Performance Improvements at Full-Chip level using Elastic on Intel XEON designs

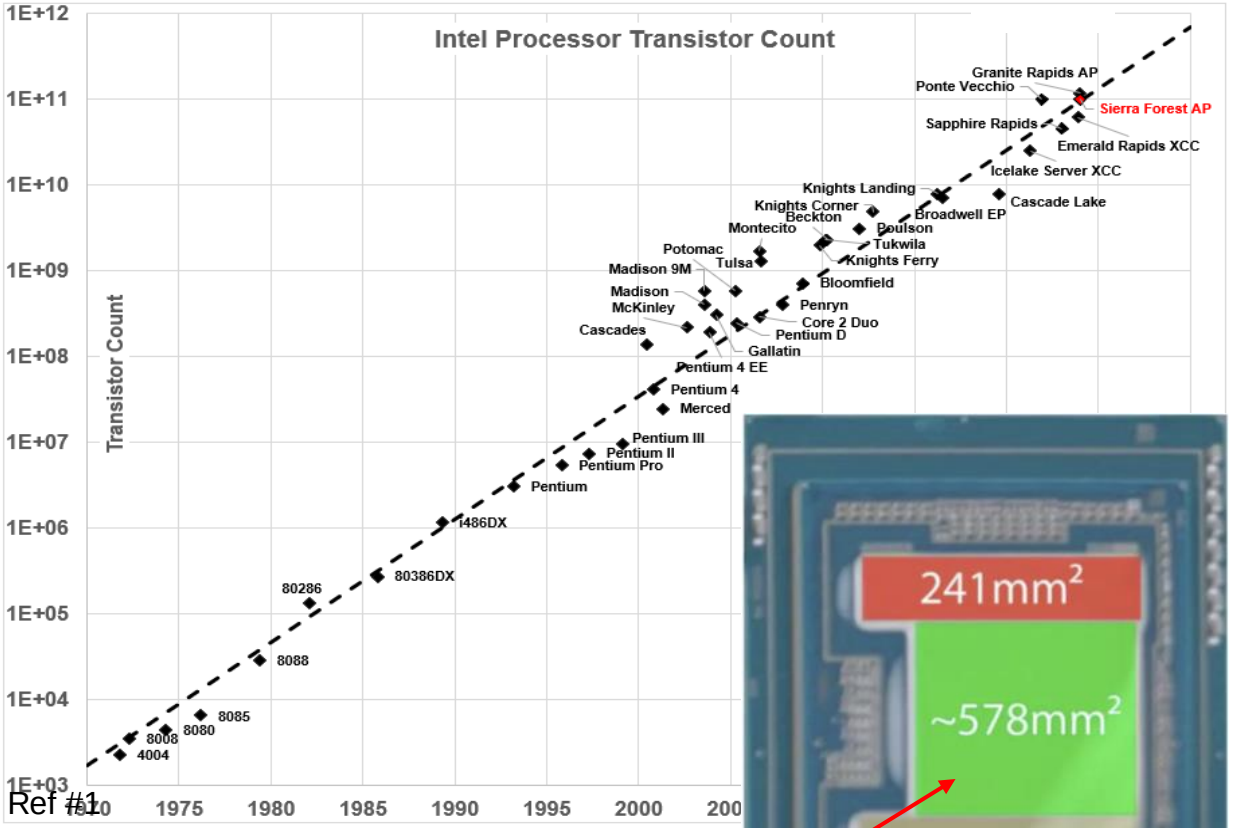
Matt Nichelson, SOC Design Engineer
Intel

Agenda

- Problem statement – “keeping verification up with Moore’s Law”
- Overview of ICV compute options
- A new way of managing FC DRC jobs at Intel
- Sierraforest DRC core/memory usage comparison
- Impact to compute costs using Elastic
- Future Enhancements
- Summary
- Q&A

Problem Statement - "keeping up with Moore's Law"

- Increasing CPU complexity puts high demand on FC DRC
- Transistor counts are increasing on Si and designs are more complex – *need innovation to reduce FC DRC runtimes and compute costs*
- A dedicated hardware pool for FC DRC/Tapein is expensive – *need to seamlessly incorporate high-capacity jobs into compute farms/cloud.*
- Sierraforest XEON product on Intel3 technology needed a way to simplify DRC signoff complexity with new internal compute cost structure.



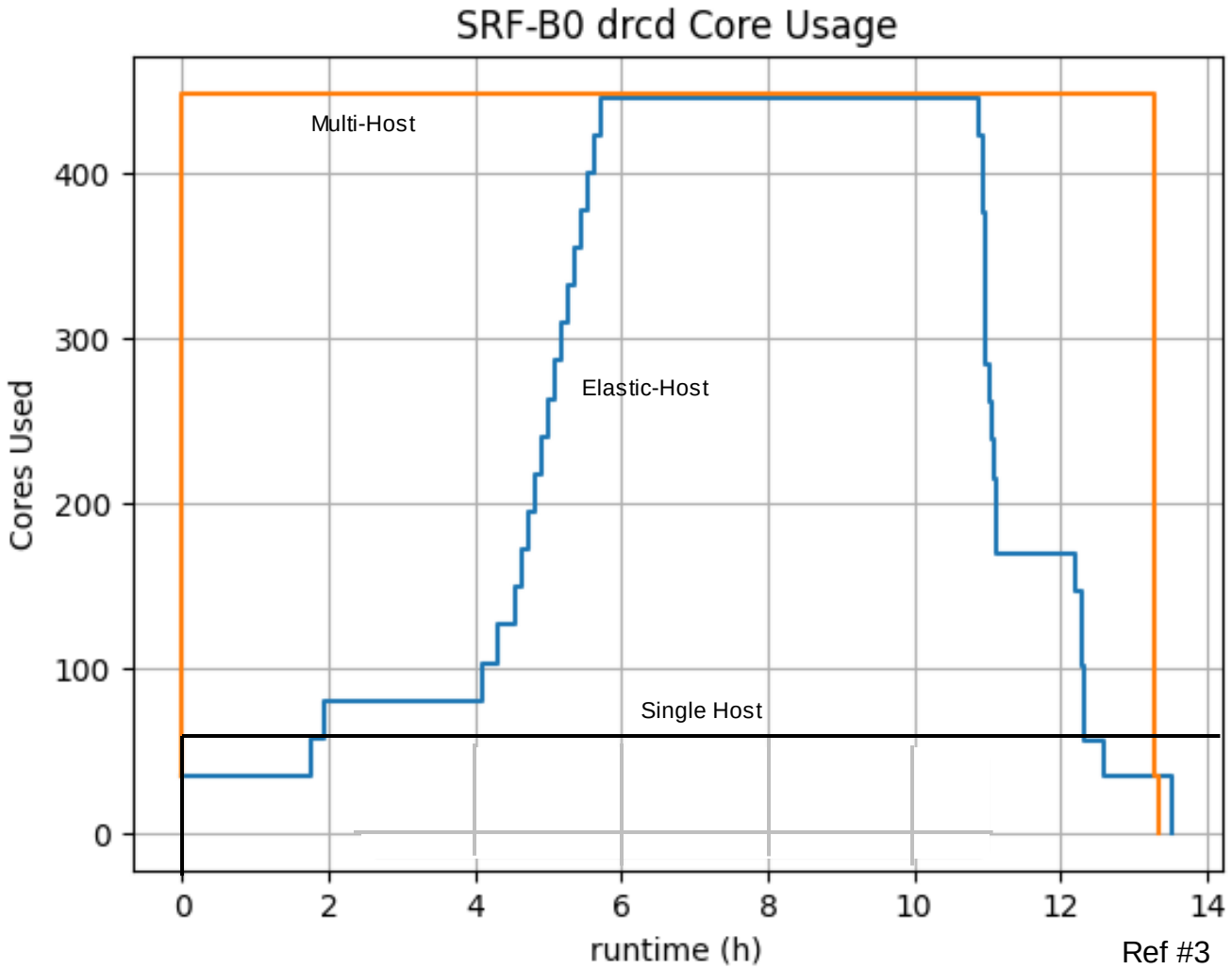
Ref #1

Sierraforest XEON Si
44B transistors, 578mm²

Ref #2

Overview of ICV compute options

- Single host
  - Small load on compute farm
  - Very long runtimes (unrealistic)
  - Starts quickly
- Multi-host
  - Large load on compute farm
  - Fast runtime
  - Long delays in starting
- Elastic CPU
  - Optimizes resources (saves $$)
  - Good runtime
  - Starts quickly
  - Dynamically add/removes hosts



Ref #3

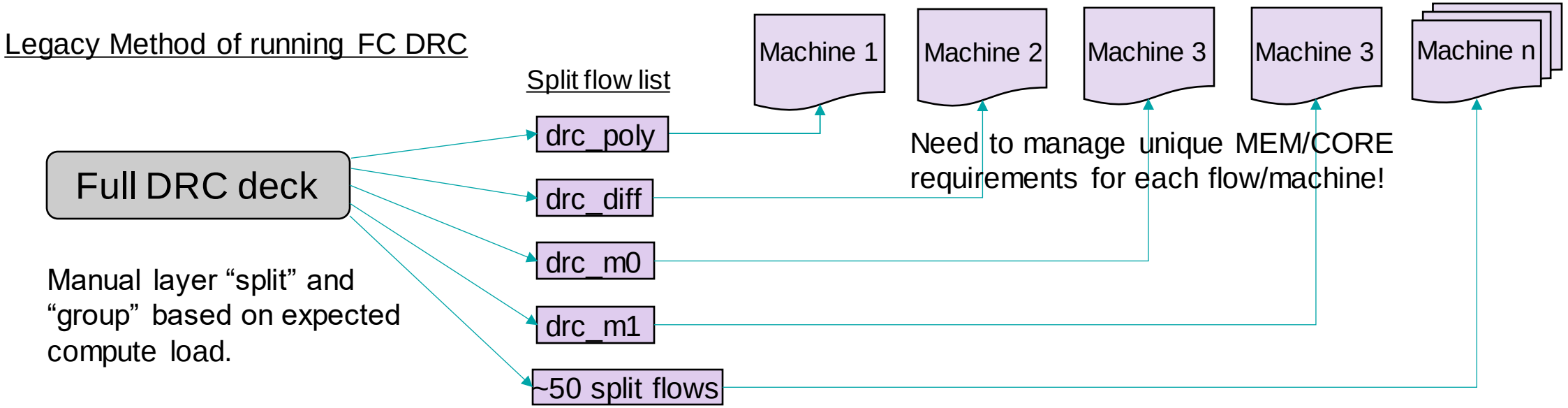
A new way of managing Full-Chip DRC jobs

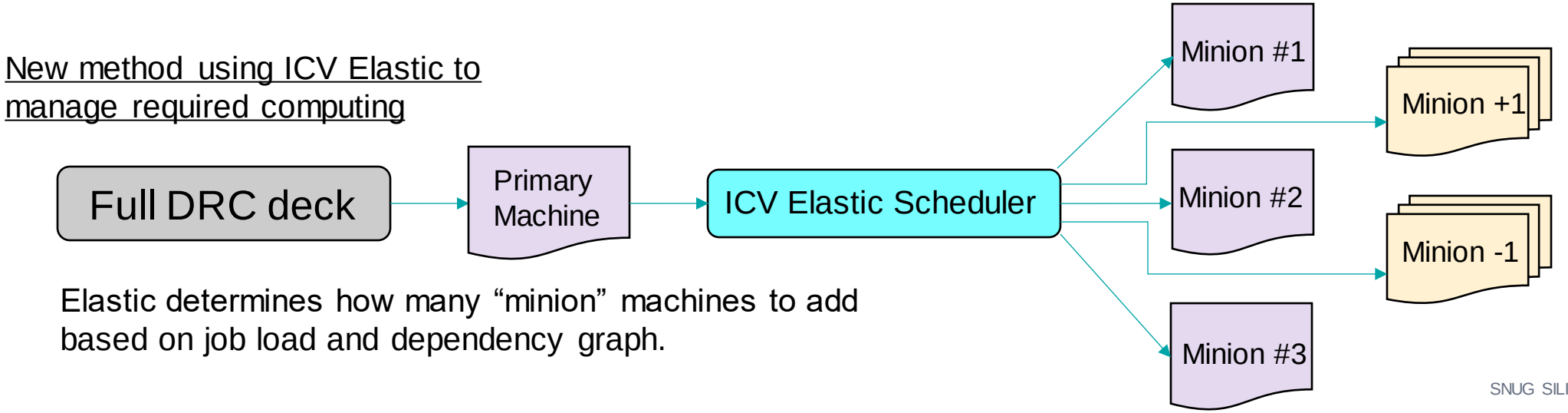
- Historically Full-Chip DRC was unable to flat DRC deck due to extremely long runtimes (multiple days).
  - Solution was to split the DRC deck in multiple flows based on individual layers (~50 flows).
- Each layer/flow was executed on a single machine
  - Unique requirements for memory/core count per flow (high overhead and wasted resources).

Main Bundle	Split Flow	Host Cores	Host Mem (GB)	Avg Mem (GB)	Peak Mem (GB)	Avg/Host Mem Usage	Peak/Host Mem Usage	Runtime
drcd	drc_NW	48	1,583,625	155,273	1,044,111	9.8%	65.9%	18h:51m:56s
drcd	drc_DF	24	790,908	77,286	528,821	9.8%	66.9%	11h:16m:12s
drcd	drc_PG	48	1,056,160	15,755	511,584	1.5%	48.4%	2h:59m:04s
drcd	drc_PL	48	1,583,625	83,507	699,801	5.3%	44.2%	14h:14m:30s
drcd	drc_M1	16	790,911	45,425	525,506	5.7%	66.4%	21h:05m:45s
drcd	drc_M2	16	790,911	49,383	283,887	6.2%	35.9%	10h:32m:47s
drcd	drc_M3	48	2,113,123	59,048	526,190	2.8%	24.9%	11h:46m:38s
drcd	drc_M4	16	790,911	26,021	529,851	3.3%	67.0%	6h:08m:56s
drcd	drc_M5	24	1,056,170	33,419	522,032	3.2%	49.4%	5h:08m:15s

- With the introduction of ICV Elastic, Intel transitioned from *split flows* to running *flat drc* and saw significant improvements by allowing the ICV engine to dynamically distribute the full DRC deck across multiple machines. Elastic uses the ICV Validator NXT feature tool license.

A new way of managing Full-Chip DRC jobs





“Elastic” computing adds and removes hosts

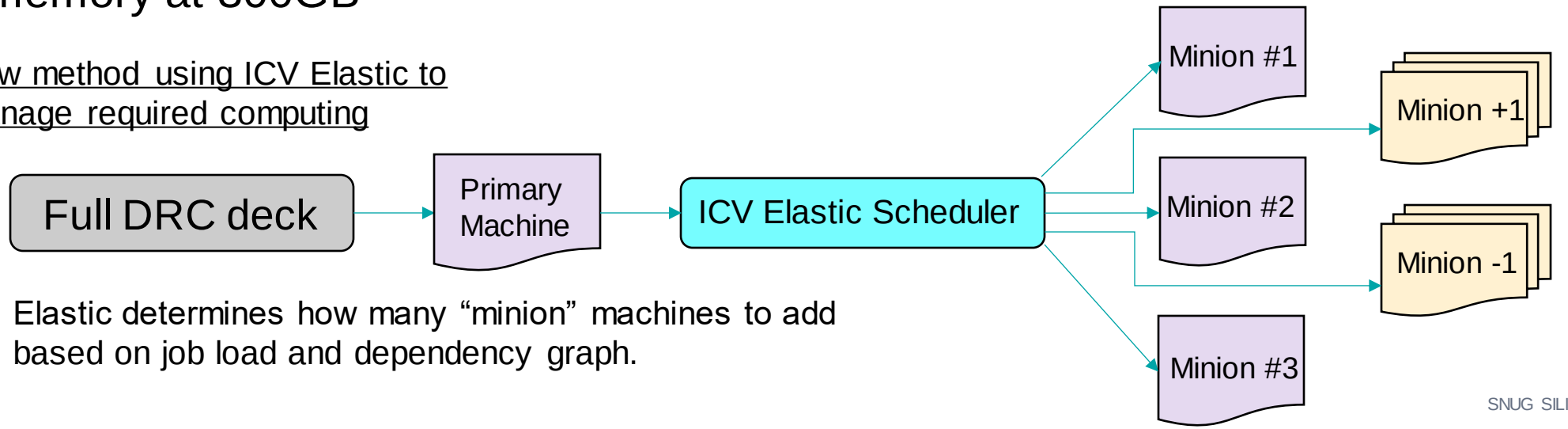
1. Verification jobs starts on “primary” machine
   – On Sierraforest this machine was typically 1.5TB or 2TB for DRC/Antenna
2. ICV internal engine determines when to add/remove additional “minions”
   – DRC data shows average memory on minion hosts to be under 100GB, with average peak memory at 300GB

Primary Machine

IC Validator Machine Memory Report

```
Average = 282.381 GB, Peak = 1827.267 GB
Average = 77.246 GB, Peak = 645.058 GB
Average = 147.957 GB, Peak = 478.579 GB
Average = 53.625 GB, Peak = 324.603 GB
Average = 56.651 GB, Peak = 289.514 GB
Average = 49.986 GB, Peak = 307.020 GB
Average = 55.684 GB, Peak = 243.382 GB
Average = 39.450 GB, Peak = 215.606 GB
Average = 49.449 GB, Peak = 280.682 GB
Average = 61.323 GB, Peak = 344.481 GB
```

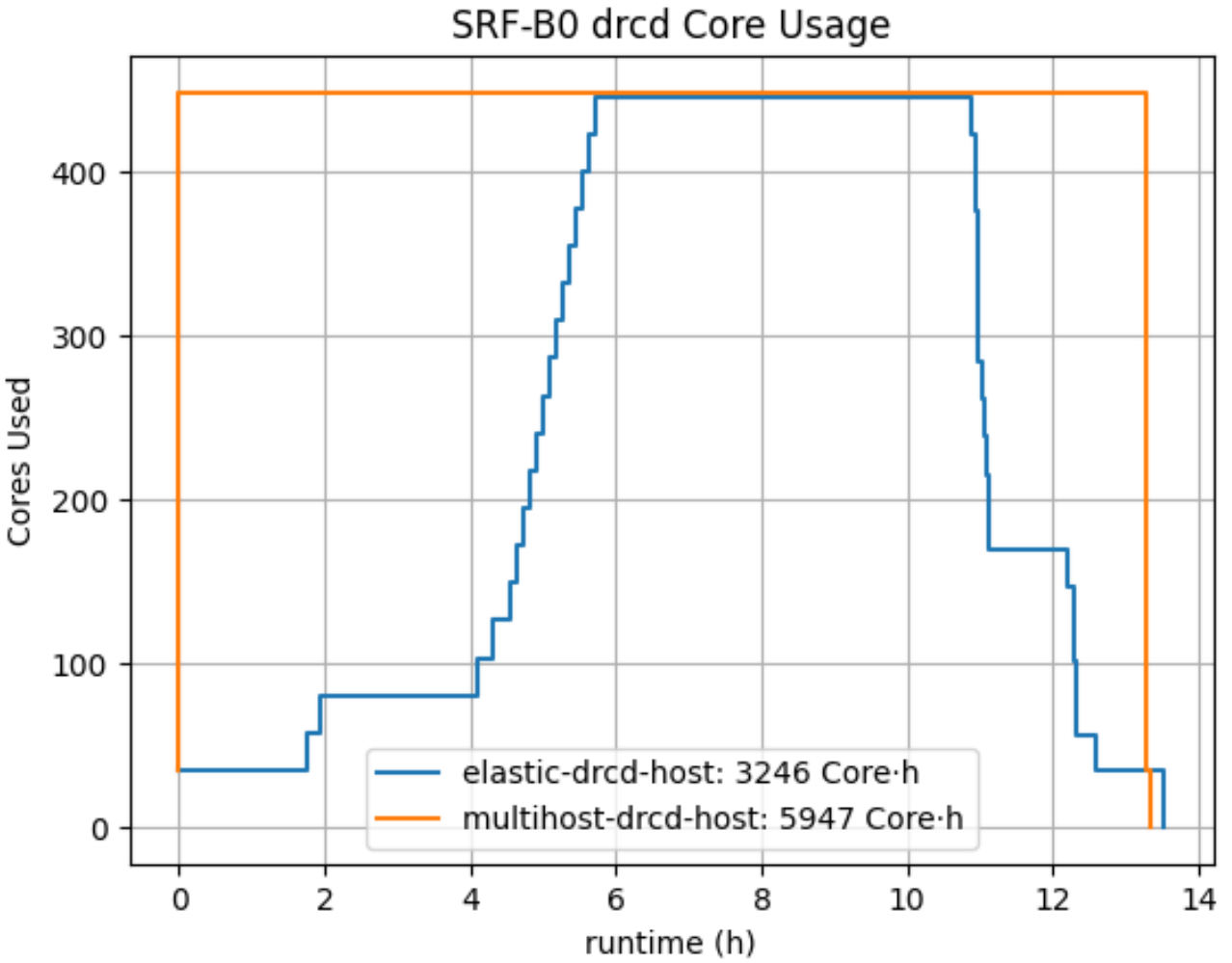
New method using ICV Elastic to manage required computing



Elastic determines how many “minion” machines to add based on job load and dependency graph.

Sierraforest DRC and core usage comparison

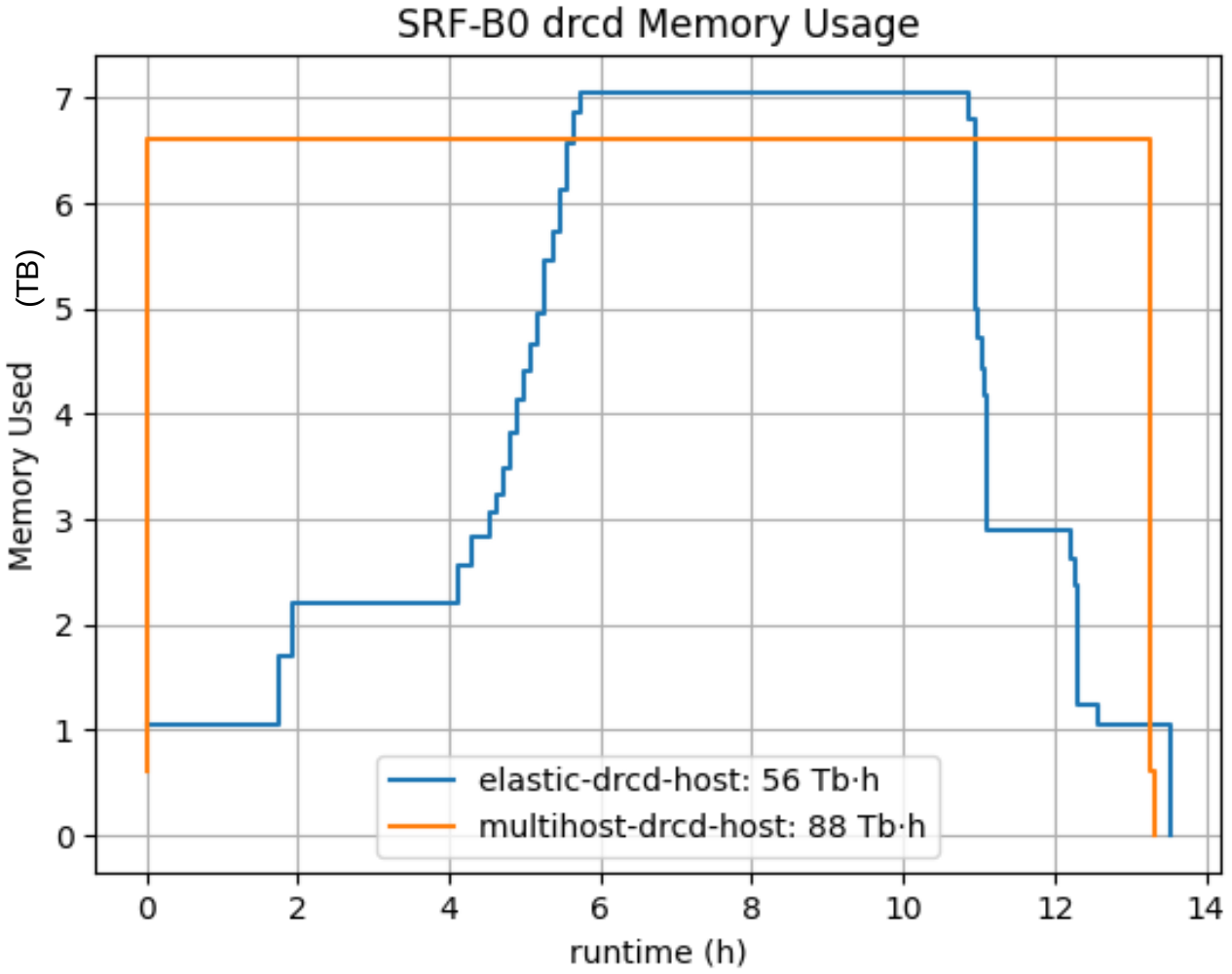
- Multi-host graph shows a fixed cost of **440** cores regardless of process load.
- Elastic ramps up to **440** cores as load increases and then releases cores as job starts to finish.
- Multihost consumed **5,947** core*hour
- Elastic consumed **3,246** core*hour
- **45%** core cost savings with elastic



Ref #4

Sierraforest DRC and memory usage comparison

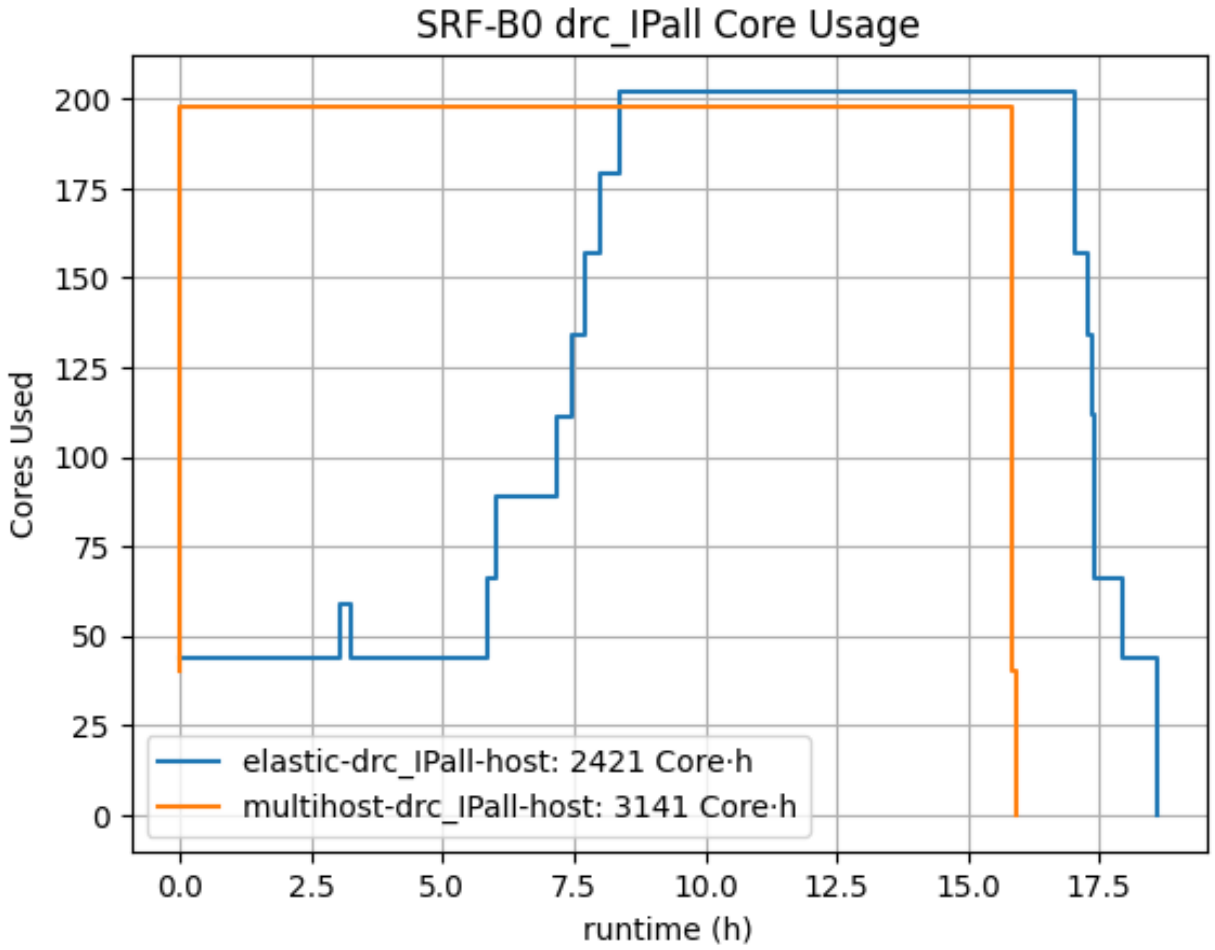
- Multi-host graph shows a fixed memory cost of **6.6TB** throughout the entire run.
- Elastic ramps up to **7TB** memory as load increases and then releases memory as job starts to finish.
- Multihost consumed **88TB** memory*hour
- Elastic consumed **56TB** memory*hour
- **36%** memory savings with elastic



Ref #5

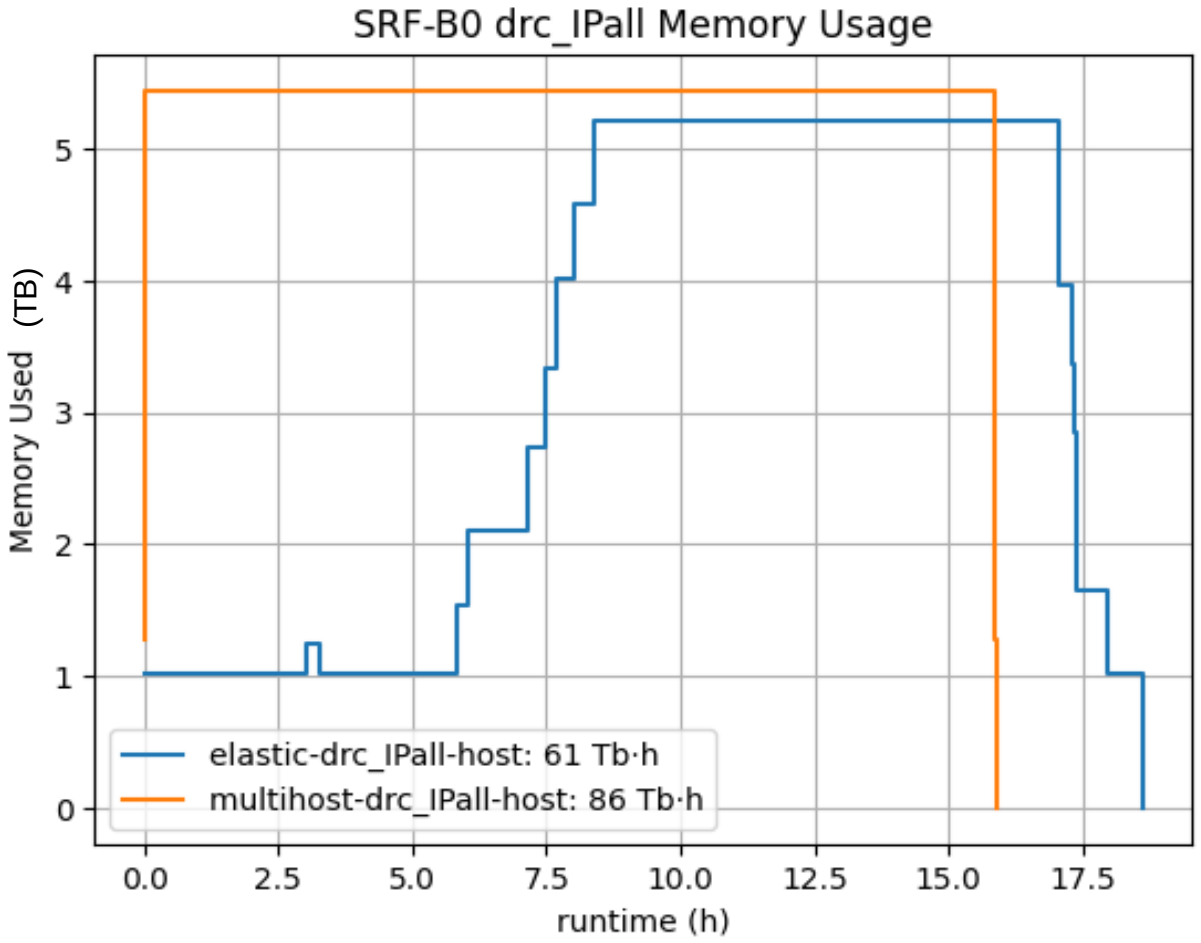
Sierraforest Antenna and core usage

- Multihost graph shows a fixed cost of **196** cores regardless of process load.
- Elastic ramps up to **210** cores as load increases and then releases cores as job starts to finish.
- Multihost consumed **3,141** cores over time.
- Elastic consumed **2,421** cores over time.
- **23%** core cost savings with elastic
- Multihost runtime was **2.5** hours faster than elastic.



Sierraforest Antenna and memory usage

- Multihost graph shows a fixed memory cost of **5.5TB** throughout the entire run.
- Elastic ramps up to **5.2TB** memory as load increases and then releases memory as job starts to finish.
- Multihost consumed **86TB** total over time.
- Elastic consumed **61TB** total over time.
- **29%** memory savings with elastic.



Disk Space consumed by DRC Elastic run

```
Network Disk Usage Peak=1.538 GB (no group)
Group File Disk Usage Peak=345.472 GB
scee414310.sc.intel.com : Local Disk Usage Peak=396.856 GB
scci223104.sc.intel.com : Local Disk Usage Peak=153.450 GB
sccl013704.sc.intel.com : Local Disk Usage Peak=142.533 GB
scci223103.sc.intel.com : Local Disk Usage Peak=140.162 GB
sccl023705.sc.intel.com : Local Disk Usage Peak=177.257 GB
sccl023708.sc.intel.com : Local Disk Usage Peak=137.231 GB
sccl063704.sc.intel.com : Local Disk Usage Peak=132.464 GB
scee062502.sc.intel.com : Local Disk Usage Peak=126.237 GB
scdg083110.sc.intel.com : Local Disk Usage Peak=120.735 GB
scee062503.sc.intel.com : Local Disk Usage Peak=121.631 GB
sccl063708.sc.intel.com : Local Disk Usage Peak=125.176 GB
scee012505.sc.intel.com : Local Disk Usage Peak=139.750 GB
sccl073702.sc.intel.com : Local Disk Usage Peak=121.885 GB
scdh074303.sc.intel.com : Local Disk Usage Peak=142.228 GB
scdh074307.sc.intel.com : Local Disk Usage Peak=112.643 GB
scee012501.sc.intel.com : Local Disk Usage Peak=129.664 GB
scdg173708.sc.intel.com : Local Disk Usage Peak=126.296 GB
scdg173709.sc.intel.com : Local Disk Usage Peak=138.528 GB
scci223108.sc.intel.com : Local Disk Usage Peak=139.251 GB
```

mnichels@scc920013 : du -ksh drc/
1.4G drc/

Impact to compute costs using Elastic

- Using ICV Elastic has direct impact to project cost
  - Higher memory usage and higher core count == higher system requirements and >> cost per job
  - Savings from Elastic come from optimizing the required resources “on the fly”

Flow	Mode	Cost
DRC	Multi-host	$50.00
DRC	Elastic	$30.85
	Savings	38.30%
Antenna	Multi-host	$35.00
Antenna	Elastic	$28.07
	Savings	19.80%

Normalized data

Machine Performance

Fastest per-CPU performance
- Single Socket
- Fewer cores / scaling penalty

Slower per-CPU performance
- More cores / scaling penalty

	Platform Performance		
Memory	Fast	Faster	Fastest
8GB	1	1.3x	1.6x
16GB	1.3x	1.7x	2.2x
32GB	1.7x	2.2x	2.7x
64GB	2.7x	3.5x	4.3x
128GB	6.7x	8.7x	10.8x
256GB	8.3x	10.9x	13.5x
512GB	16x	21x	26x
1TB	32x	41.8x	51.7x
1.5TB	100x		
Up to 6TB	200x		

Machine Availability

Higher Quantities
- Bias: Smaller / quicker jobs

Limited Quantities
- Specialized workloads & critical path compute

Intel Confidential Provided to Synopsys Under NDA

Ref #6

Future Enhancements

- Enable elastic to accept “tiered” minion hosts with varied memory size based on prediction of elastic job scheduler.
  - Compute “heavy” elastic threads (predicted) request larger minion memory class
  - Compute “light” elastic threads (predicted) request smaller minion memory class
  - This distribution of minion sizes will push some threads to smaller/cheaper/faster hardware.

- Using historical runtime info to help guide correct minion hardware future job submissions.
  - Similar to request above but with a more solid prediction engine that records n-1, n-2 elastic jobs for better hardware forecast.

- Reduce host memory footprint for minion jobs to fall into less expensive and more available compute hardware in the Cloud.

Summary

- Increased CPU complexity + transistor counts on Intel Sierraforest project putting high demand on Full-Chip DRC turn around time.
- Intel transitioned to Elastic for large blocks and Full-Chip level vs split flows.
- Elastic enables significant improvements in reducing compute resources for both DRC and Antenna over Multi-host:
  - **DRC** reports **45%** fewer cores*hour and **36%** less memory*hour
  - **Antenna** reports **23%** fewer cores*hour and **29%** less memory*hour
- Cloud cost services are also reduced with Elastic by requiring less hardware for equivalent job throughput:
  - **DRC** reports **38%** cost savings
  - **Antenna** reports **20%** cost savings
- Continued advancements in memory optimization will help push Cloud costs down even further.


THANK YOU
YOUR
INNOVATION
YOUR
COMMUNITY

References

- Reference #1
  - "Intel Processor Transistor Count", Grant McFarland, Intel (PE)
- Reference #2
  - "SierraForest package photo", Tom's hardware
  - https://www.tomshardware.com/news/intel-announces-288-core-processor-5th-gen-xeon-arrives-december-14
- Reference #3, #4, #5
  - Sierraforest CORE/MEM data plots Elastic vs Multi-host, Jon Krause, Intel (PE)
- Reference #6
  - "Batch Compute: Machine Performance & Availability", Rick Ferreri, Intel, (SrPE)