Automated Design of Computer Clusters

Konstantin S. Solnushkin <konstantin@solnushkin.org>

WWW: *http://ClusterDesign.org/saddle*

ClusterDesign.org

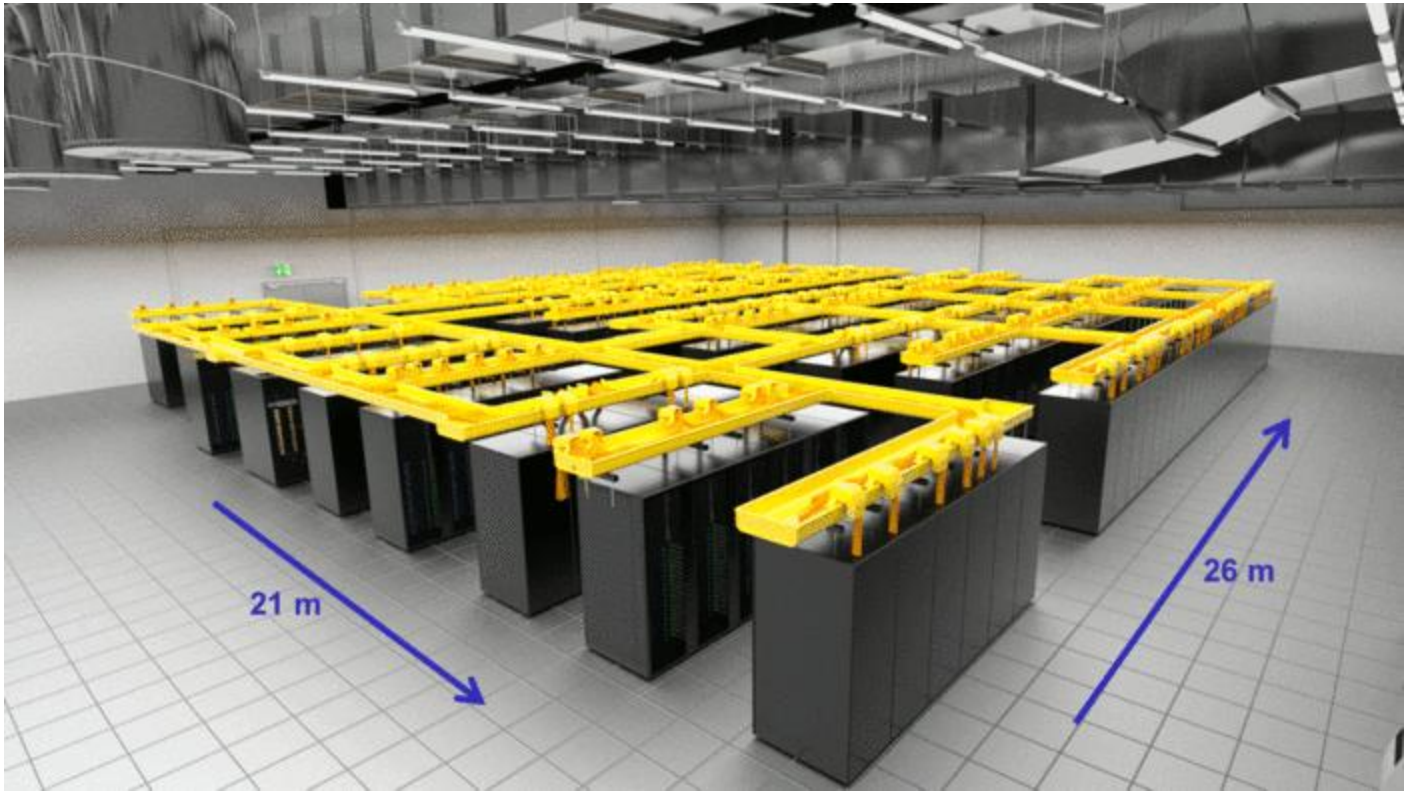
About the author

- B.S. (Hons) and M.S. (Hons) from the Saint Petersburg State Polytechnic University
- Member of the Association for Computing Machinery (ACM)
- Member of the Program Committee of the ISC HPC Conference

Agenda

- Related work
- Introduction and motivation: why automate?
- Design workflow
- Criterion function
- Performance modelling, direct and inverse
- Modularity of the CAD System
- Graph representation of configurations
- SADDLE, *the* CAD tool for cluster and datacentre design
- Economic characteristics
- Scientific contribution

Related Work



Related Work

What was before?

- *R1*, a production rule expert system
  - Created by John P. McDermott in late 1970s
  - Used to to configure VAX-11/780 minicomputers made by Digital Equipment Corporation
  - Operated on a set of 480 rules representing domain knowledge
  - Took various mechanical and power constraints into account
  - Produced detailed assembly documentation including floor plans and cable wiring tables
  - **Set the quality standard for future tools**

- ICOS, an Intelligent Concurrent Object-Oriented Synthesis methodology
  - Created by Pao-Ann Hsiung et al. in 1998
  - Focused on design of multiprocessor systems (but not cluster computers)
  - With object-oriented approach, system components are modelled as classes with hierarchical relationships between them
  - Previously synthesised subsystems can be reused as building blocks of new designs
  - Machine learning and fuzzy logic are used to determine feasibility of the reuse

Related Work

What was before (continued)?

- Vendor tools, e.g.:
  - IBM Standalone Solutions Configuration Tool (SSCT)
  - Hewlett-Packard BladeSystem Power Sizer
  - Don’t try to predict performance

- “Cluster Design Rules” (CDR) by William R. Dieter and Henry G. Dietz, University of Kentucky (ca. 2005)
  - A pioneering effort, but no longer maintained
  - Includes performance models for Linpack and SWEEP3D benchmarks
  - You can’t plug-in your own performance model: the software is not modular

Introduction and Motivation

Cluster Design Tools from ClusterDesign.org

is a replacement for those out-of-date frameworks:

- *Very* modular
- Can design data centres, too
- Developed since 2012
- Source code available

Benefits of design automation (1 of 3)

- Generate only feasible configurations of building blocks (compute nodes)
- Quickly select configurations which are the best according to some metric (e.g., price/performance ratio)
- Automatically assess performance (if performance models are available)
- Accurately estimate metrics for the whole solution: cost, power, weight, space, etc. No more “educated guesses”!

Benefits of design automation (2 of 3)

- Perform a more thorough search of the design space than a human engineer can do (and in less time)

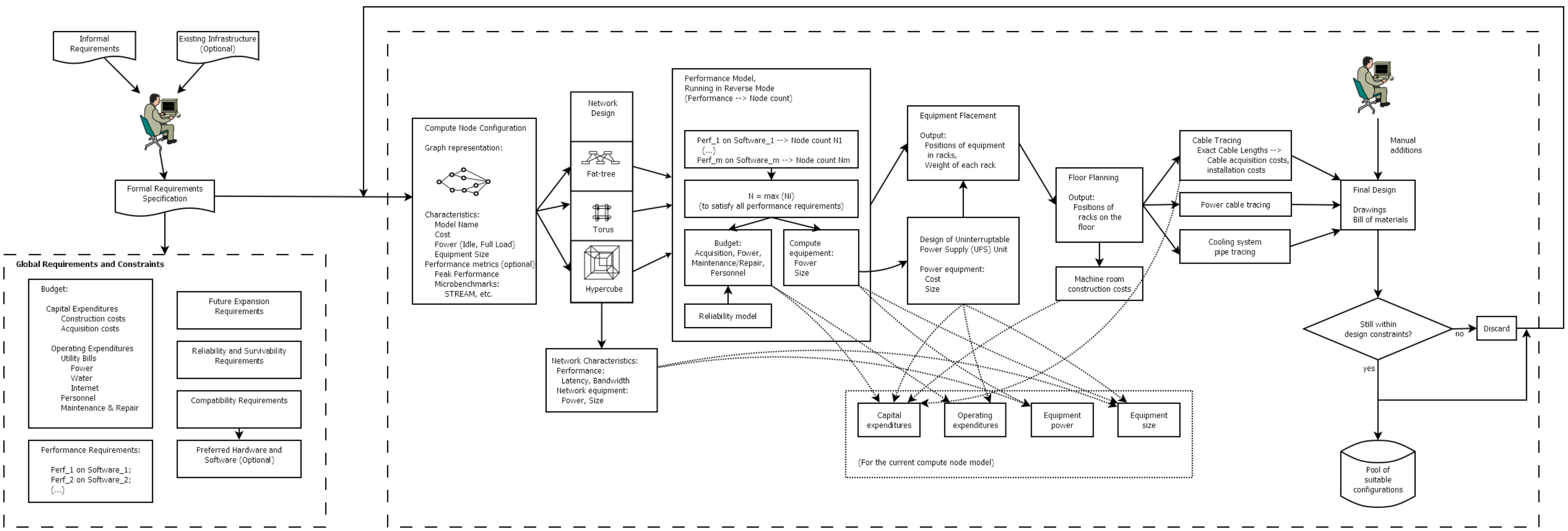
- Produce the set of documents that help streamline procurement and assembly processes
  - Bill of materials – what to buy
  - Technical and economic metrics – cost, power, weight, whatever
  - Cabling diagrams

Benefits of design automation (3 of 3)

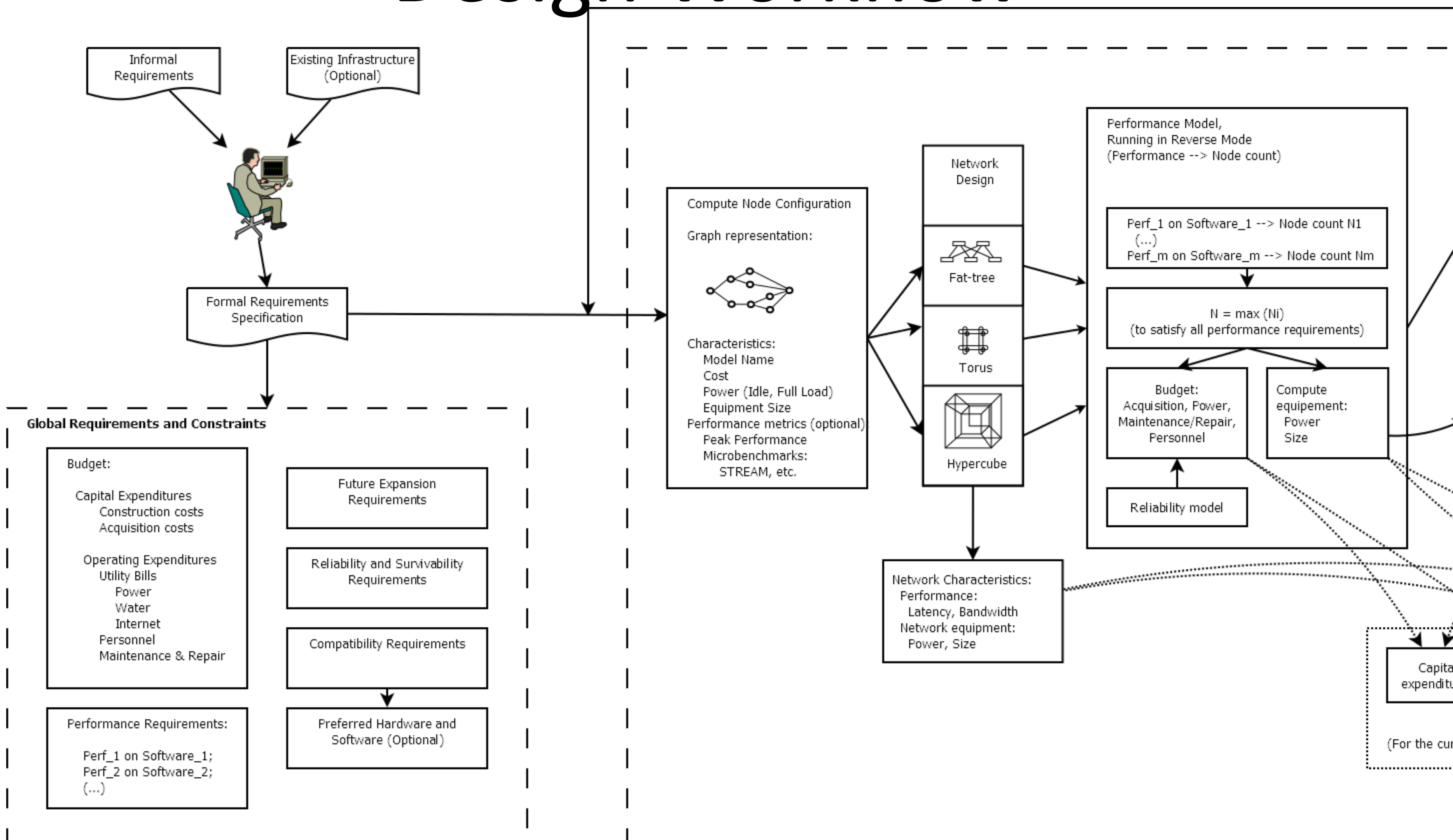
- Design of a whole supercomputer is a complement to low-level EDA
- With EDA, you design things like CPUs or memory modules
- Then, you can "plug" the result of chip-level design into a whole-system design
- EDA subsystems then become modules of a larger CAD system that designs a whole supercomputer

Design Workflow

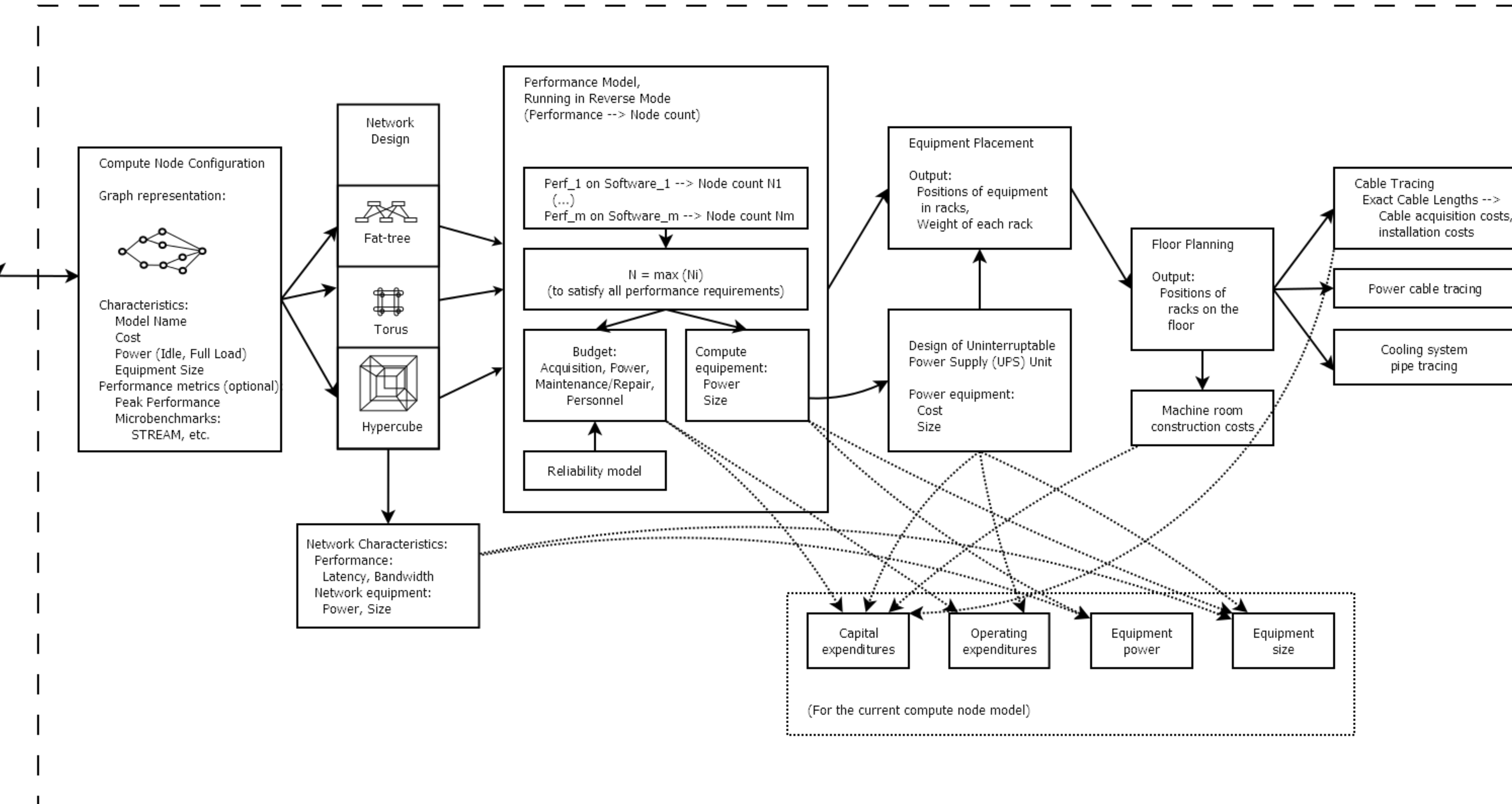
Design Workflow



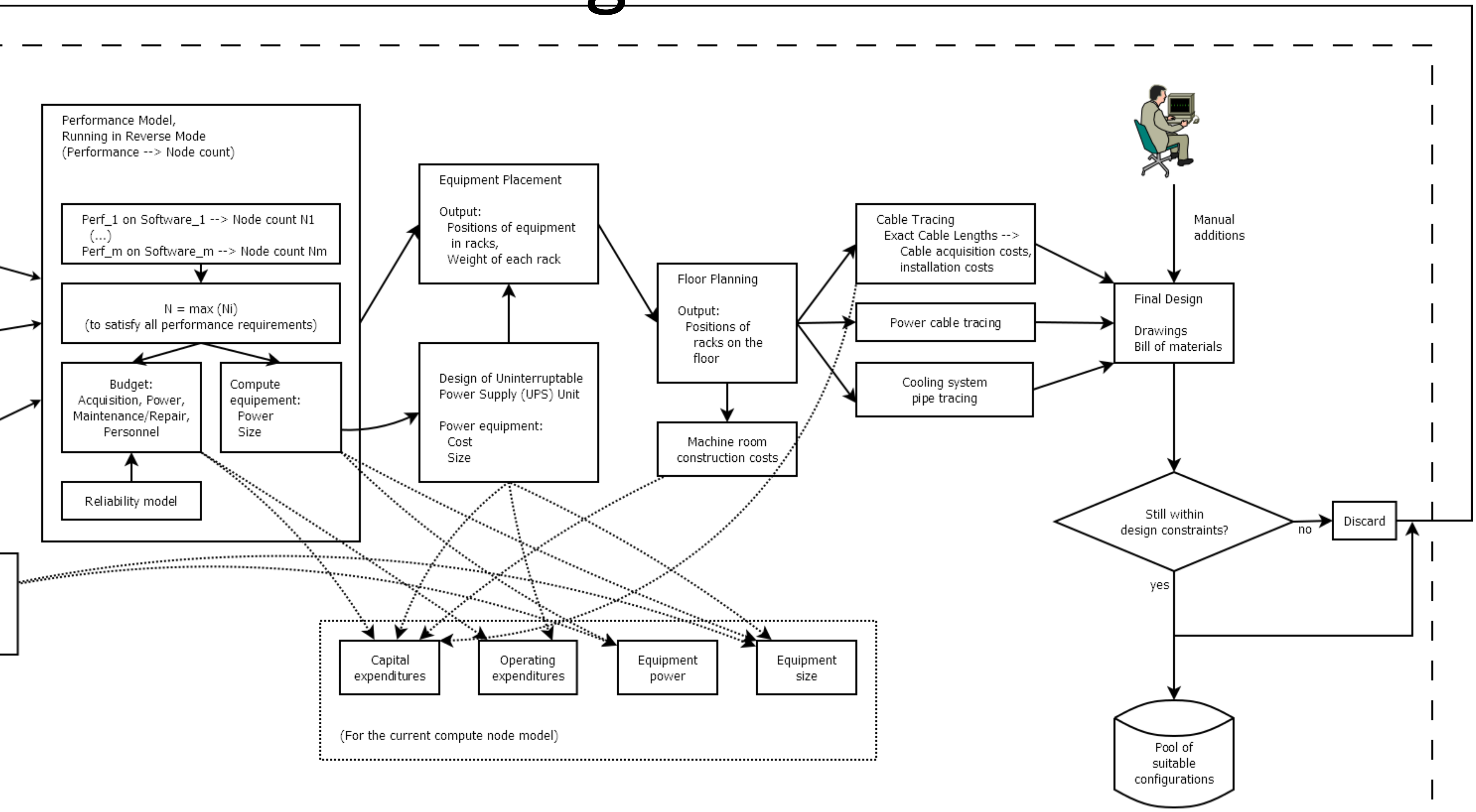
Design Workflow



Design Workflow



Design Workflow



"TCO to Performance" ratio as a criterion function

Criterion functions

- Linear combination of technical and economic characteristics – doesn't work, because weights are assigned arbitrarily

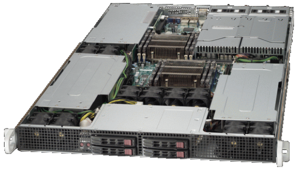
- "Performance per watt" and "Performance per watt per Euro" – trendy but do not work, because they are not robust:
  - slight perturbations in the values of characteristics significantly change the ordering of candidate solutions

Criterion functions

- The objective measure is the total cost of ownership
- The criterion function then is the "TCO / Performance" ratio
- Non-linear, and exhaustive search impossible because of combinatorial explosion
- Requires the application of heuristics and constraints
  - Heuristics alone weed out 90% of unpromising solutions

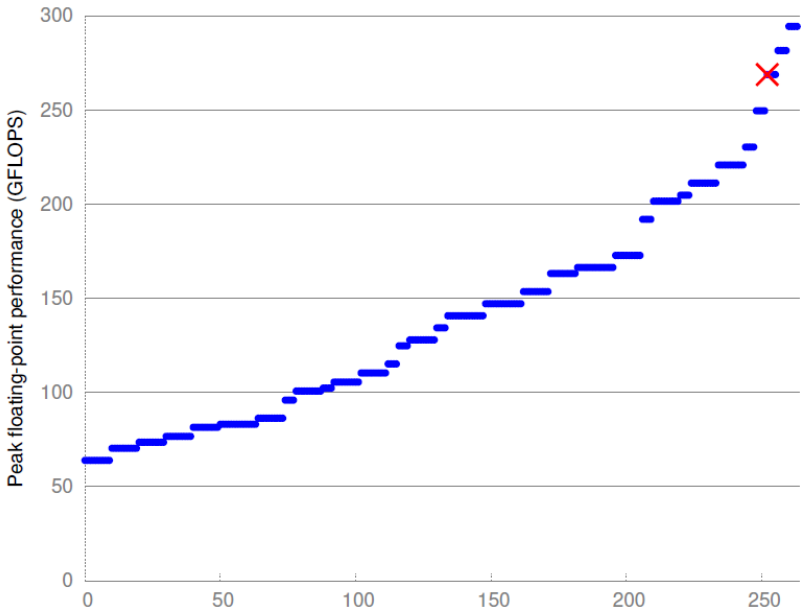
Criterion functions

- Example: take a common compute node that can have 264 valid configurations

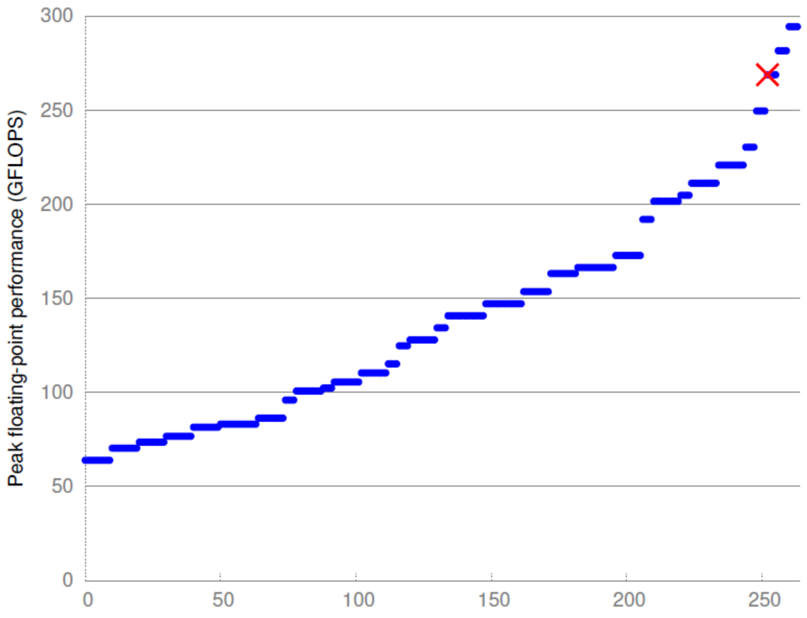


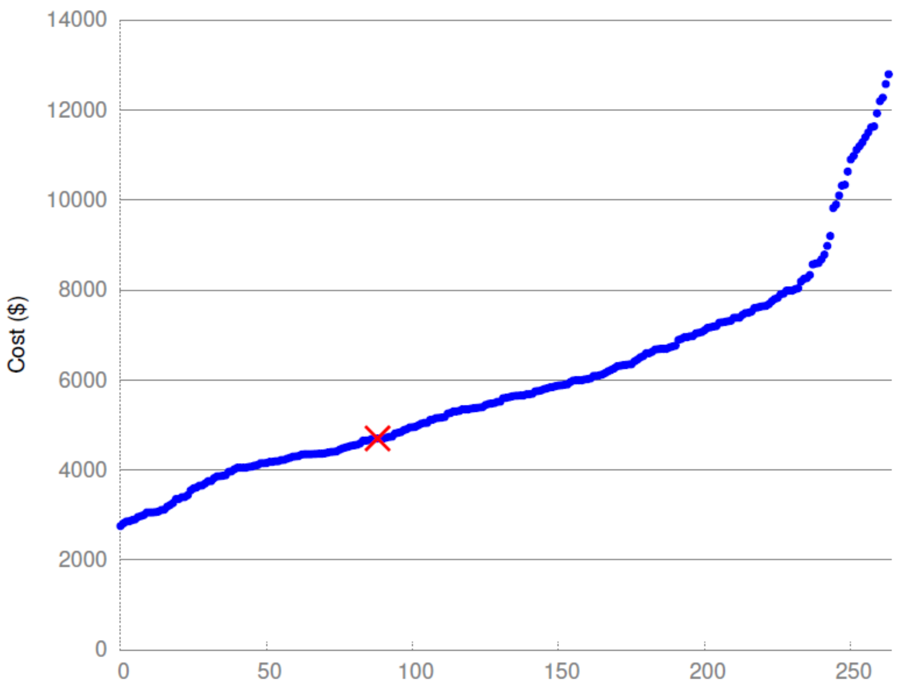
- Put a constraint: select only those configurations that can achieve performance of 240 tasks/day on the ANSYS “truck_111m” benchmark
  – 136 configurations satisfy this constraint
- Display those 136 configurations along the “Cost” and “Performance” axes

Are the 264 configurations really different?

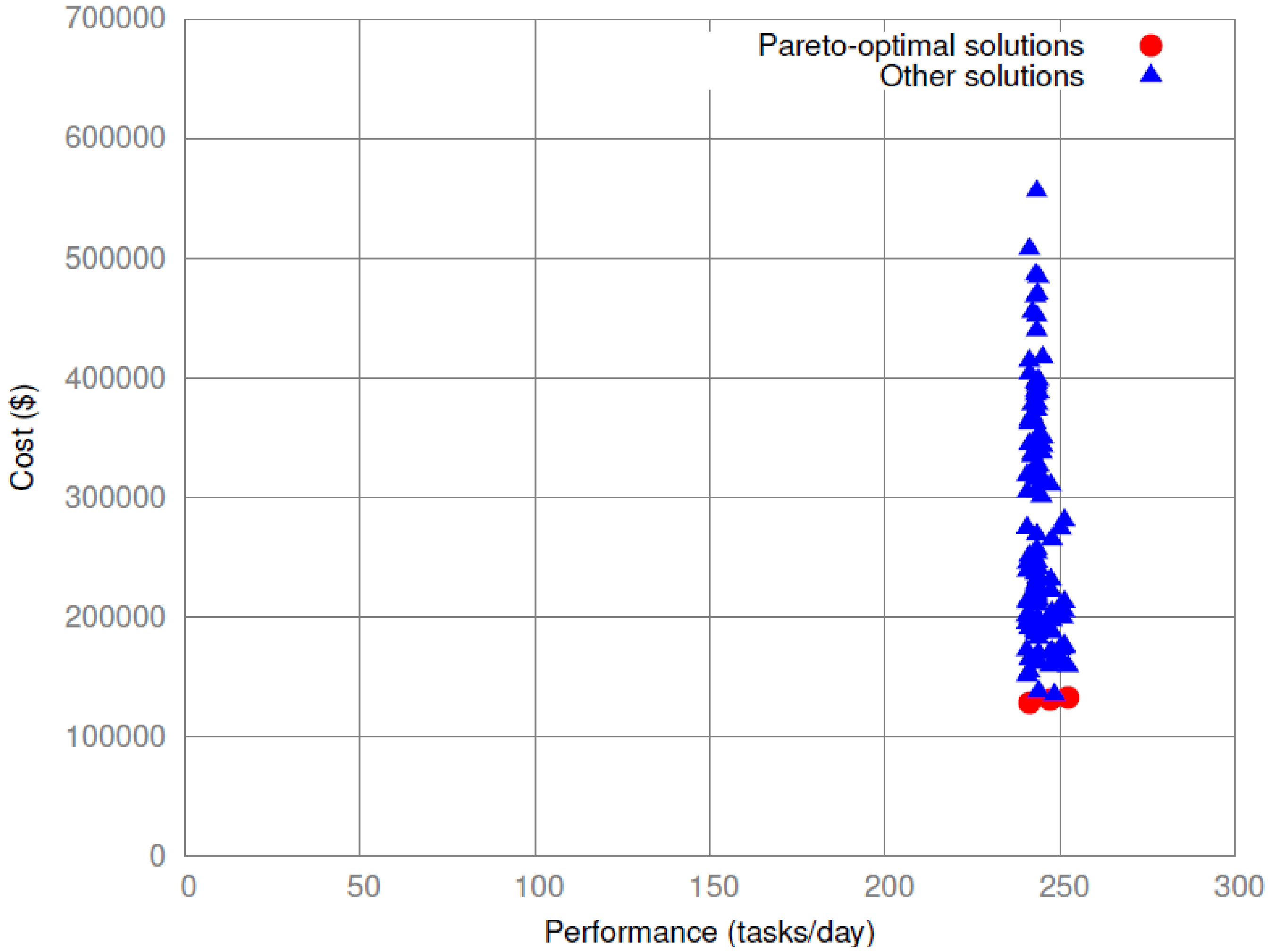


Are the 264 configurations really different?

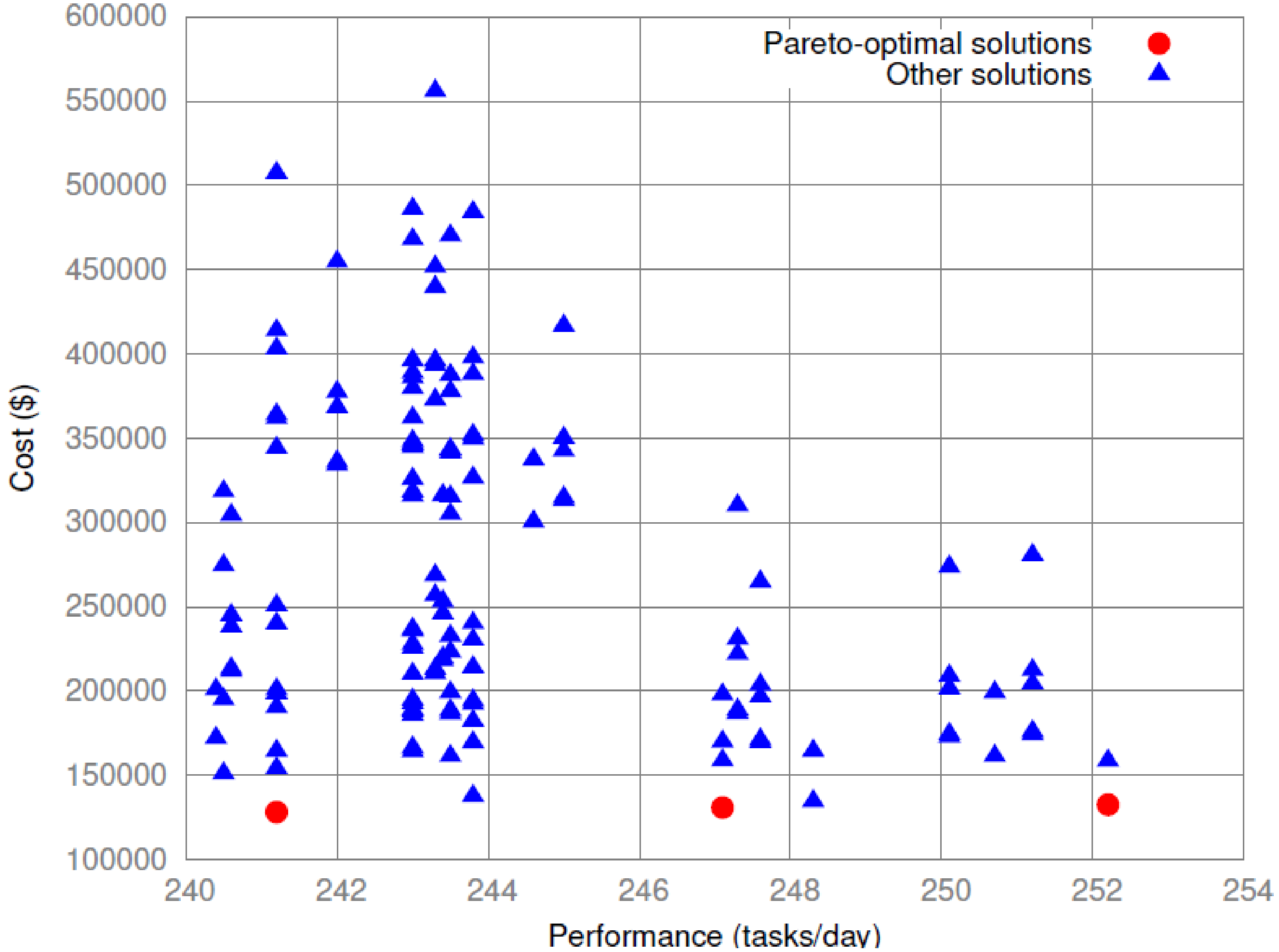




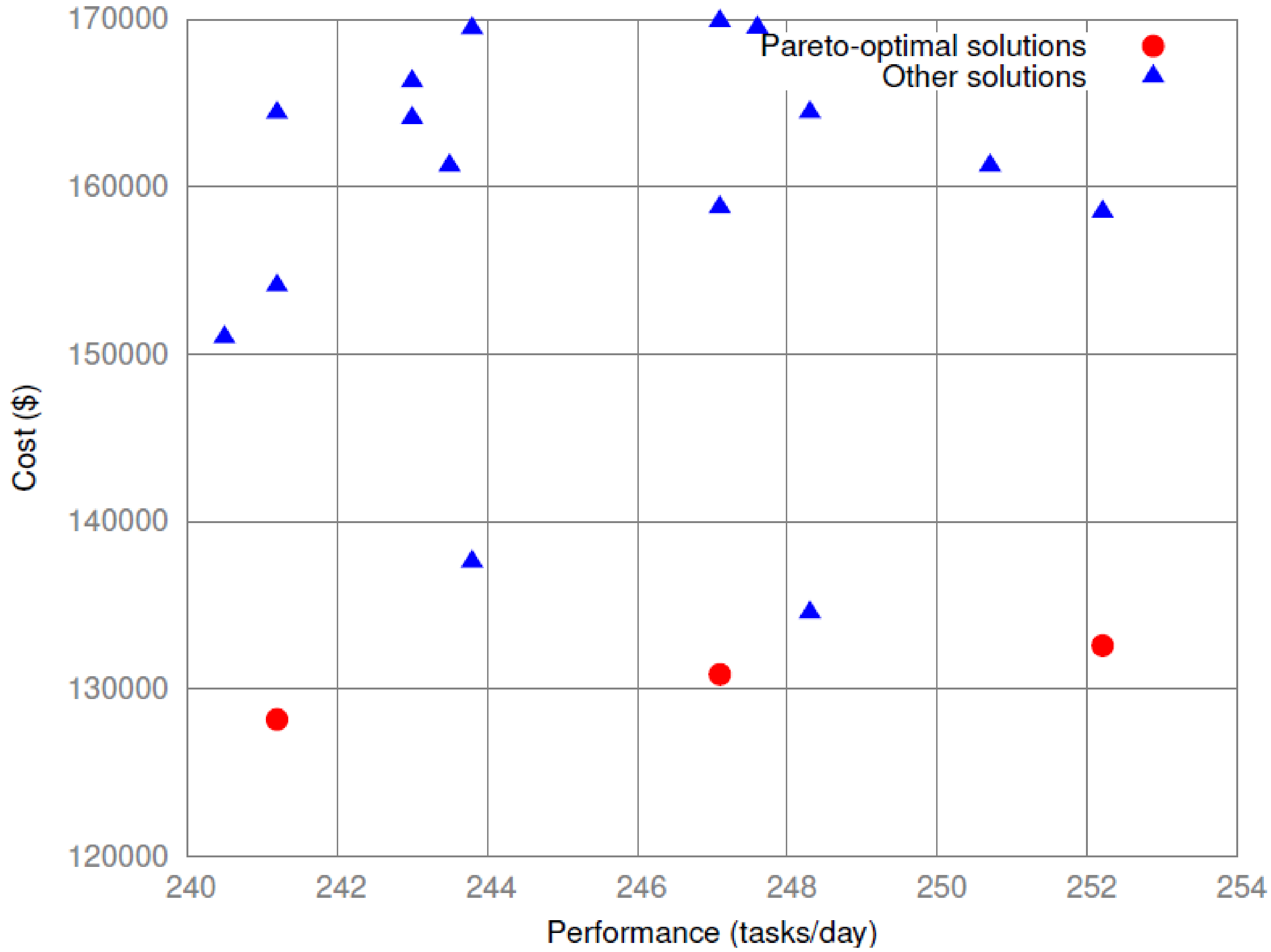
Criterion functions



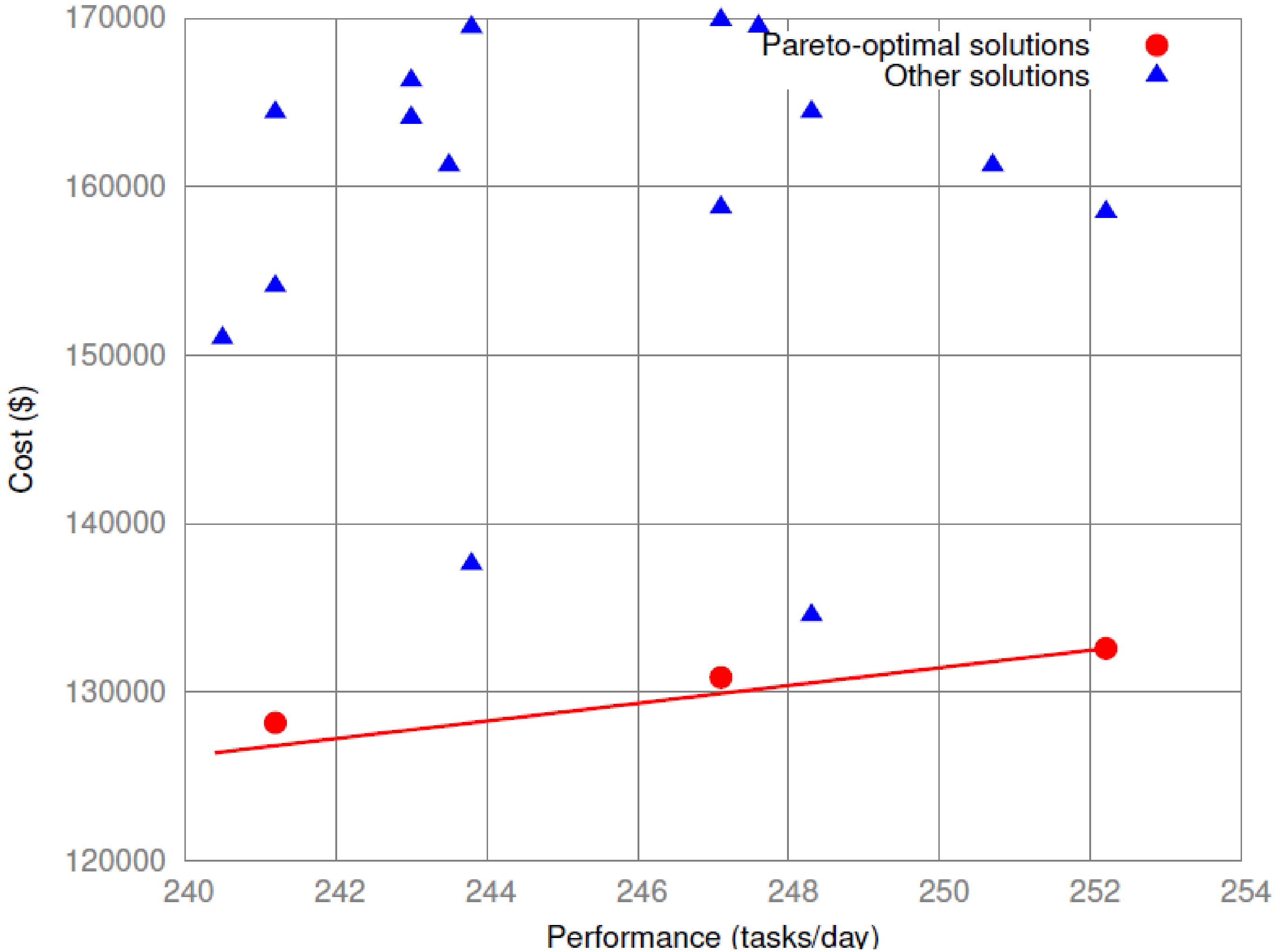
Criterion functions



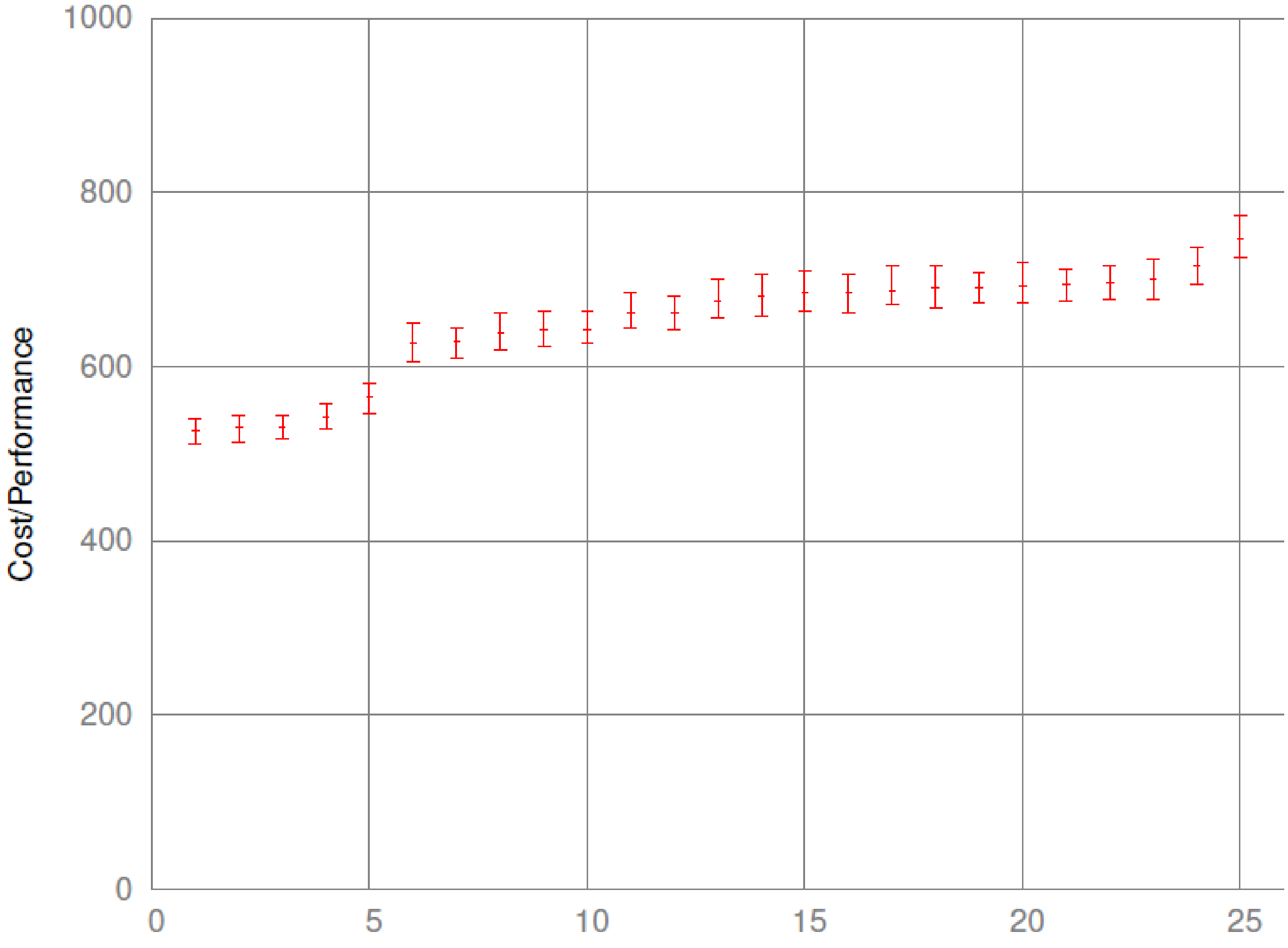
Criterion functions



Criterion functions



Criterion functions



Performance Modelling, Direct and Inverse

Performance Modelling, Direct and Inverse

- Direct performance modelling:
  - Given the number of compute blocks (nodes, cores, etc.)
  - and their parameters (CPU frequency, cache size, etc.),
  - calculate performance on a given task

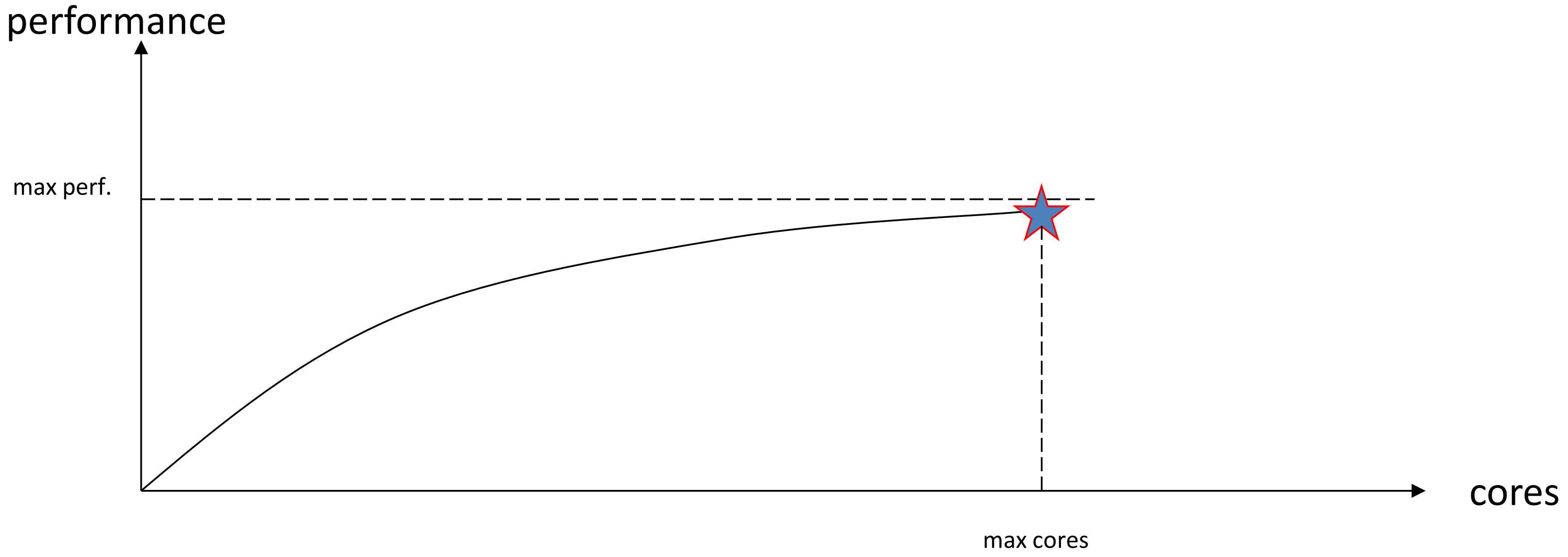
(No. of blocks, Block parameters) → Performance

- Inverse performance modelling
  - Given the performance you need to achieve
  - and parameters of compute blocks,
  - calculate the number of those blocks

(Performance, Block parameters) → No. of blocks

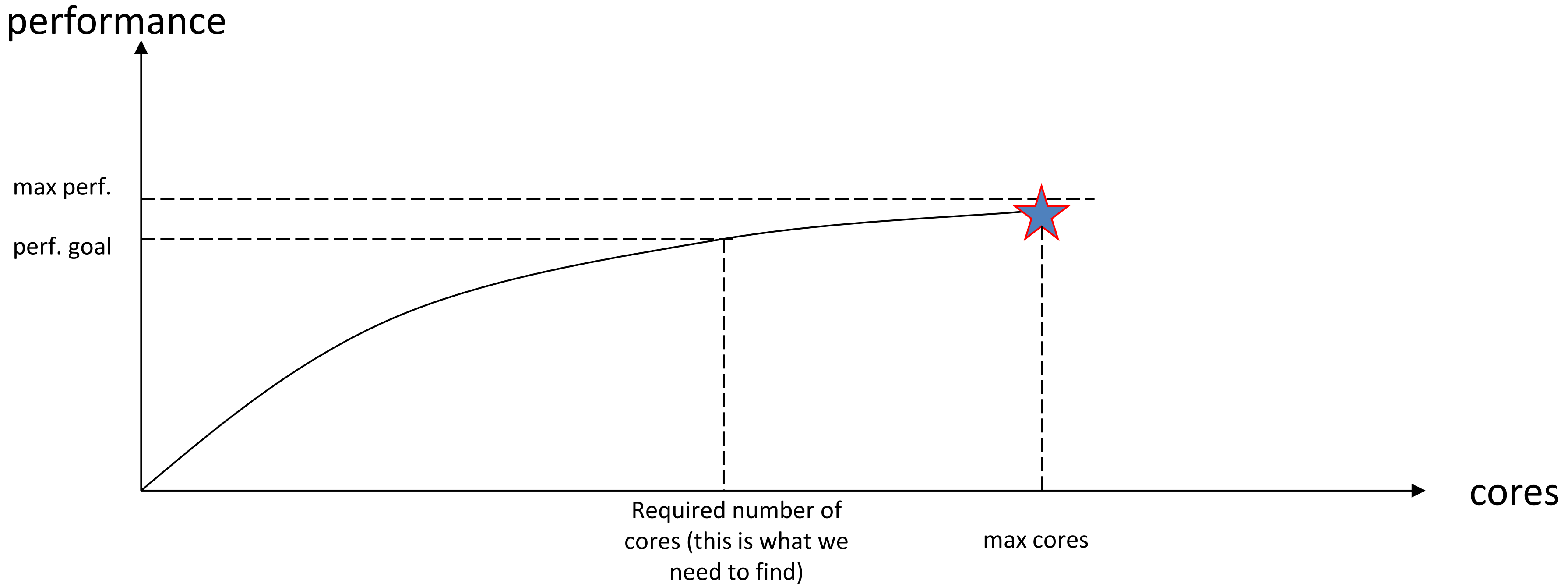
Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



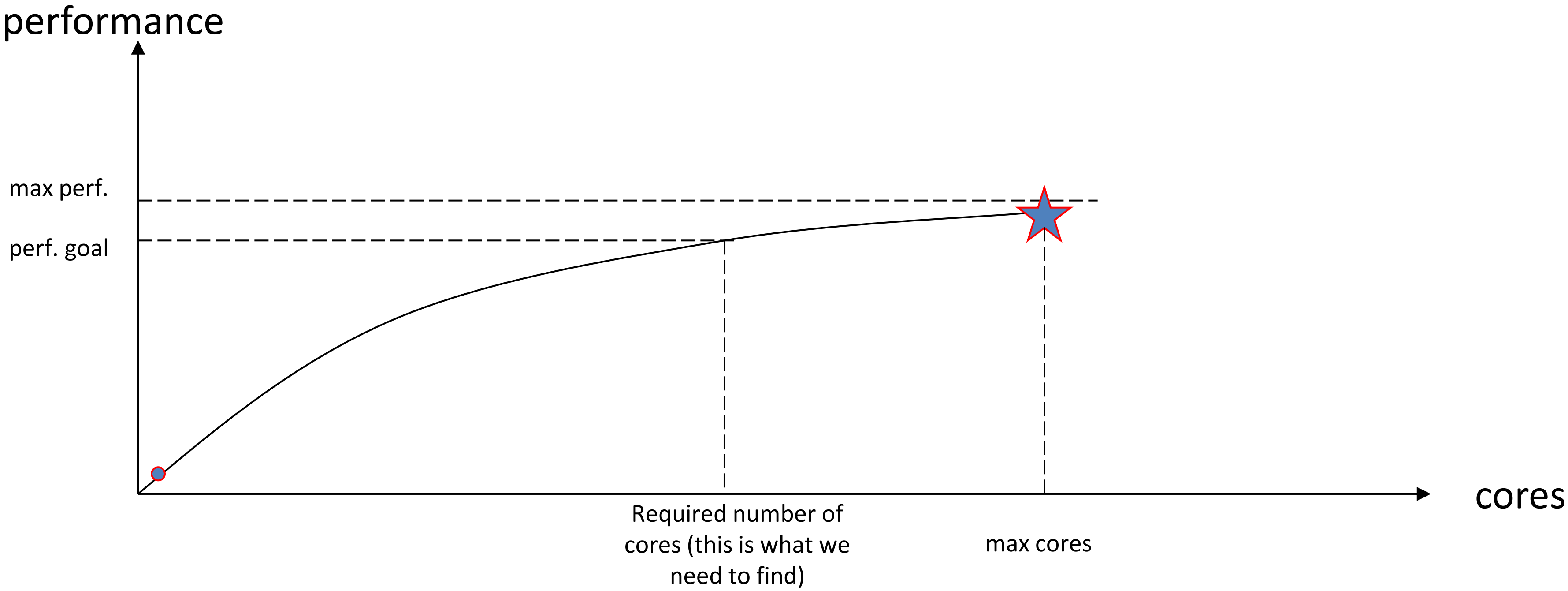
Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



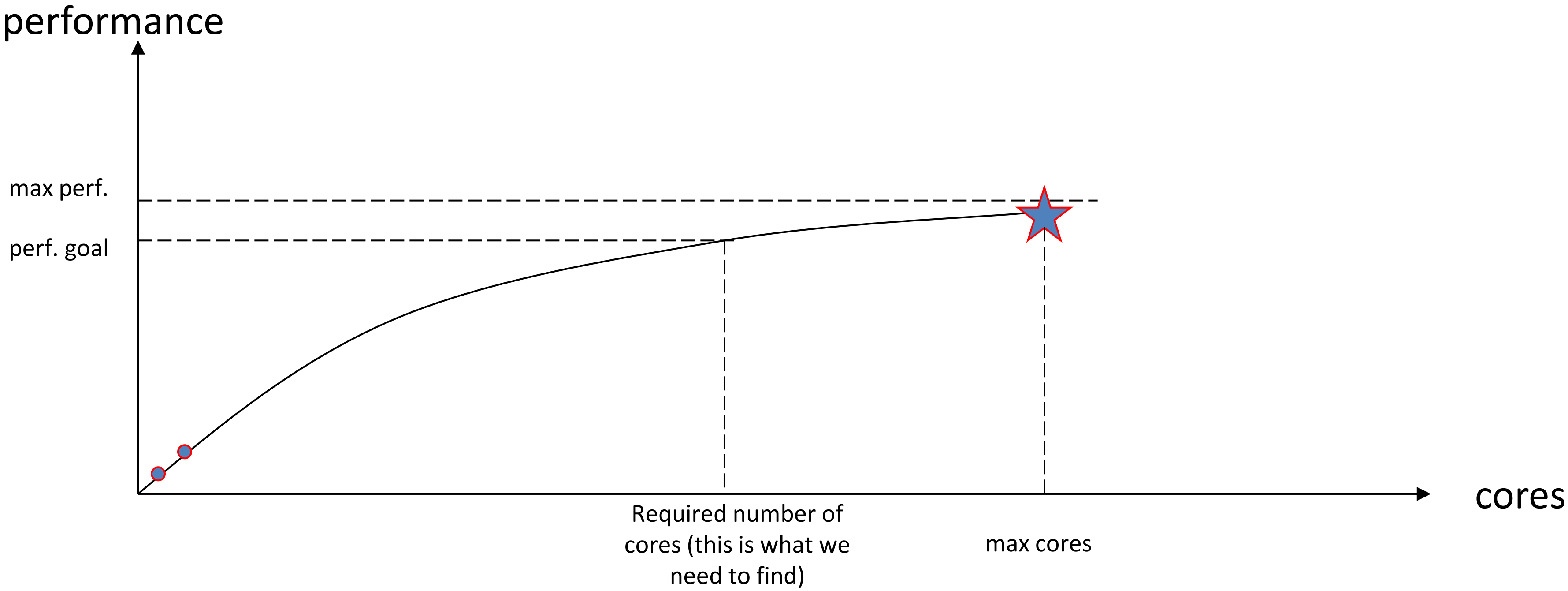
Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



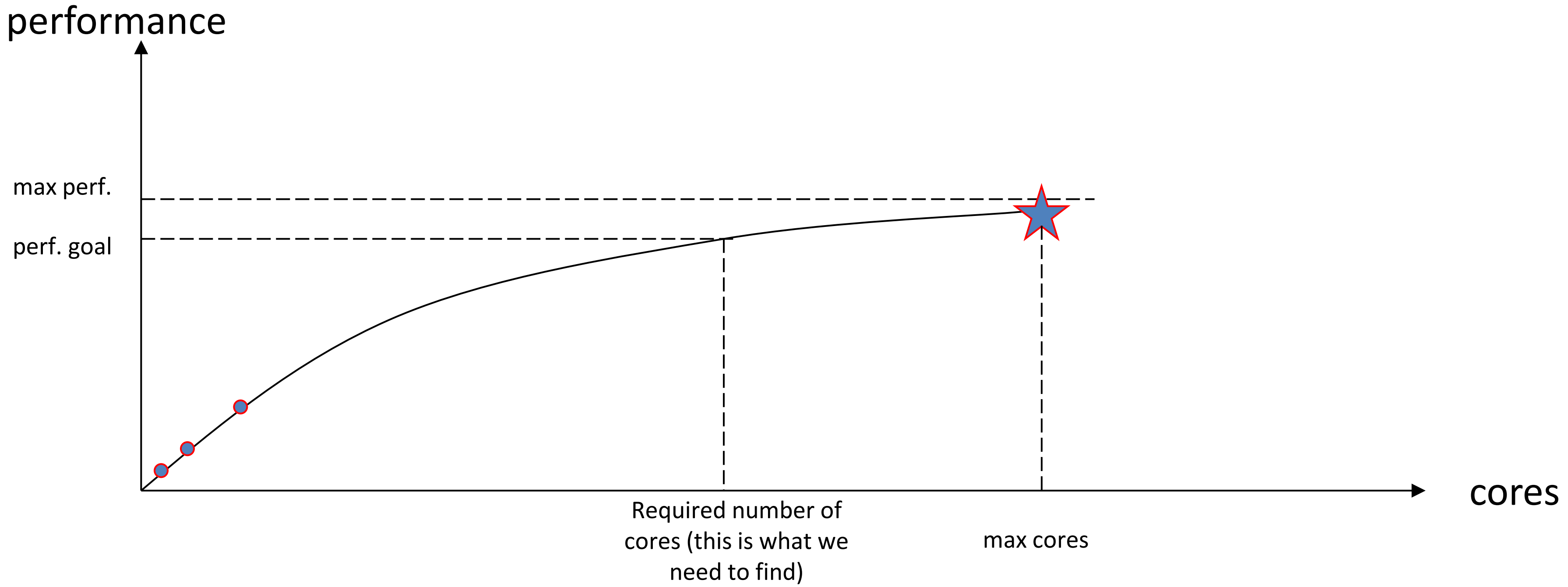
Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



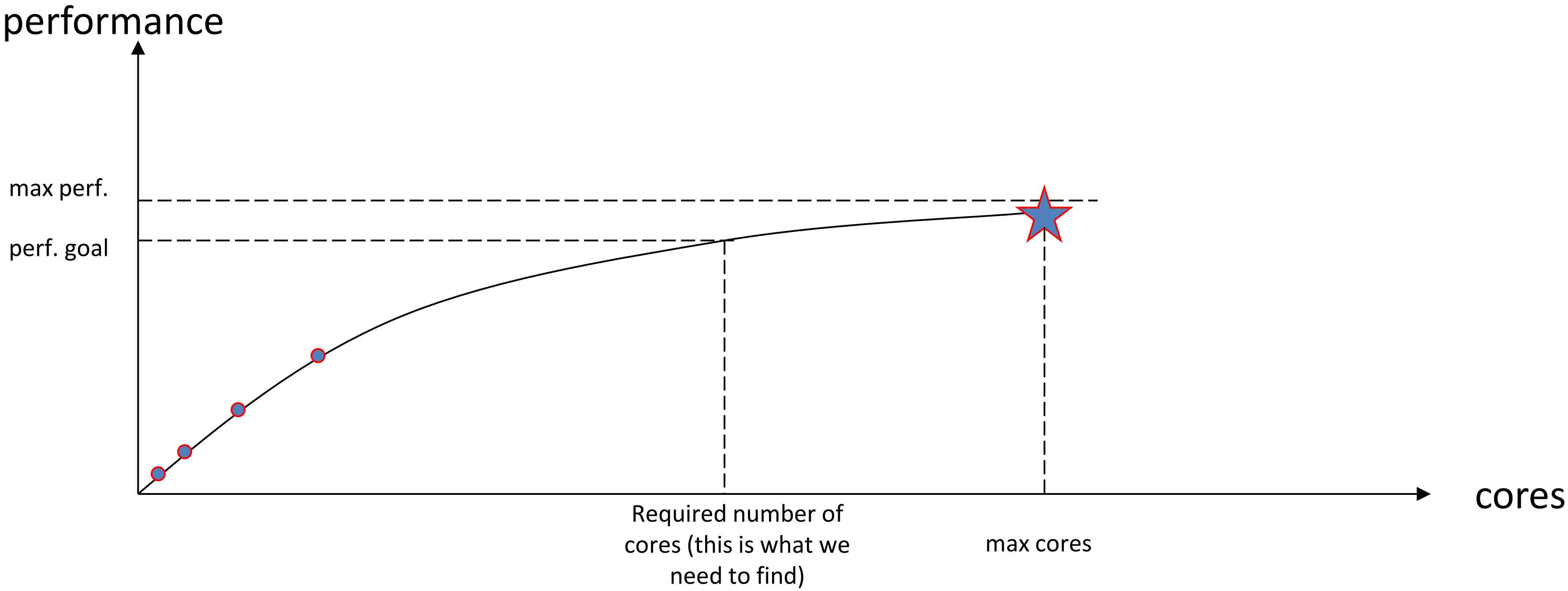
Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



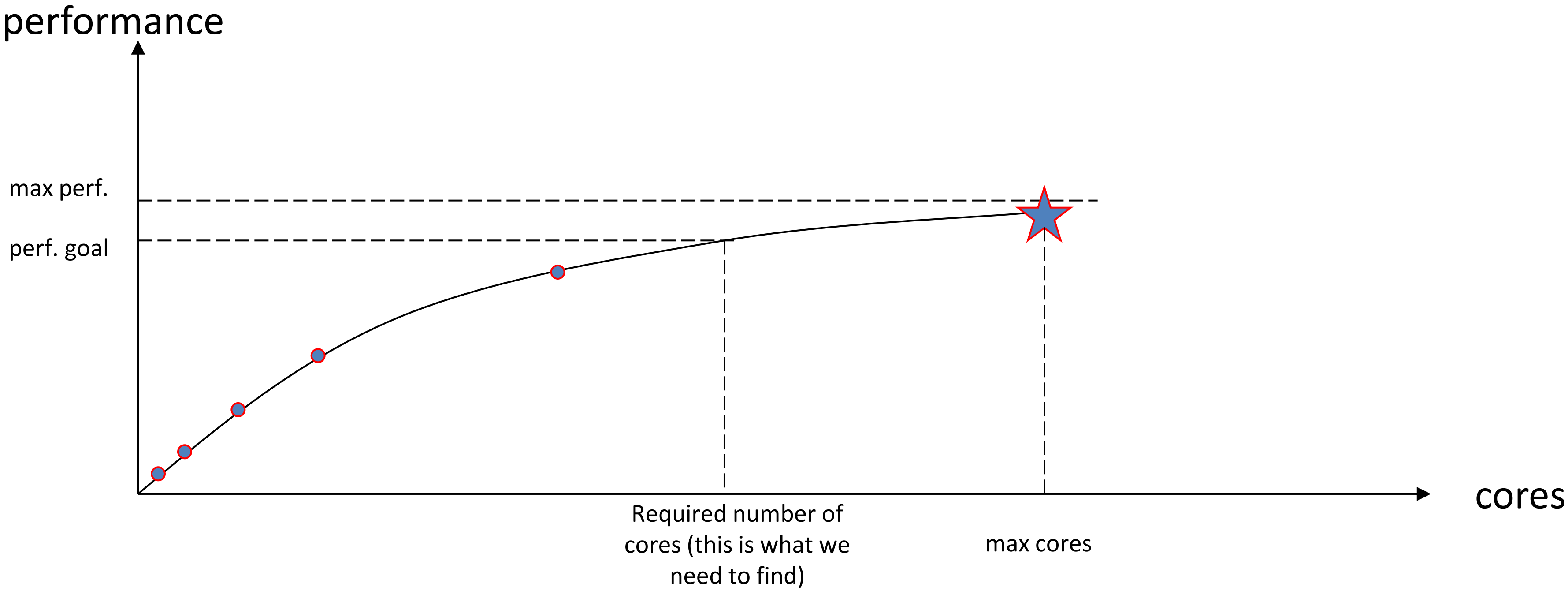
Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



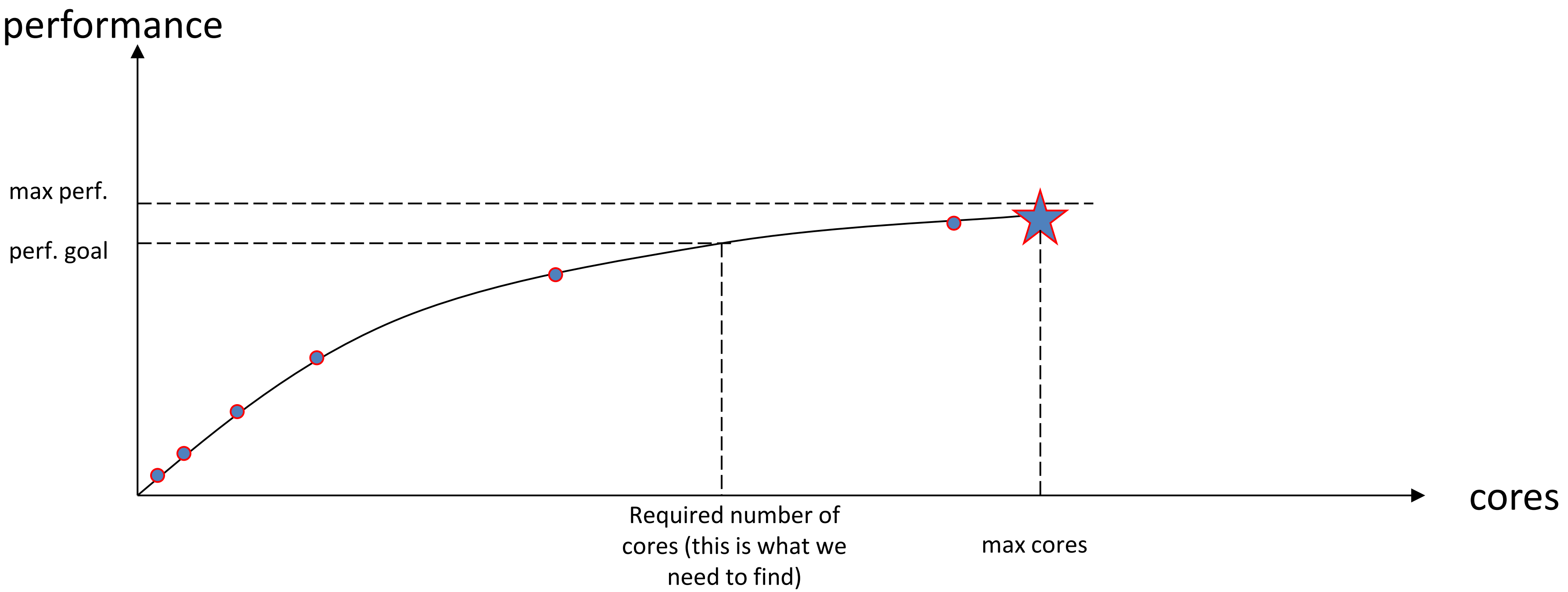
Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



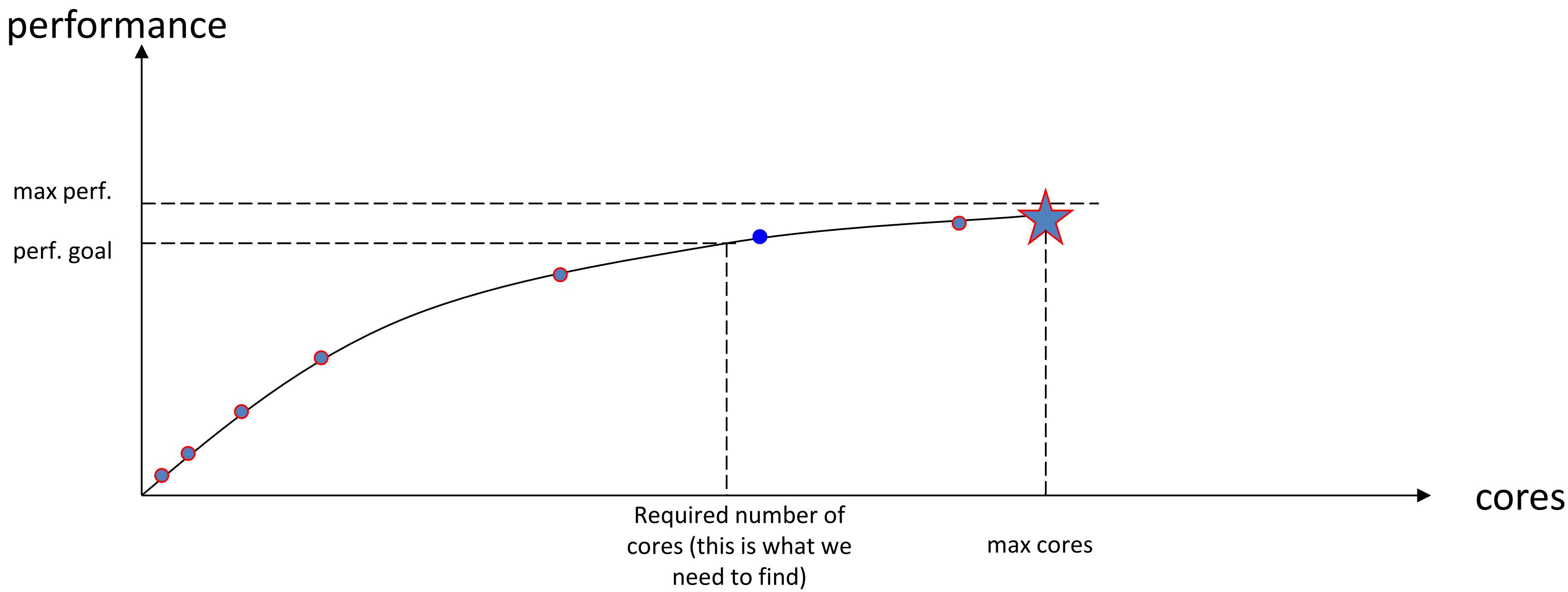
Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



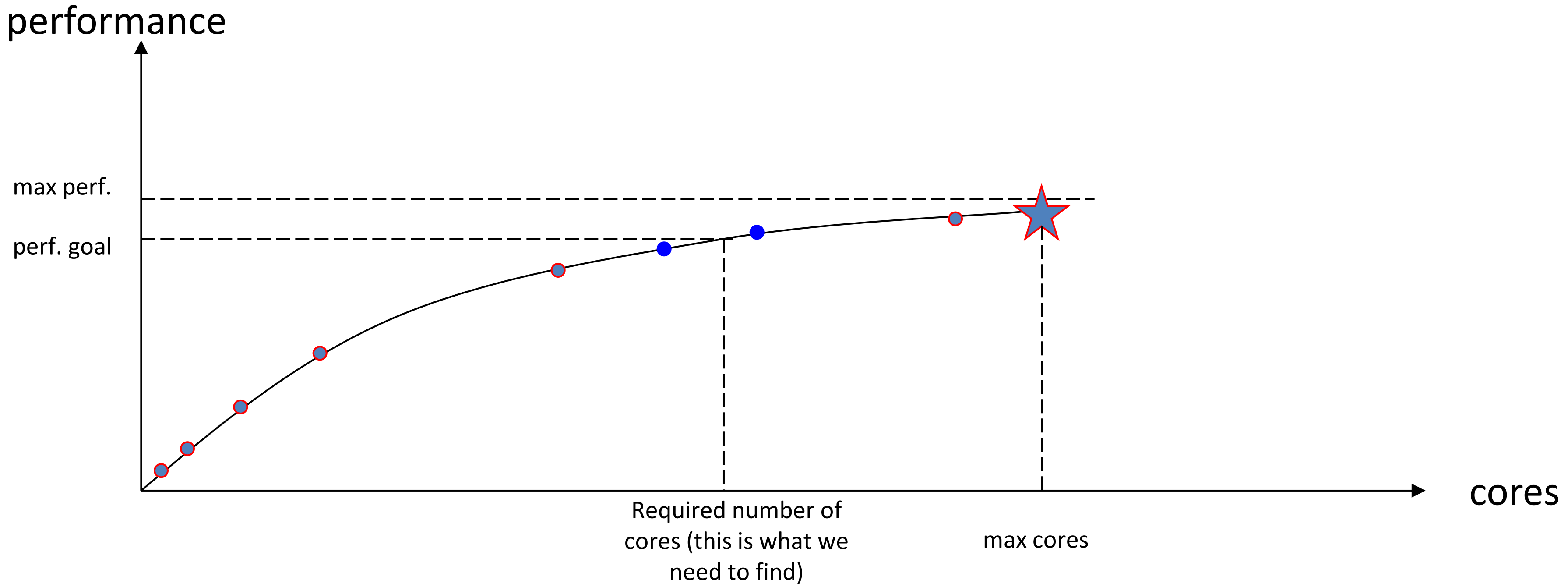
Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



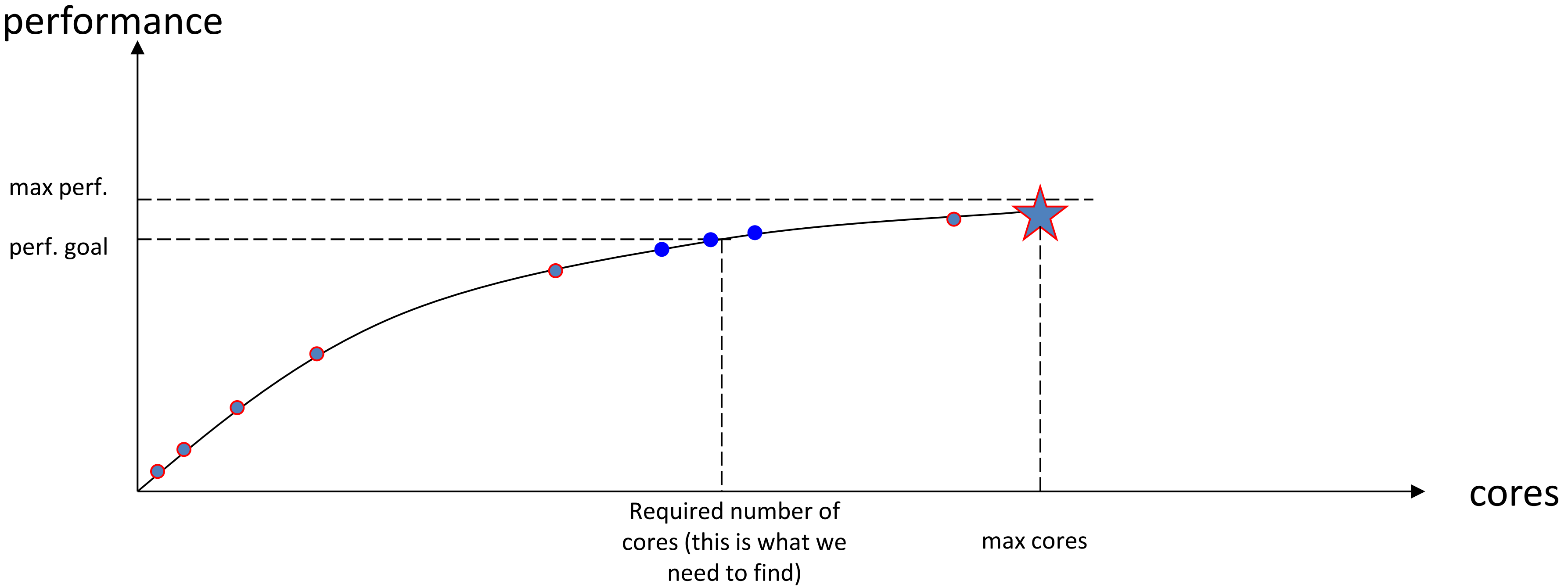
Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



Performance Modelling, Direct and Inverse

- Inverse performance modelling requires running a direct performance model multiple times
  - And each run can be expensive and time-consuming
- A two-stage iterative process



Performance Modelling, Direct and Inverse

- Performance models can be very different in their internal structure
- Ranging from tables and analytical formulae…
- …to neural networks
- Latest trends: use HW/SW co-design:
  - Run cycle-accurate simulations of codes (Verilog/VHDL simulations) or use FPGA prototyping
  - Then, use chip-level performance results to design higher levels of the system
  - Work in this field is being done at Sandia Laboratory: http://sst.sandia.gov/

Modularity of the CAD System

Performance model

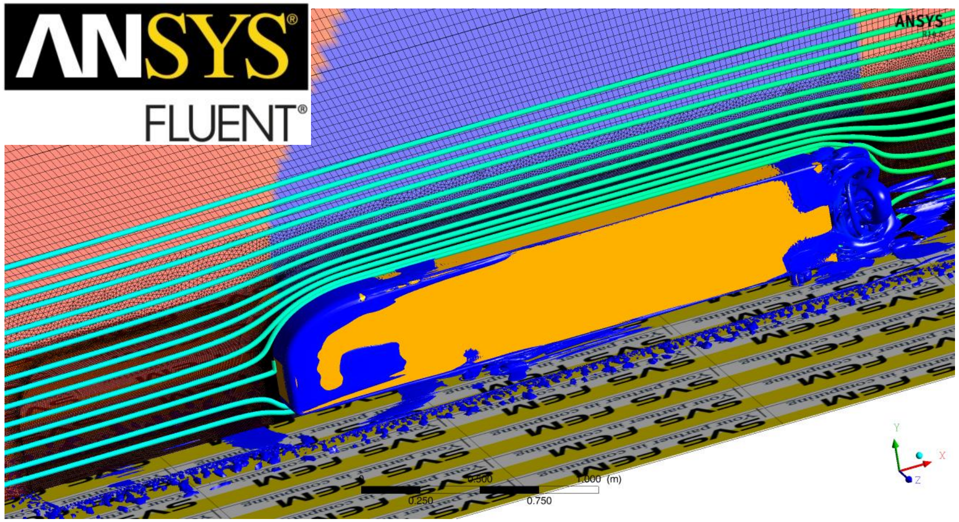


Image source: SVS FEM

A very simple performance model for ANSYS Fluent 13.0, for the "truck_111m" benchmark (External Flow Over a Truck Body)

See: http://ClusterDesign.org/ansys-fluent-simple-performance-model/

Performance model

A very simple performance model for ANSYS Fluent 13.0, for the “truck_111m” benchmark (External Flow Over a Truck Body)

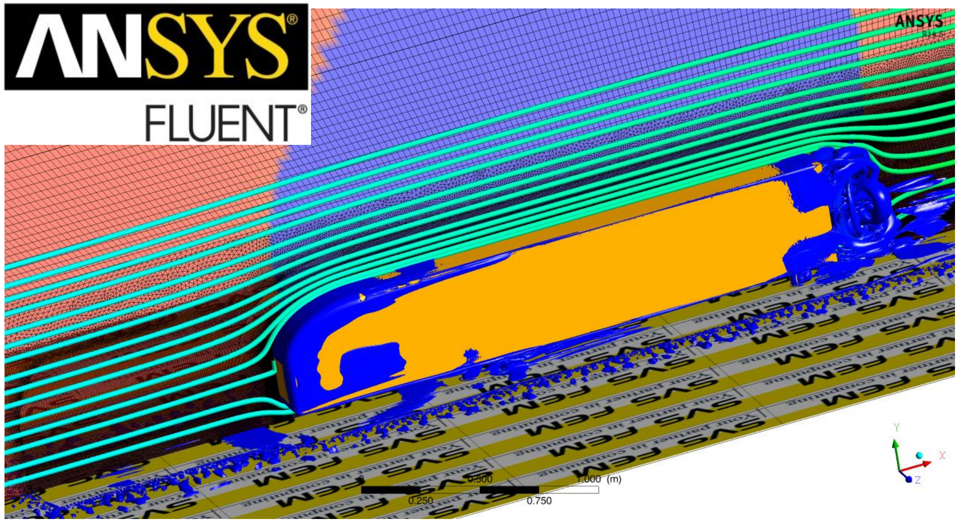


Image source: [SVS FEM]

```
software=ANSYS FLUENT 13.0.0
benchmark=truck_111m
perf_model_id=Demo model with linear approximation of efficiency, March 2012
cores=1204
network_tech=Infiniband-4X-QDR
performance_throughput_mode=False
time_to_solution=86.4
max_rating_at_cores=3072
max_rating=1943.7
performance=1000.5
```

Performance model

A very simple performance model for ANSYS Fluent 13.0, for the “truck_111m” benchmark (External Flow Over a Truck Body)

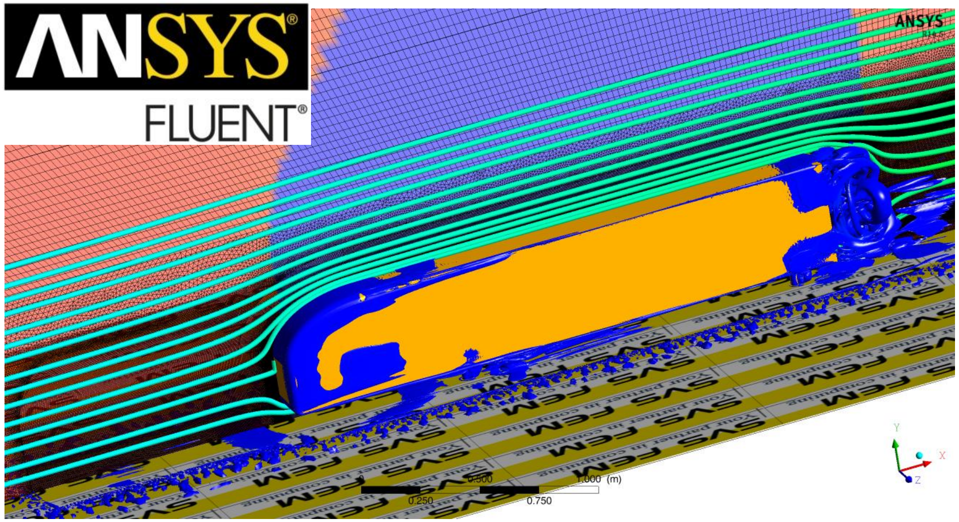


Image source: [SVS FEM]

```
software=ANSYS FLUENT 13.0.0
benchmark=truck_111m
perf_model_id=Demo model with linear approximation of efficiency, March 2012
cores=1204
network_tech=Infiniband-4X-QDR
performance_throughput_mode=False
time_to_solution=86.4
max_rating_at_cores=3072
max_rating=1943.7
performance=1000.5
```

← Requested by the user (performance=1000.5)

Performance model

A very simple performance model for ANSYS Fluent 13.0, for the "truck_111m" benchmark (External Flow Over a Truck Body)

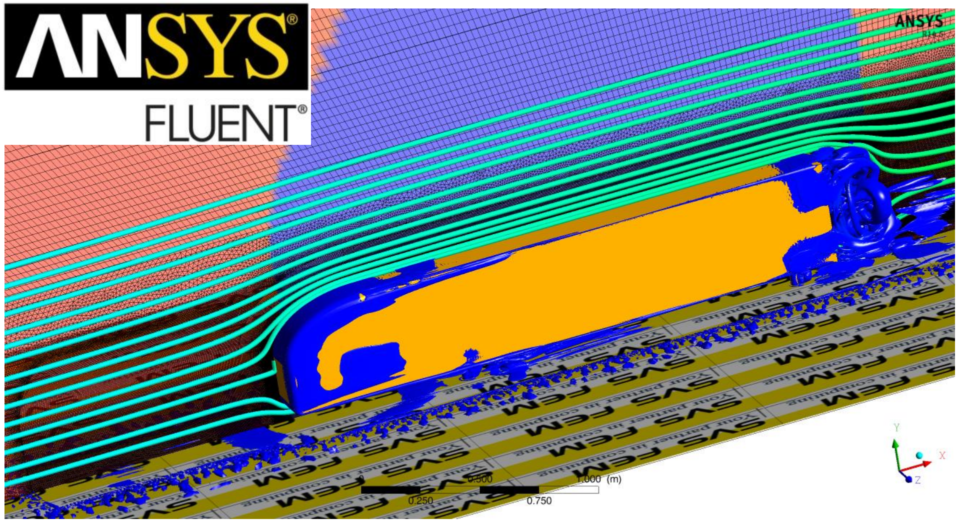


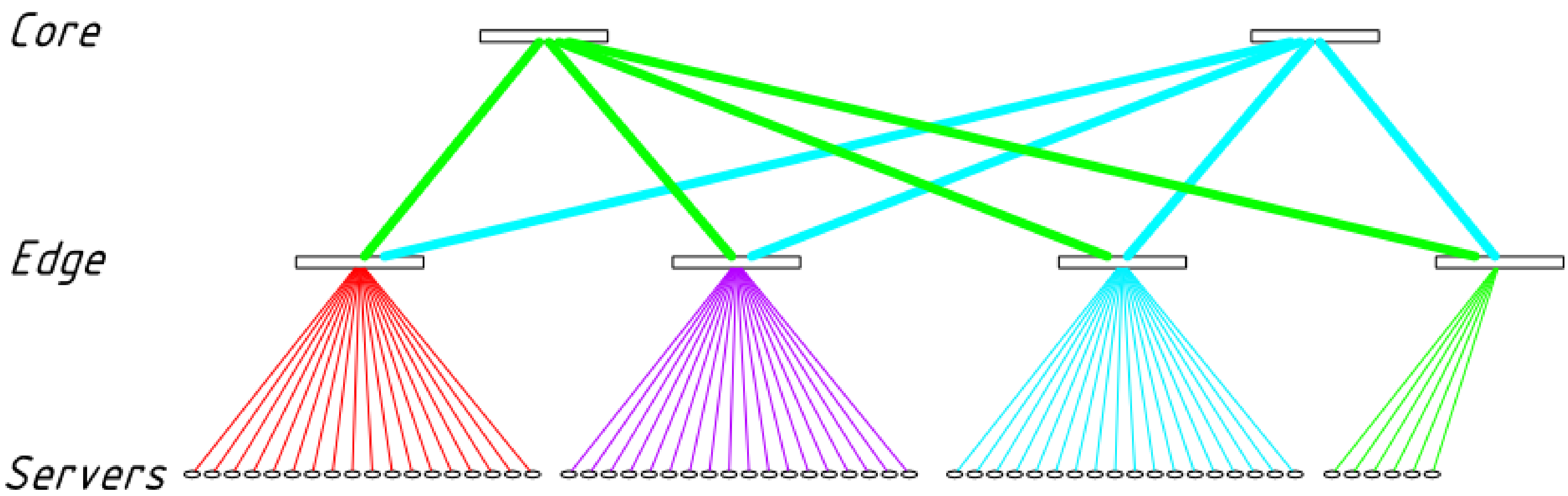
Image source: SVS FEM

```
software=ANSYS FLUENT 13.0.0
benchmark=truck_111m
perf_model_id=Demo model with linear approximation of efficiency, March 2012
cores=1204
network_tech=Infiniband-4X-QDR
performance_throughput_mode=False
time_to_solution=86.4
max_rating_at_cores=3072
max_rating=1943.7
performance=1000.5
```

cores=1204 ← Returned by the performance model

performance=1000.5 ← Requested by the user

Fat-tree and torus network design



See "Teach Yourself Fat-Tree Design in 60 Minutes", http://ClusterDesign.org/fat-trees/

K.S.Solnushkin. Automated Design of Two-Layer Fat-Tree Networks, 2013. arXiv:1301.6179 [cs.DC] (BIBTEX)

Fat-tree and torus network design

clusterdesign.org/cgi-bin/network/network

DISCLAIMER: USE THIS TOOL AT YOUR OWN RISK!

Now using ***fat-tree*** topology. [Change topology]

Network equipment vendor ID: mellanox

[Show fat-tree database]

Design your network

How many nodes will you initially have in your network?
Specify the number of compute nodes in your cluster. The more nodes you have, the more edge and core switches will be required. 600

Up to how many nodes will your network expand in the future?
If you plan to expand your cluster in the future (perhaps, in several stages), you can specify how many nodes it will have in its biggest configuration. The core level will be designed based on this number. If you plan for no expansion, simply leave this field equal to zero. Try different values for this expansion margin and observe how the required number of switches changes accordingly. 0

What is the maximum allowed blocking factor for your network?
Use "1" to design non-blocking networks. Fractional values are accepted (such as 1,0). Remember that for some parallel applications performance degradation may be higher than decrease in the total cost of your cluster computer, so creating a blocking network may not be worth it. 1 : 1

Prefer easily expandable networks (more intuitive) ☑

Output type:

- (•) Human-readable output
- ( ) Comma-separated values
- ( ) Name-value pairs

[Design your fat-tree network]

See also: detailed description: fat-trees, torus networks, price disclaimer and help for automated queries.

Fat-tree and torus network design

- What do the results look like?

Edge switches port distribution

To compute nodes:	18
To the core level:	18
Resulting blocking factor:	1.0

← Network for 10,000 nodes

Procurement Information

Model of edge switch:	Mellanox SX6036 (36 ports)
Initial number of edge switches:	556
Model of core switch:	Mellanox SX6536 (with 558 ports)
Number of core switches:	18
Cables:	20008

Human-readable

Quality Metrics

Links between core and edge layers run in bundles of (denotes wiring regularity):	1
Core level port utilization (denotes used ports), percent:	100

Technical Characteristics

Power of network equipment, watts:	267138
Weight of network equipment, kilograms:	10089.2
Size of network equipment, in rack mount units:	1114
Cost of network (switches and cables):	17427100

Fat-tree and torus network design

- What do the results look like?

```
max_network_blocking_factor=1.0
max_network_cost=0
max_network_equipment_size=0
max_network_power=0
max_network_weight=0
network_blocking_factor=1.0
network_core_level_utilization=100
network_core_ports=558
network_core_switch_count=18
network_core_switch_model=Mellanox SX6536 (with 558 ports)
network_core_switch_size=31
network_cost=17427100
network_edge_ports_to_core_level=18
network_edge_ports_to_nodes=18
network_edge_switch_count=556
network_edge_switch_model=Mellanox SX6036 (36 ports)
network_edge_switch_size=1
network_edge_uniform_distribution=False
network_equipment_size=1114
network_expandable_to=10008
network_link_count=20008
network_links_run_in_bundles=1
network_objective_function=17427100.0
network_power=267138
network_prefer_expandable=True
network_spare_ports=8
network_topology=fat-tree
network_weight=10089.2
nodes=10000
nodes_future_max=10000
```

← Network for 10,000 nodes

Readable by SADDLE

UPS Sizer

Choose the optimal UPS for your computing needs

What is the total power of your computing hardware that requires UPS backup, in watts?
Type here the total power, in watts, of all hardware that needs backup electrical power: compute nodes, network hardware, storage systems, and -- optionally, but recommended -- cooling systems. [12500]

How long should be the battery backup time, in seconds?
If backup time is not important, leave this blank (or zero) [0]

Calculate

See also: detailed description, help for automated queries.

Learn more at: http://ClusterDesign.org/ups-sizing/

UPS Sizer

Choose the optimal UPS for your computing needs

What is the total power of your computing hardware that requires UPS backup, in watts?
Type here the total power, in watts, of all hardware that needs backup electrical power: compute nodes, network hardware, storage systems, and -- optionally, but recommended -- cooling systems. 12500

How long should be the battery backup time, in seconds?
If backup time is not important, leave this blank (or zero) 0

Calculate

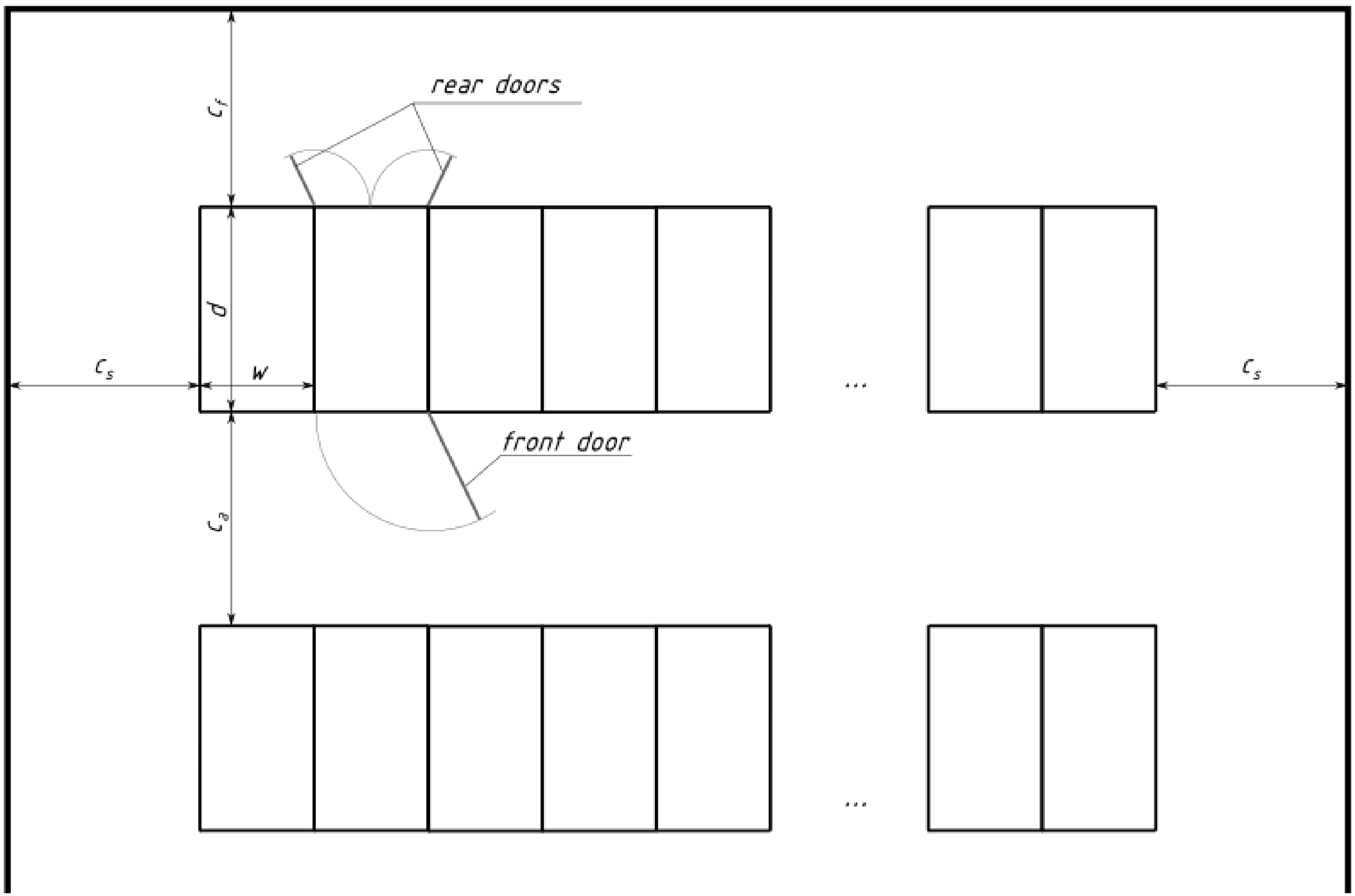
See also: detailed description, help for automated queries.

```
ups_backup_time=720
ups_cost=1586000
ups_cost_per_kw=1057.3
ups_heat=90000
ups_model=Liebert APM (up to 45kW)
ups_partitioning=33*45000+1*15000
ups_power_rating=1500000
ups_size_racks=34
ups_weight=16422
```

← Output for 1,5 MW

Learn more at: http://ClusterDesign.org/ups-sizing/

Floor Planning



Floor Planning

Calculate the floorspace size required to house your racks

How many racks do you need to place on the floor? The algorithm will try to find floorspace dimensions as close to a square shape as possible.	12
What is the rack width, in metres? Use the default value or enter your own.	0.6
What is the rack depth, in metres? Use the default value or enter your own.	1.2
Clearances on the sides of rack rows, in metres: Makes sure that sides of rack rows are not too close to the walls. Use the default value or enter your own.	1
Clearances in front of the first row and behind the last row, in metres: Allows to move freely between rack rows and walls. Use the default value or enter your own.	1
Aisle width, in metres: Allows to open rack doors and extract equipment. Use the default value or enter your own.	1
Maximal length of a contiguous block of racks, in metres: Prevents too long rack rows that make it hard to perambulate your possessions. Use the default value or enter your own.	6

Calculate

Floor Planning

Calculate the floorspace size required to house your racks

How many racks do you need to place on the floor? The algorithm will try to find floorspace dimensions as close to a square shape as possible.	12
What is the rack width, in metres? Use the default value or enter your own.	0.6
What is the rack depth, in metres? Use the default value or enter your own.	1.2
Clearances on the sides of rack rows, in metres: Makes sure that sides of rack rows are not too close to the walls. Use the default value or enter your own.	1
Clearances in front of the first row and behind the last row, in metres: Allows to move freely between rack rows and walls. Use the default value or enter your own.	1
Aisle width, in metres: Allows to open rack doors and extract equipment. Use the default value or enter your own.	1
Maximal length of a contiguous block of racks, in metres: Prevents too long rack rows that make it hard to perambulate your possessions. Use the default value or enter your own.	6

Calculate

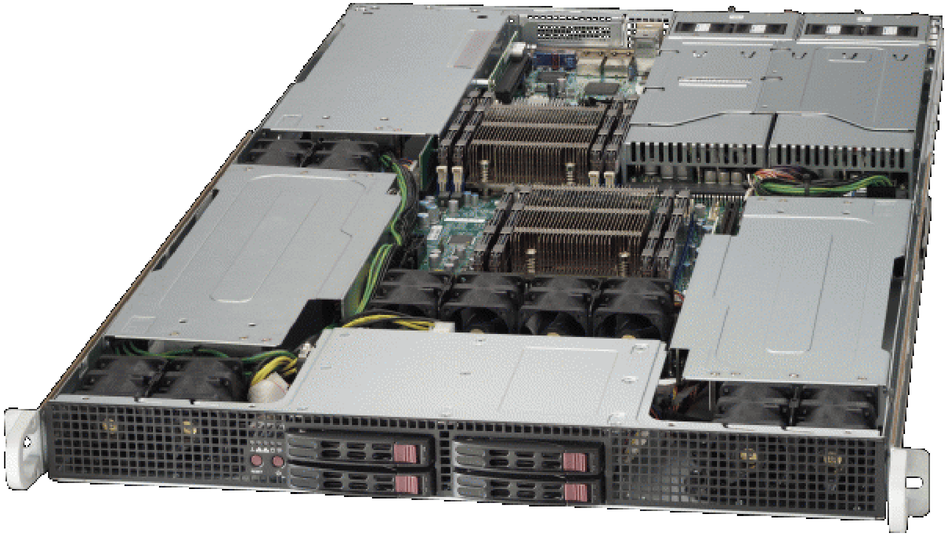
```
c_a=1.0
c_f=1.0
c_s=1.0
d=1.2
floor_space_plan_formula=(9+8)*6
floor_space_size=187.44
floor_space_x_dimension=13.2
floor_space_y_dimension=14.2
gaps=1
l_xc=6.0
racks=100
racks_per_row=17
rows=6
segments=2
w=0.6
```

Learn more at: http://ClusterDesign.org/floorplanning/

Graph representation of configurations

Or how to choose only compatible components

Choosing only compatible components

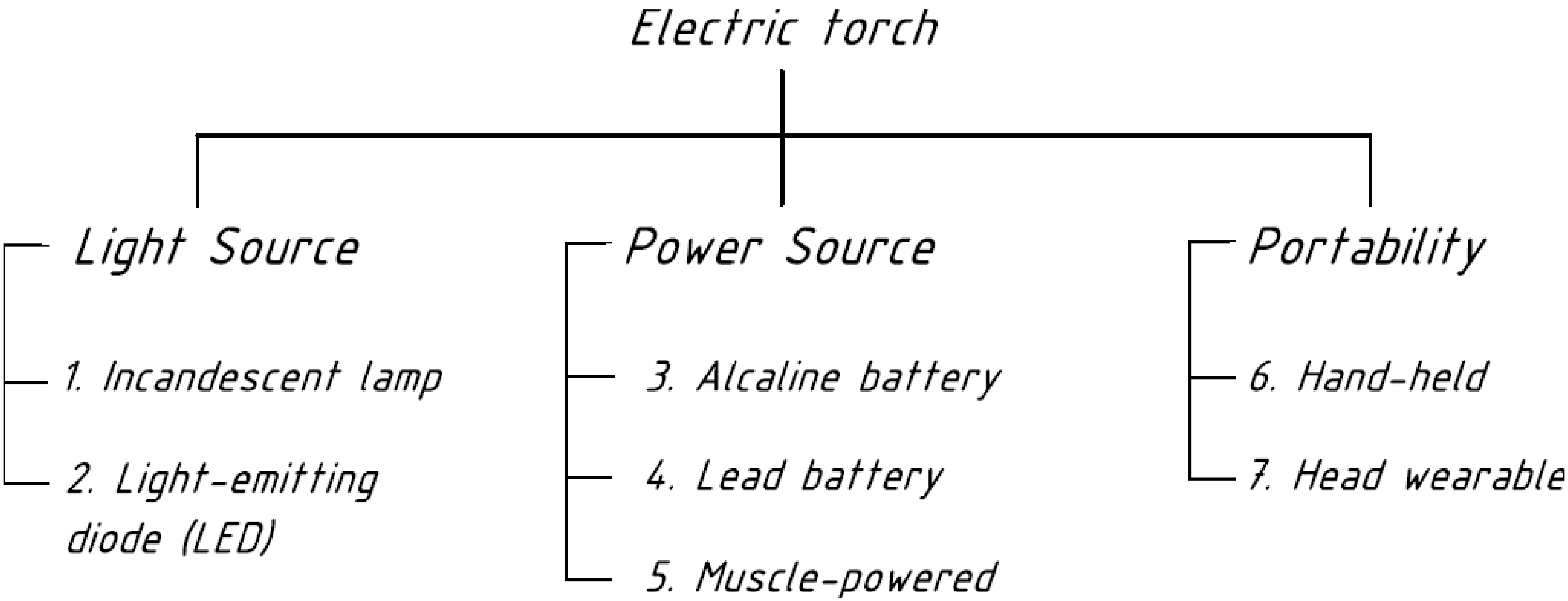


Colfax CX1350s-XP5 1U Rackmount Server

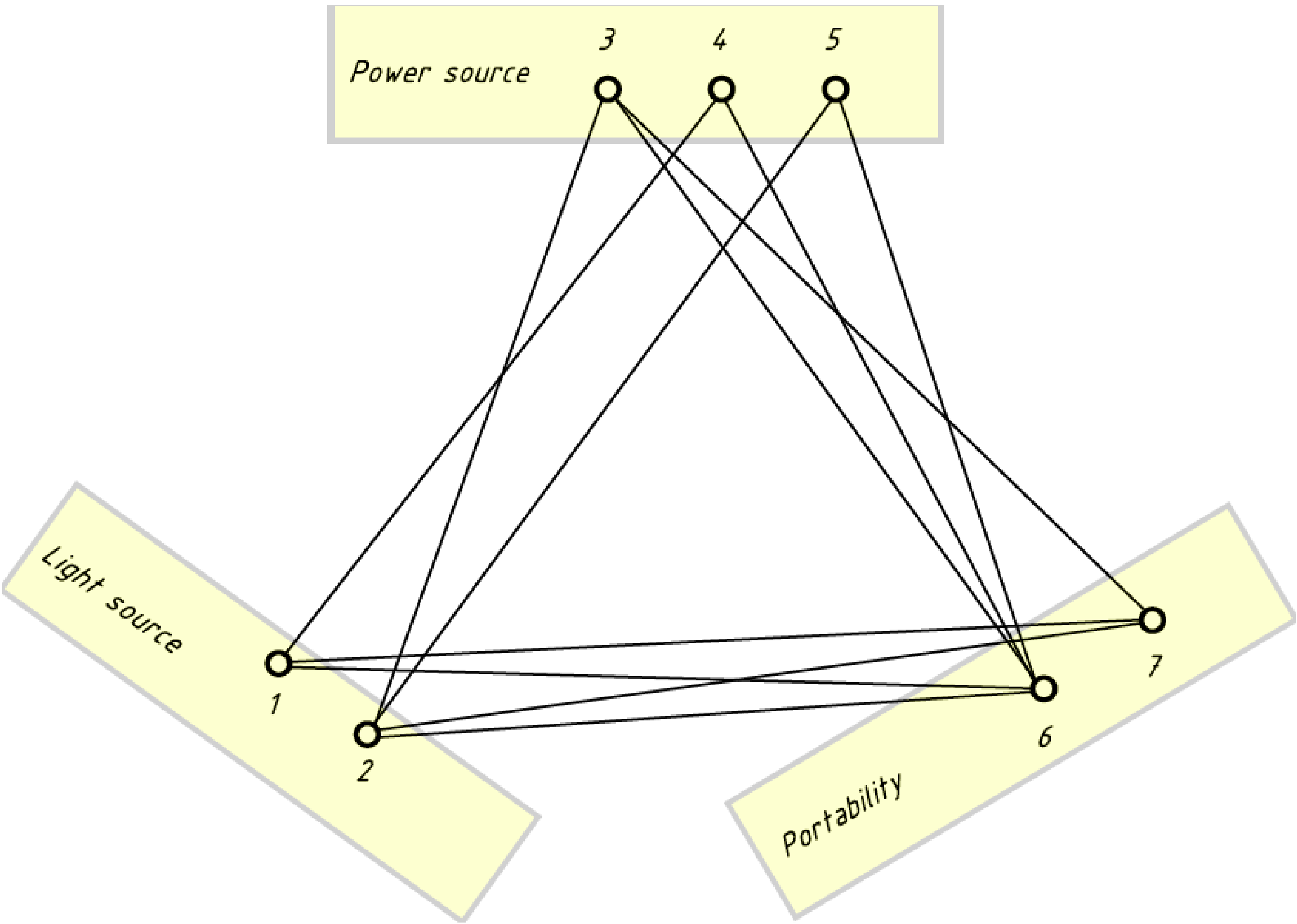
- Quite resembles compute nodes in [Tianhe-2](), the fastest supercomputer in the world as of November 2014
- Up to 2 Intel® Xeon® Processors E5-2600 V2 Series
- Up to 3x Intel® Xeon Phi™ Coprocessors

Image source: http://www.colfax-intl.com/nd/Servers/CX1350s-XP5.aspx

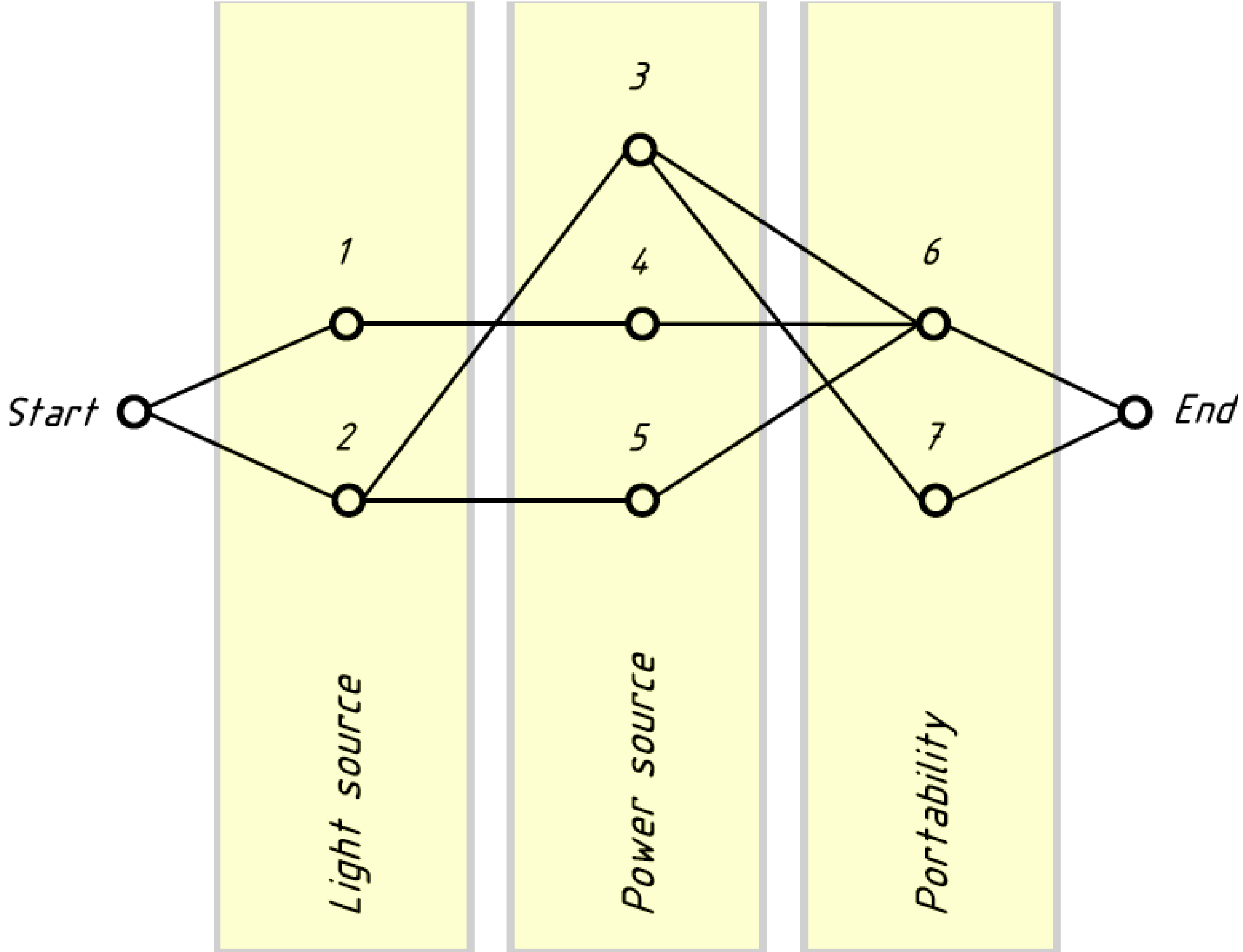
Graph Representation



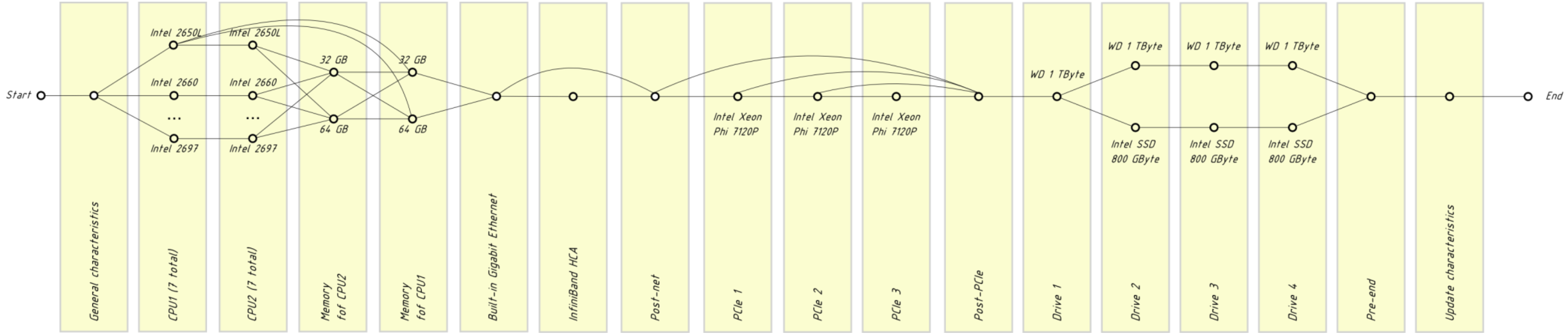
Graph Representation



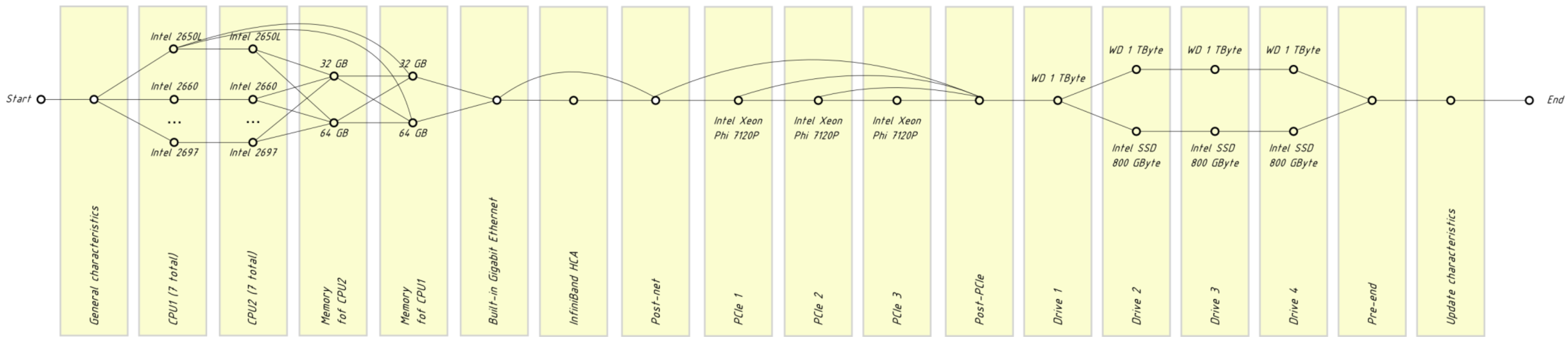
Graph Representation



Graph Representation

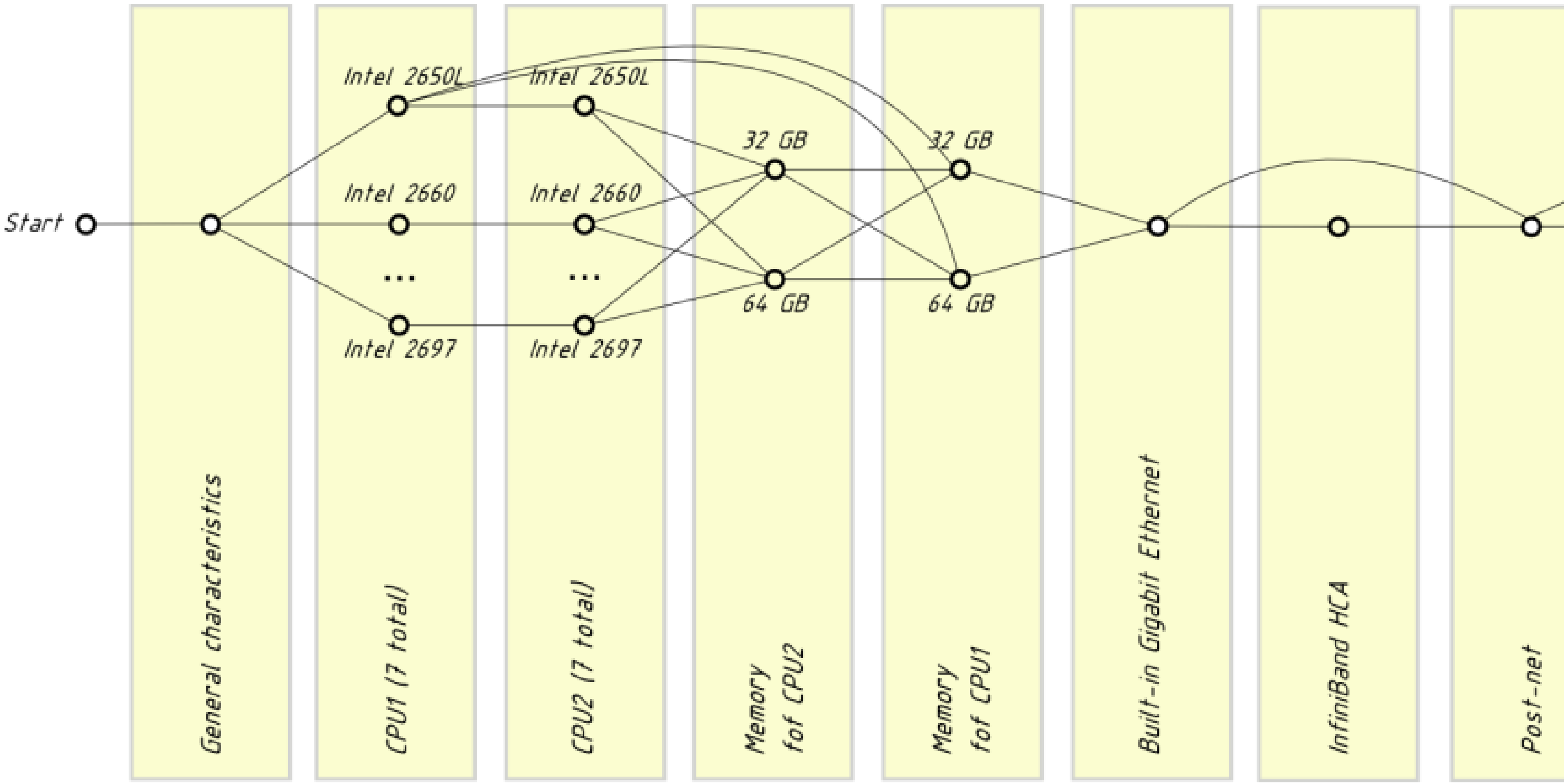


Graph Representation

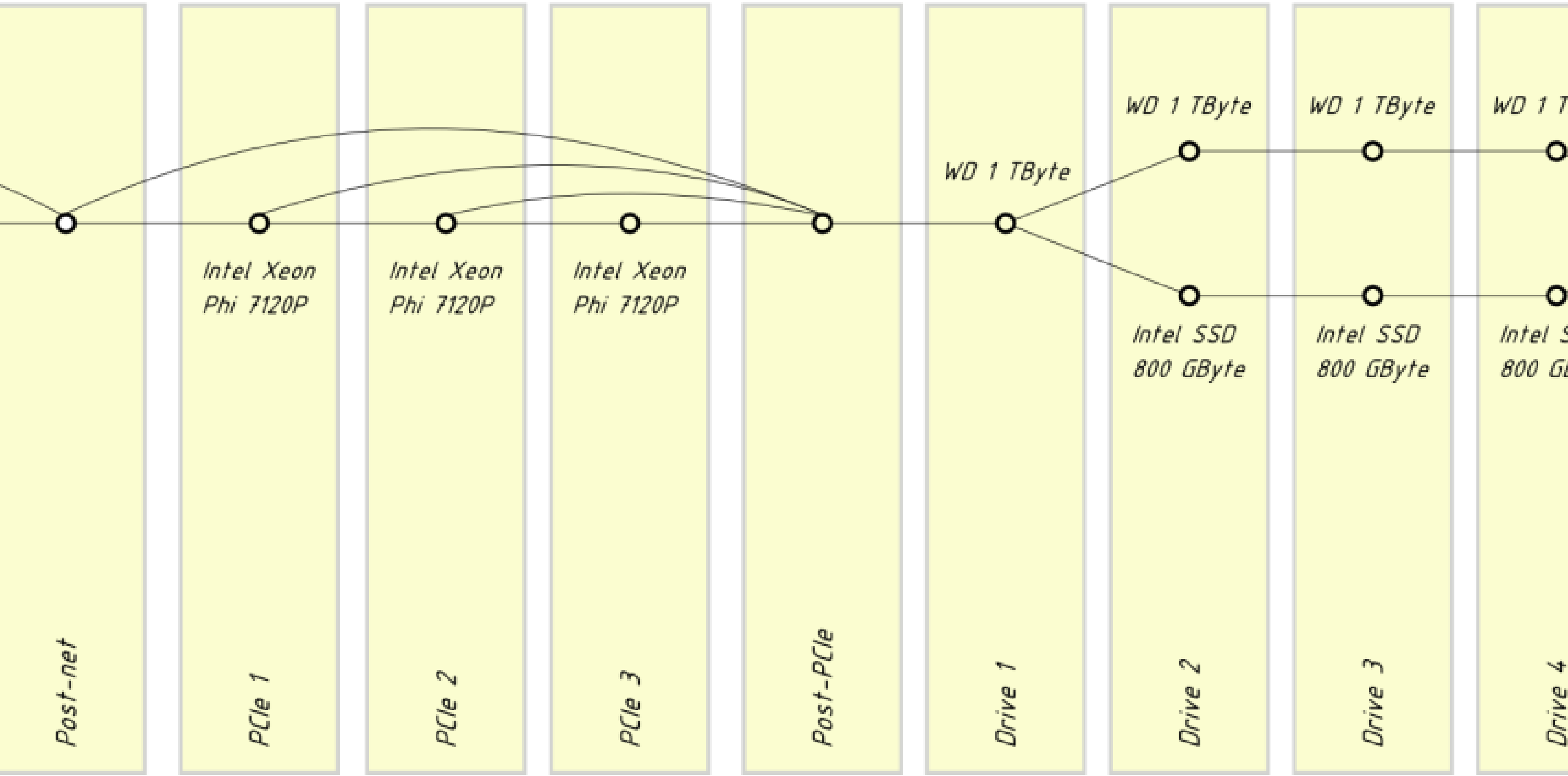


- ***Partitions*** are components that can be configured in several ways. Example: "CPU1"
- ***Vertices*** in a partition represent possible configurations of a component
  - Example: 32 GB or 64 GB of memory
  - Can be fictitious: used to assign expressions that will be evaluated during graph traversal
- ***Edges*** represent compatibility between components or, more generally, "what can be connected to what"

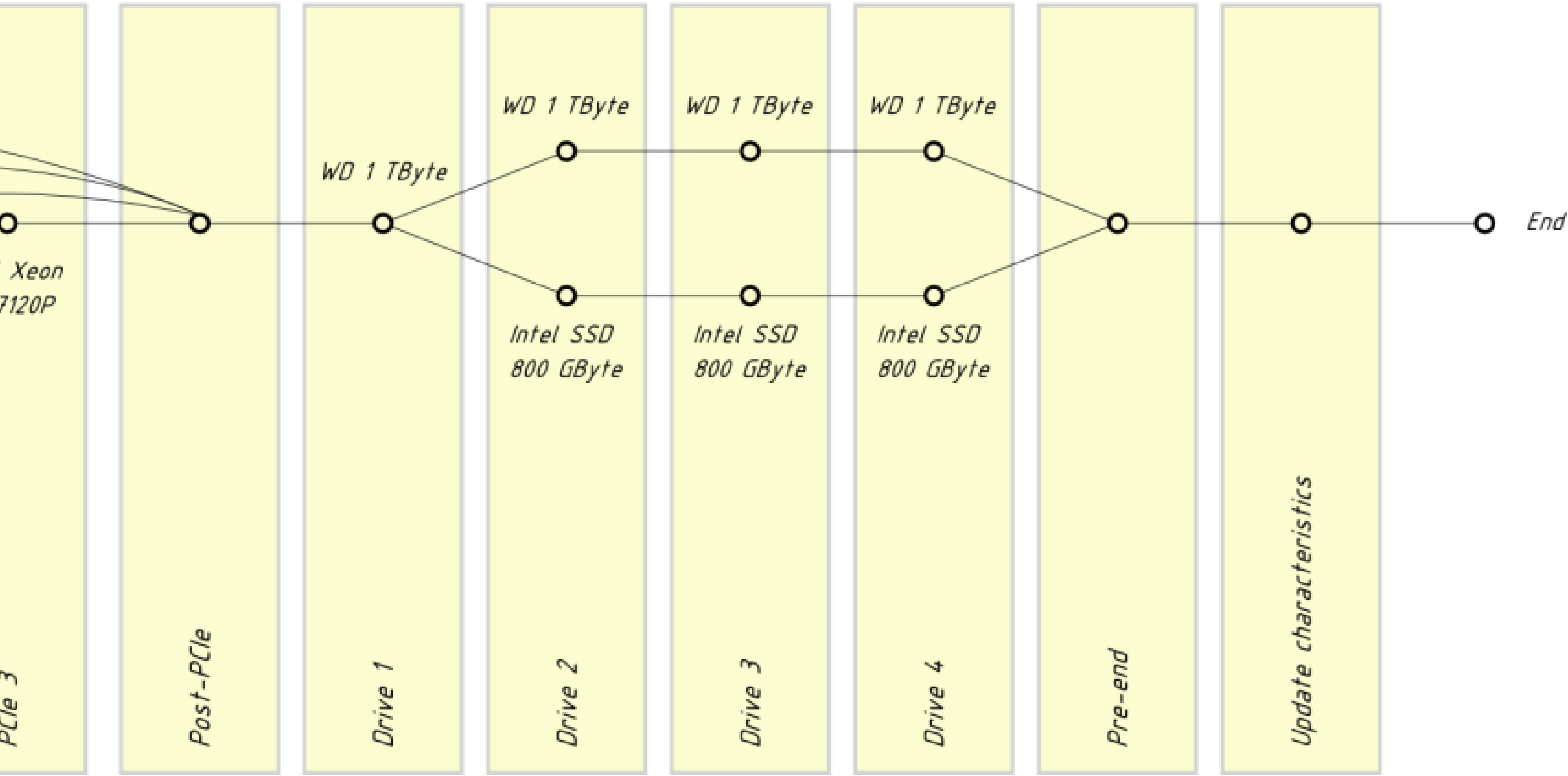
Graph Representation Zoom-in



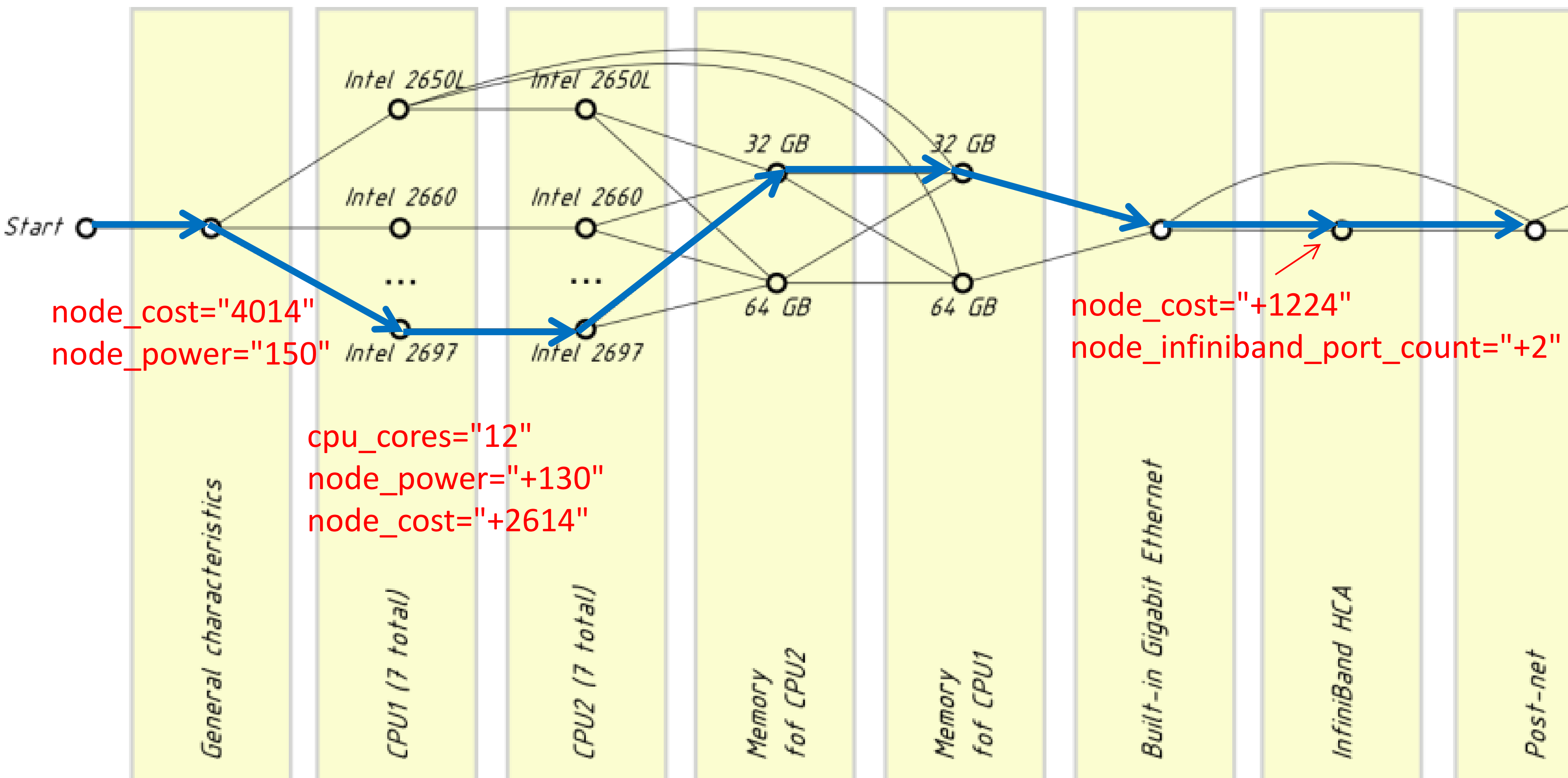
Graph Representation Zoom-in



Graph Representation Zoom-in



Graph Traversal and Metric Evaluation



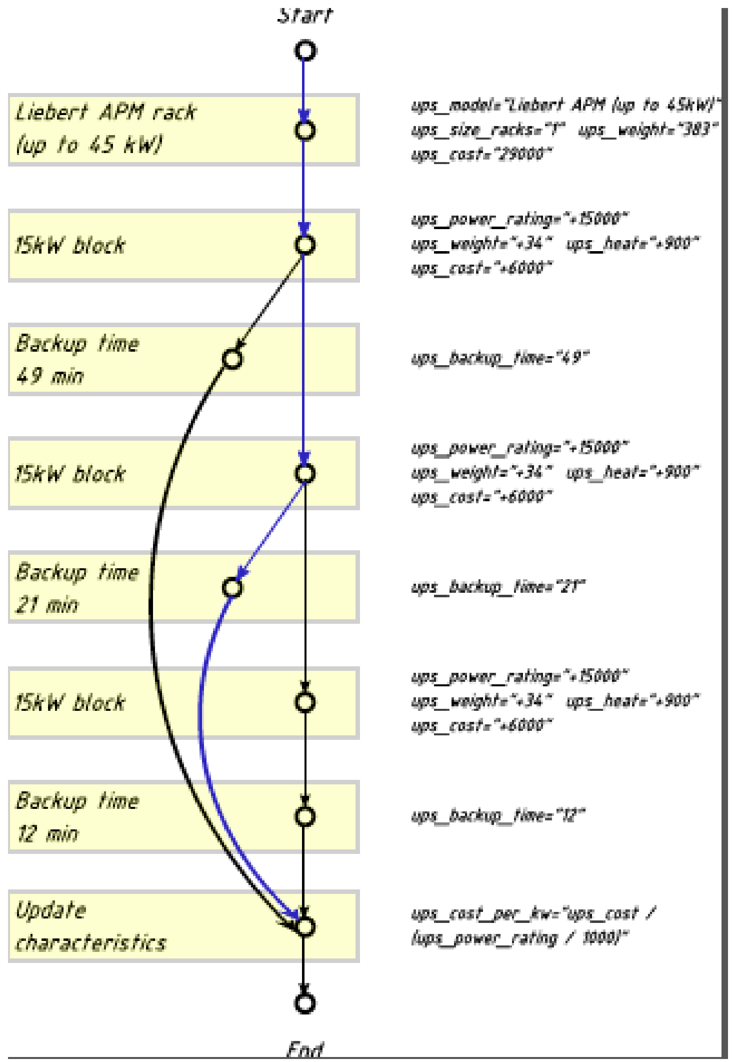
Defining Vertices and Edges In XML Is Actually Easy

```
<!-- Intel Xeon Phi Coprocessor 7120P (16GB, 1.238 GHz, 61
core). Adds 1208 GFLOPS of peak floating-point performance -->
<item node_cost="+4129" node_peak_performance="+1208"
accelerator_model="+ Intel Xeon Phi 7120P" accelerator_vendor=
"Intel" accelerator_count="+1" node_power="+300">Intel Xeon Phi
Coprocessor 7120P</item>
```

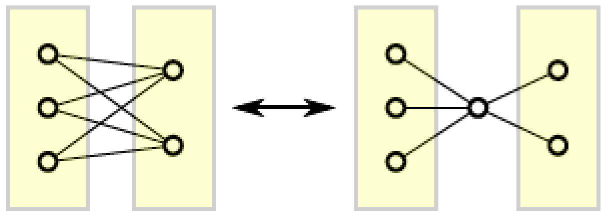
```
<!-- Memory options for CPU1. We make sure that CPU1 and CPU2
have the same amount of memory (for symmetry), although in
general this is not strictly necessary. "to" is not used,
hence defaults to "from". Copies of vertices are created
automatically. -->
  <edge from="32 GB" from-partition="RAM_FOR_CPU2"
  to-partition="RAM_FOR_CPU1"></edge>
  <edge from="64 GB" from-partition="RAM_FOR_CPU2"
  to-partition="RAM_FOR_CPU1"></edge>

  <!-- If only CPU1 is present, it can be connected to
  RAM_FOR_CPU1. -->
  <connect from-partition="CPU1" to-partition="RAM_FOR_CPU1"
  ></connect>
```

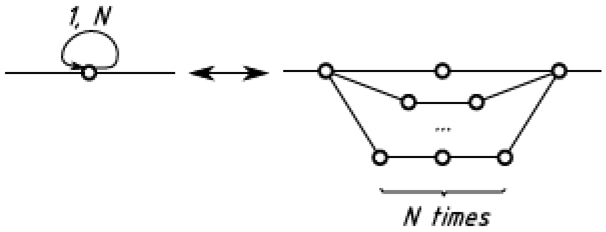
Graph representation for UPS system



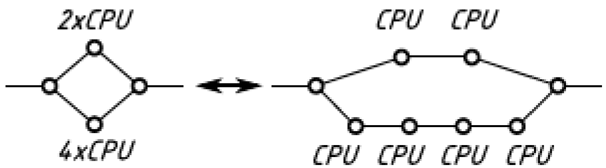
Graph operations



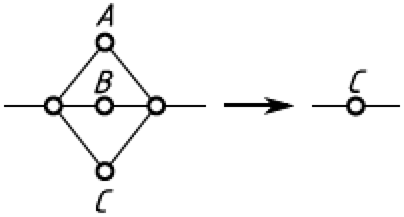
(a) Introducing an auxiliary vertex instead of a biclique



(b) Loop transformed into a sequence of paths



(c) Notation to represent graph copies



(d) Choosing a locally optimal vertex

SADDLE, a CAD tool in your pocket

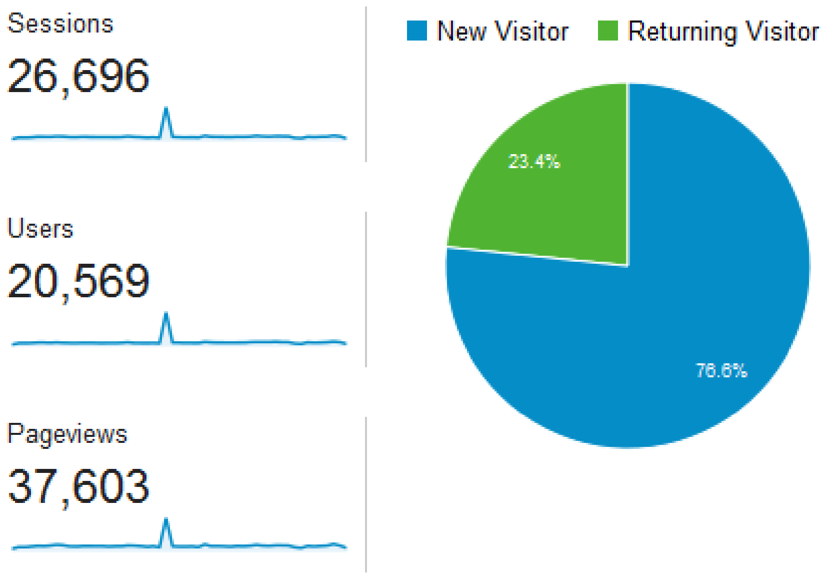
- SADDLE itself is a bunch of Python scripts:

bom.py	18 КБ	Python File
cabling.py	16 КБ	Python File
callmod.py	6 КБ	Python File
colours.py	3 КБ	Python File
common.py	3 КБ	Python File
conflist.py	17 КБ	Python File
database.py	6 КБ	Python File
dbcliquery.py	2 КБ	Python File
eqgroups.py	21 КБ	Python File
evaluate.py	17 КБ	Python File
floorplan.py	3 КБ	Python File
multipart.py	5 КБ	Python File
parsetab.py	10 КБ	Python File
racks.py	39 КБ	Python File
saddle.py	25 КБ	Python File
strconst.py	5 КБ	Python File
svgoutput.py	38 КБ	Python File

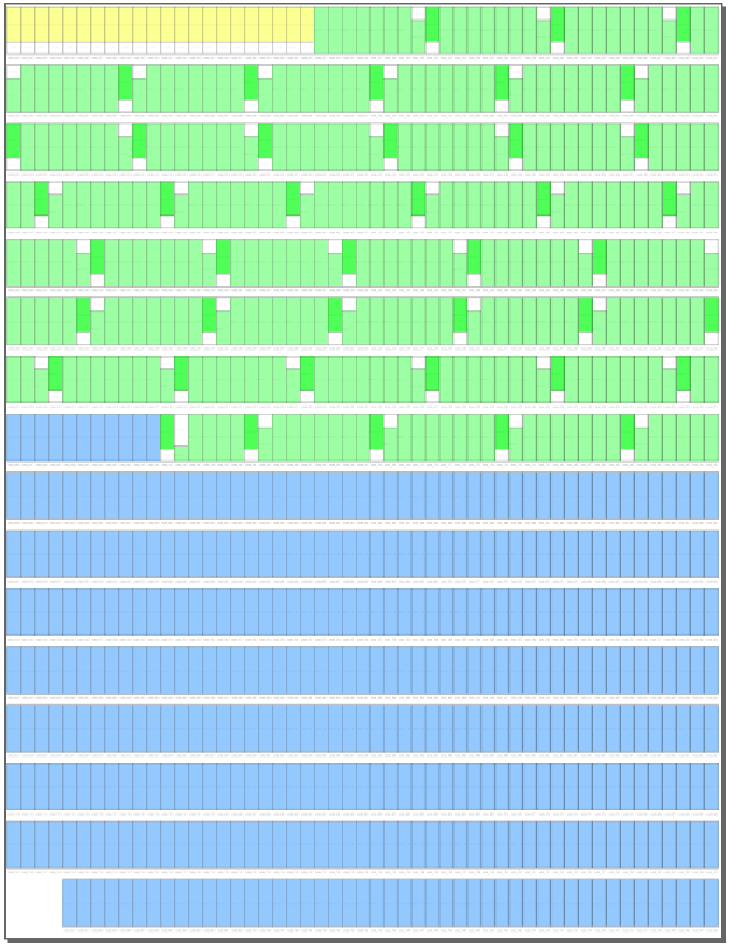
- Design modules invoked by SADDLE are separate programs, queried via network for flexibility

SADDLE, a CAD tool in your pocket

- SADDLE is hosted at *ClusterDesign.org*
- 20,000+ visitors in the past 12 months
- ~100 downloads of the software suite



Let's design a machine like Tianhe-2: 55 PFLOPS and 3 Intel Xeon Phi accelerators per node



Let's design a machine like Tianhe-2: 55 PFLOPS and 3 Intel Xeon Phi accelerators per node

```
Design-wide metrics
-------------------

System lifetime, years:             3
Electricity price per kWh*hour:     0.13
Rack stationing costs per year:     3,000
Capital expenditures:               460,255,815 (86.01% of TCO)
Operating expenditures:             74,890,723 (13.99% of TCO)
Total cost of ownership:            535,146,538
Power, W:                           19,781,853
Tomato equivalent, kg/day:          7,912.7 *
Weight:                             329,887
```

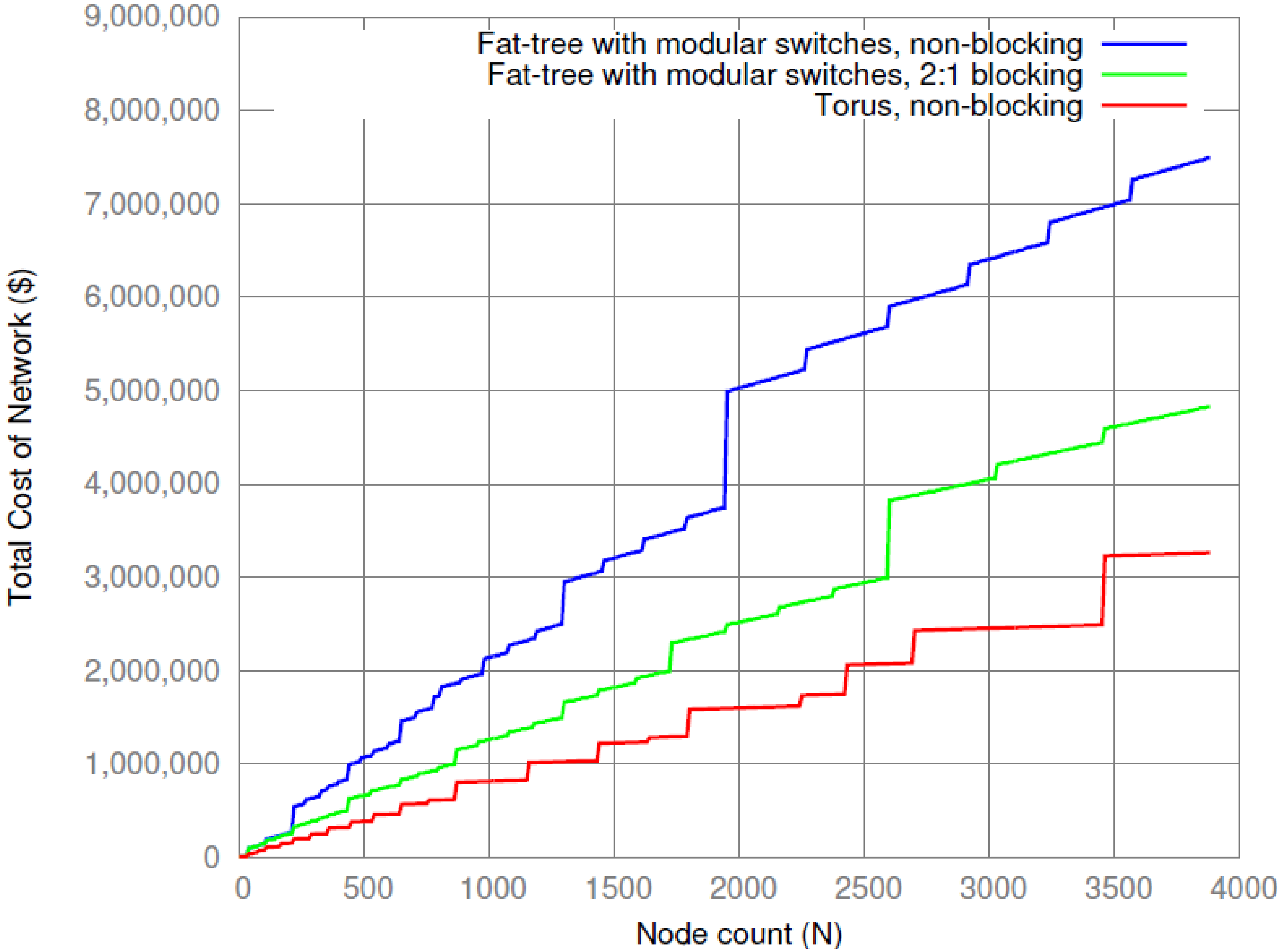
* SuperMUC, for example, can produce enough tomatoes for the whole city of Garching

Conclusions for SADDLE

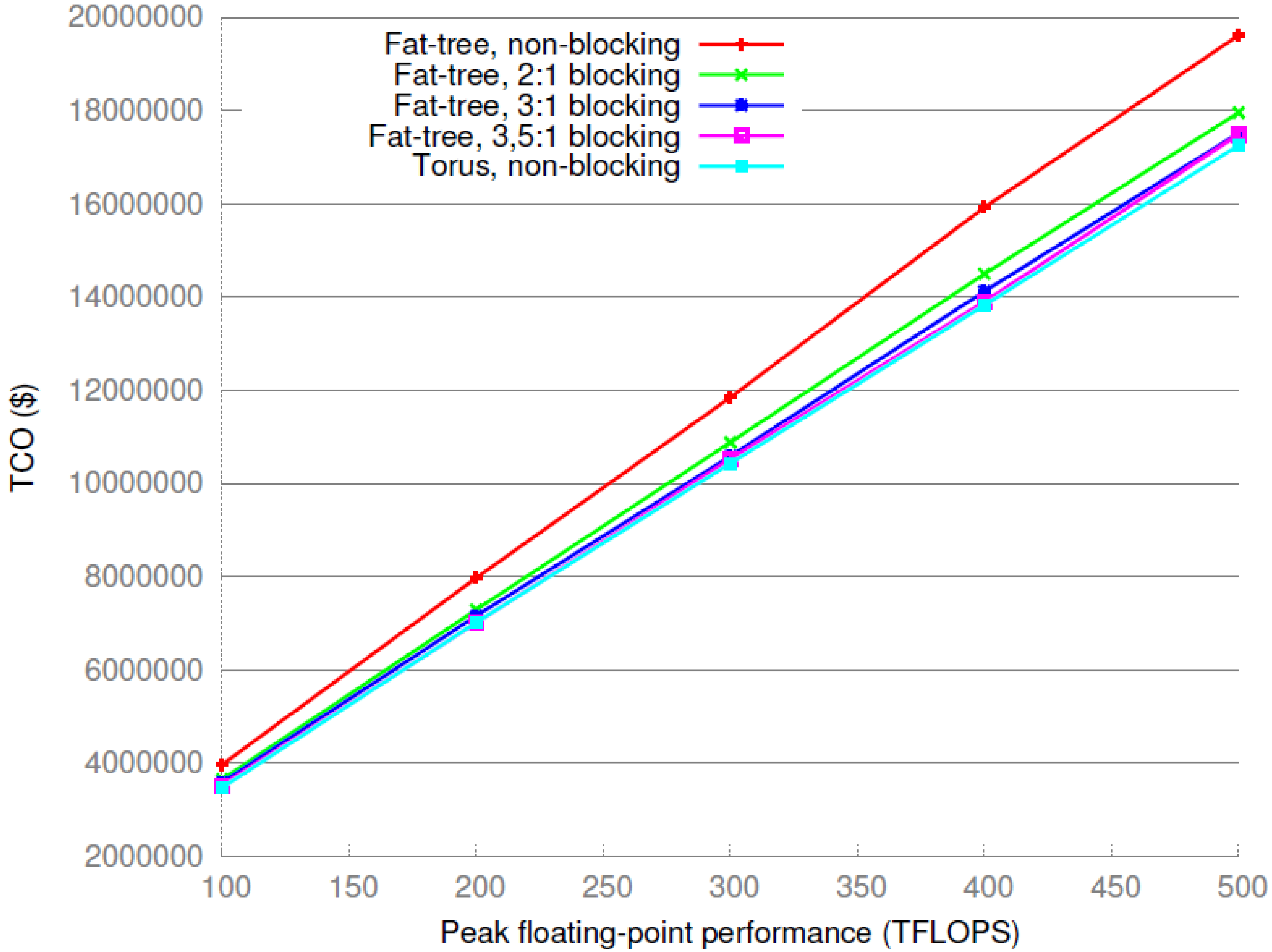
- Now, we can design cluster computers without guesswork
- The exactness of estimations is guaranteed and depends only on the quality of your performance model and the novelty of hardware prices in the database
- SADDLE can design any data centres and warehouse-scale computers: Hadoop clusters, web hosting farms, etc.

Examples of evaluation of economic characteristics

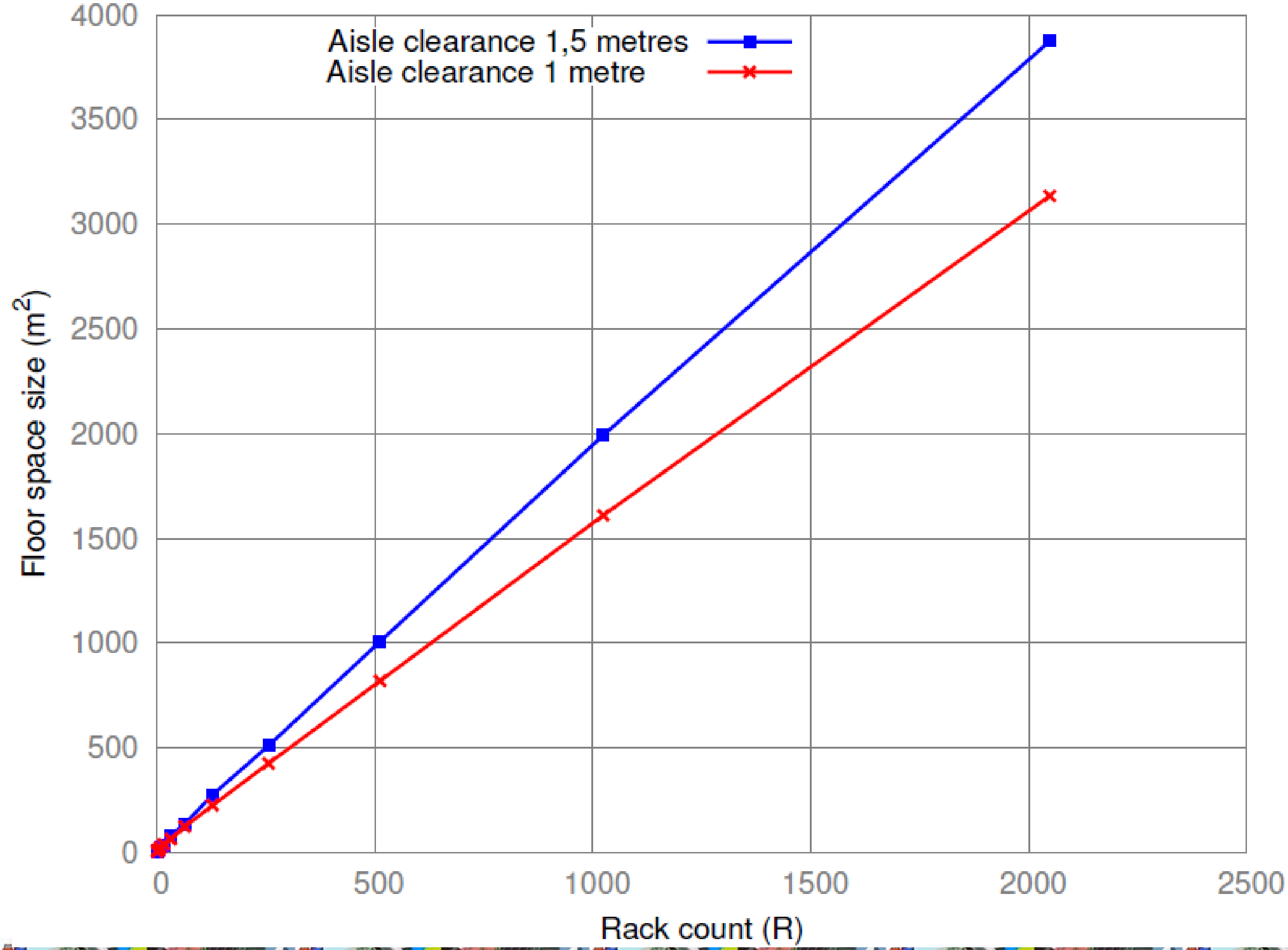
Network cost comparison



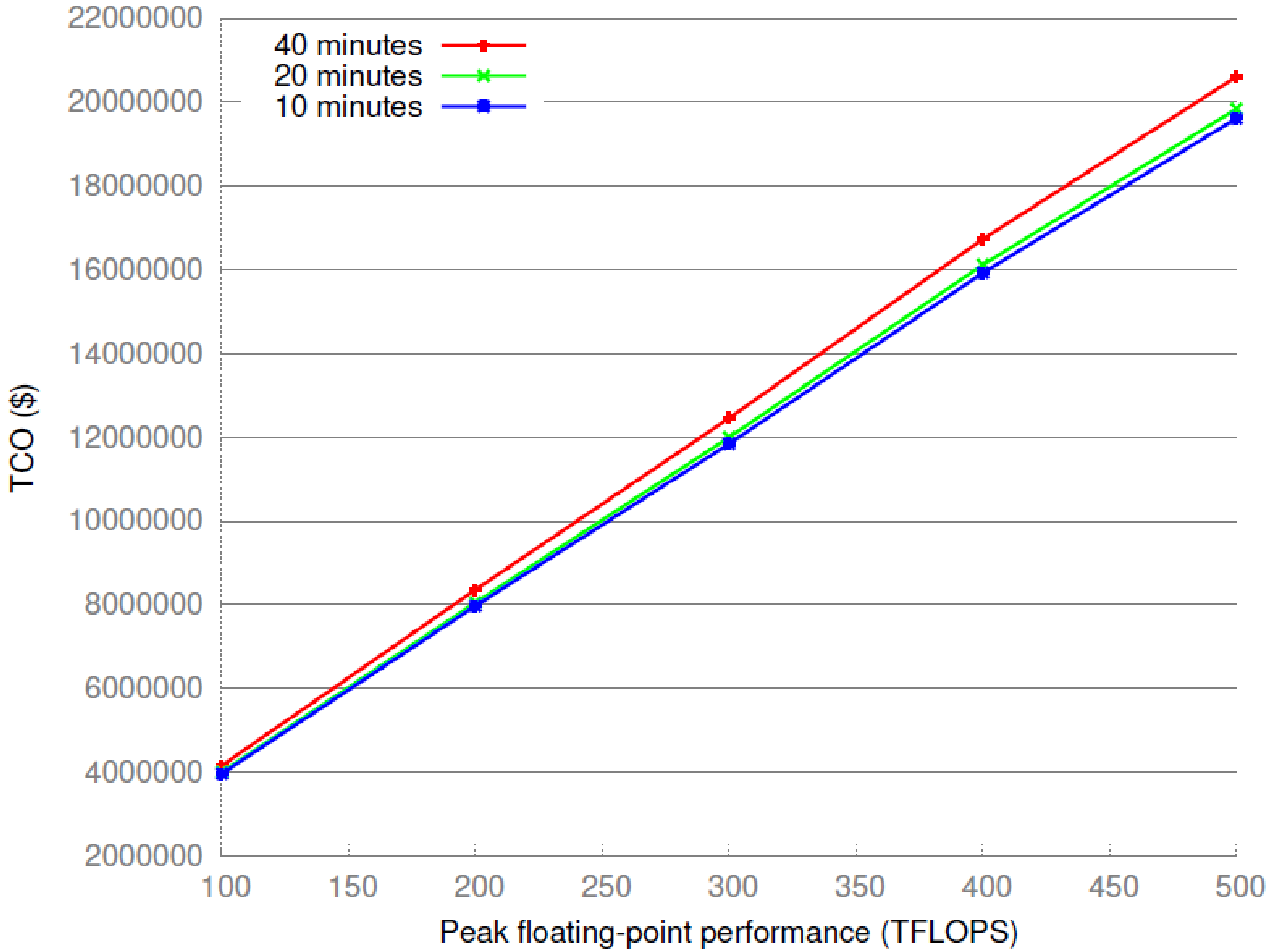
Network cost comparison



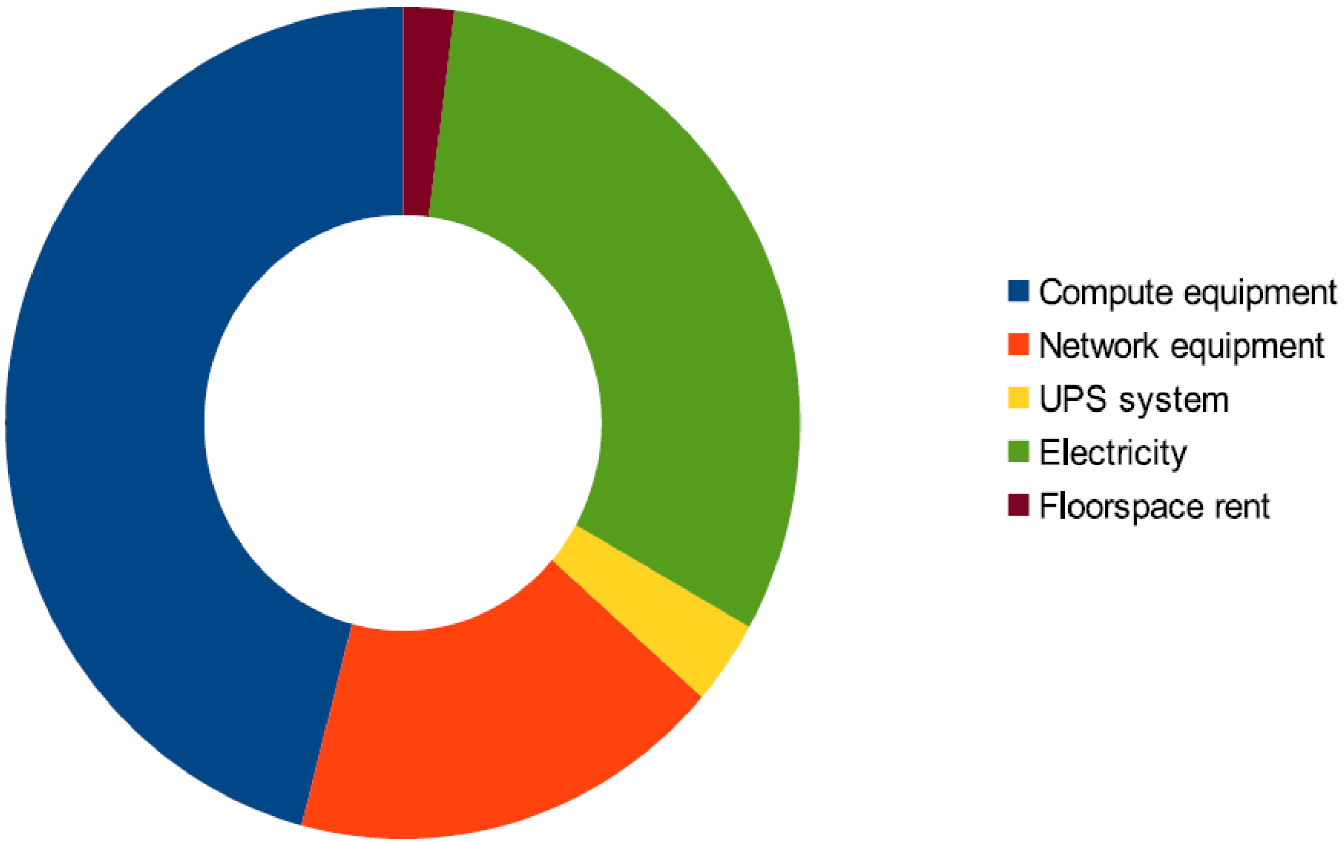
Floor size comparison



UPS cost comparison



TCO breakdown pie chart



Scientific Contribution

- CAD systems
  - the method for representing compatibility between components of arbitrary technical systems using directed acyclic multipartite graphs
- Performance modelling
  - the notion of inverse performance models
  - the two-phase iterative algorithm for inverse performance modelling

Scientific Contribution

- Computer networks
  - algorithms to design two-level fat-tree and torus networks, with arbitrary blocking factors
- Datacentre design
  - strategies and heuristics for placing equipment into racks, for the general case of non-identical equipment blocks
  - the algorithm for calculating floor space size required for the given number of racks

Scientific Contribution

- Cooling systems
  - a decision chart for choosing air preparation methods for cooling with outside air
  - an algorithm for calculating cooling capacity for cooling with outside air

Scientific Contribution

- Economics
  - a comparison of factors that influence cost and performance of cluster supercomputers
  - a quantitative analysis of using low-power ("green") memory modules
  - an overview of TCO components for supercomputers
  - a proposal to reuse waste heat from data centres for large-scale greenhouses, together with an implementation plan

"To be of use to the world is the only way to be happy"

Hans Christian Andersen

konstantin@solnushkin.org

https://www.linkedin.com/in/solnushkin

- Learn more and get the software:

  http://ClusterDesign.org/saddle