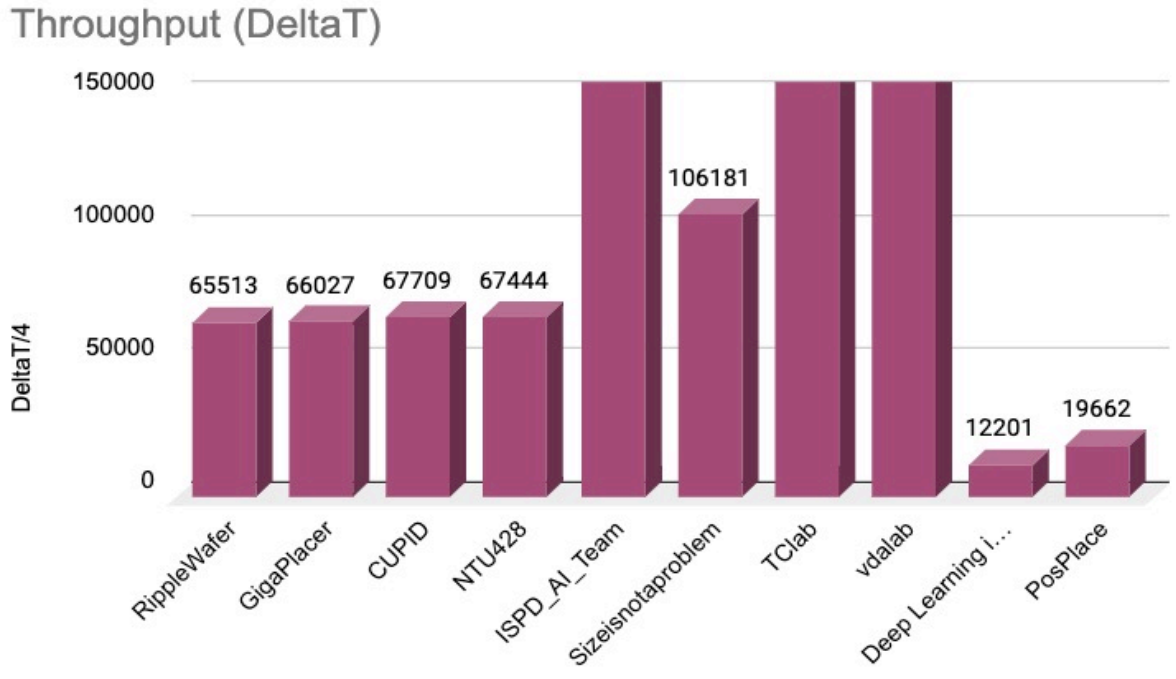
William Swartz
General Chair
ISPD2020

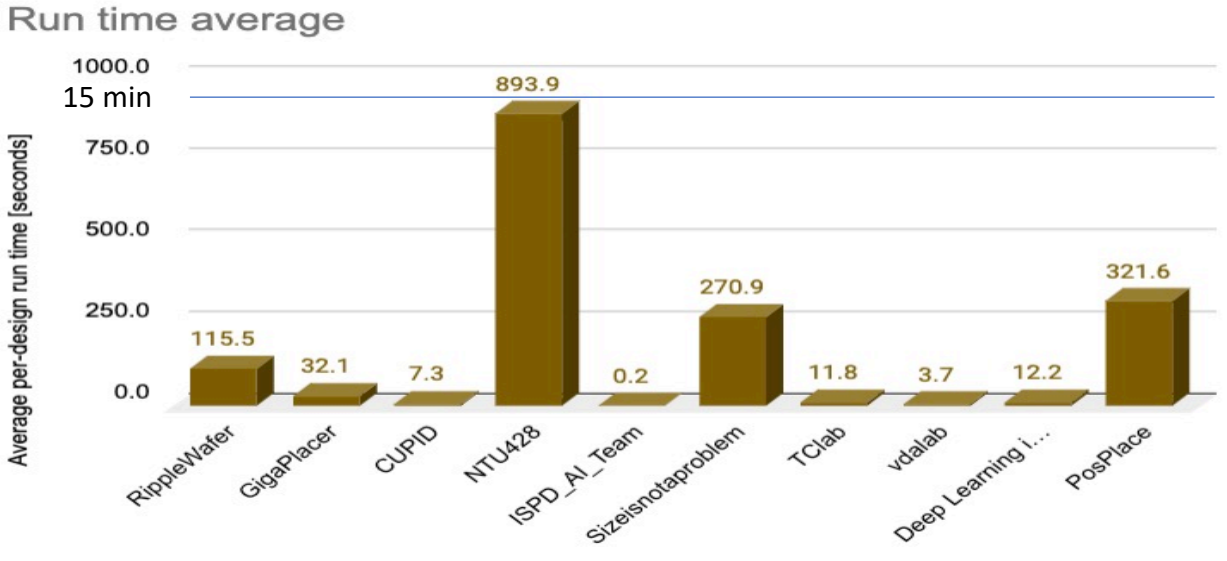
Jens Lienig
Technical Program Chair
ISPD2020

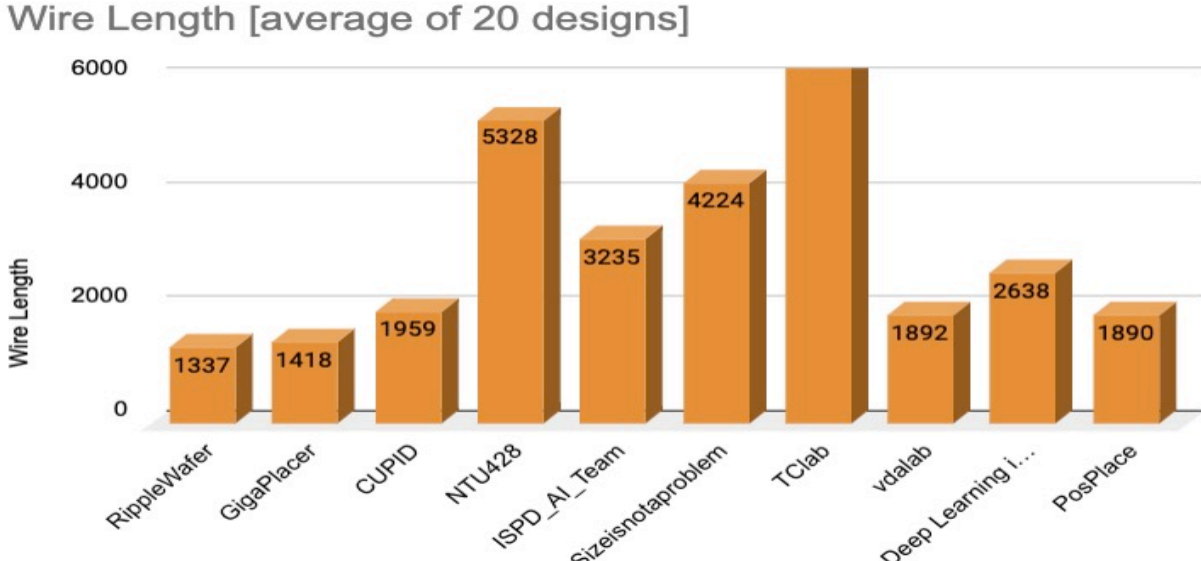
Marvin Tom, Michael James,
Vladimir Kibardin, Robby Fry,
Patrick Groeneveld,
Cerebras

Comparison over 20 kGraphs









Normalized Results per Team

	RippleWafer	GigaPlacer	CUPID	NTU428	ISPD_AI_Team	Sizeisnotaproblem	TClab	vdalab	Deep Learning is a Scam	PosPlace
compact-balanced-router	1.0000	1.0183	1.1648	1.3417	1.3898	1.8536	13.5985	99.0968	1.5917	1.5918
2x-compact-balanced-router	1.0000	1.0155	1.1015	1.2442	1.3351	1.8983	11.8840	53.6632	1.5555	1.4656
2x-full-balanced-router	1.0000	1.0005	1.0550	1.3000	1.1856	2.1119	3.6945	68.5990	1.0659	1.5143
full-balanced-router	1.0178	1.0000	1.0712	1.2512	1.2598	1.8614	8.8570	128.2265	1.6264	1.5048
compact-adapter-router-limited	1.0000	1.0033	1.1883	1.5356	1.3573	1.8163	11.2436	77.3342	1.3776	1.3698
2x-compact-adapter-router-limited	1.0000	1.0177	1.1701	1.4617	1.3280	2.3908	11.0167	43.8468	1.5649	1.3205
2x-full-adapter-router-limited	1.0000	1.0510	1.1275	2.0718	1.2262	2.8961	4.8887	46.8100	1.3145	1.4801
full-adapter-router-limited	1.0000	1.0055	1.1908	2.0600	1.2732	2.7824	7.2120	83.4424	1.7947	1.4464
resnet75-totem-deltat	1.0008	1.0000	1.1001	1.3369	1.3883	1.7689	6.7927	65.1207	1.3291	1.4036
resnet75-conv-deltat	1.0488	1.0000	1.1701	1.6677	1.8599	2.3536	8.2766	80.1715	1.5503	1.6703
resnet50-lean-totem-deltat	1.0000	1.3966	1.7500	1.8822	5.8362	15.8994	14.1839	58.4224	1.6149	
resnet50-lean-conv-deltat	1.0000	1.7212	1.8000	9.4545	5.8828	7.7051	69.6141	115.0182	2.6667	
unet-conv-deltat-t	1.0000	1.0226	1.0686	1.0401	2.9192	1.9261	7.3711	11.9287		
fcnet-lite-deltat	1.0000	1.0024	1.1004	1.6541	1.9055	4.0366	15.6187	7.9115	1.0980	
resnet75-totem-balanced	1.0000	1.0603	1.1367	2.1128	2.1342	2.4695	8.2016	43.8480	1.7541	1.2414
resnet75-conv-balanced	1.0033	1.0000	1.2808	3.3138	3.0092	4.4480	14.7217	35.3426	2.0299	1.2995
resnet50-lean-totem-balanced	1.0000	2.0215	2.1972	2.6170	7.4664	3.7611	17.4583	9.5020	2.4413	
resnet50-lean-conv-balanced	1.0000	1.3622	2.9765	6.7189	4.2593	3.7909	42.2940	9.5799	3.9020	
unet-conv-balanced	1.0742	1.0000	1.0756	1.3448	4.3734	1.8075	6.9800	9.5145		
fcnet-lite-balanced	1.0000	1.5971	1.4555	2.1914	3.7660	2.6607	25.2197	3.7391	1.3997	
Average 20 designs	**1.0072**	**1.1648**	**1.3590**	**2.3800**	**2.7578**	**3.5119**	**15.4564**	**52.5559**	**1.7599**	**1.4423**
average % worse vs the best	**1%**	**16%**	**36%**	**138%**	**176%**	**251%**	**1446%**	**5156%**	**76%**	**44%**
Ranking	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**

Questions, Feedback?

- We have more problems to be solved.
- Both Los Altos and Toronto

Thank You!