Of Apples and Oranges

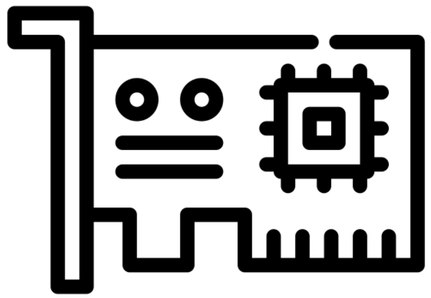
Fair Comparisons in Heterogenous Systems Evaluation

Hugo Sadok, Aurojit Panda, Justine Sherry

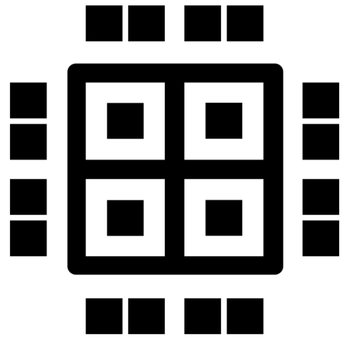
Carnegie Mellon University

NYU

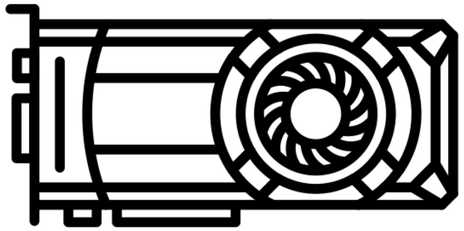
Systems Research Increasingly Uses Accelerators



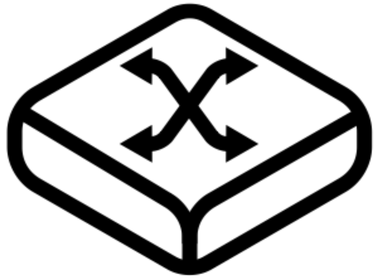
SmartNICs



FPGAs



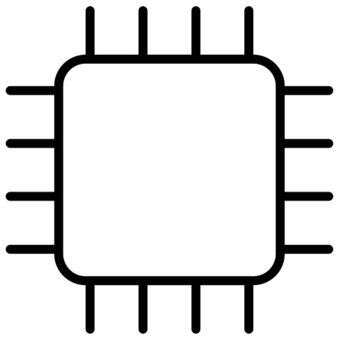
GPUs



Programmable Switches

Common story in networking and systems literature…

CPU



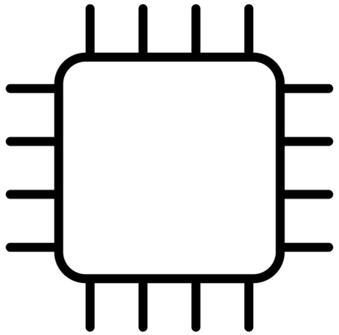
Original System

e.g., Firewall

Original system runs on a traditional CPU

Common story in networking and systems literature...

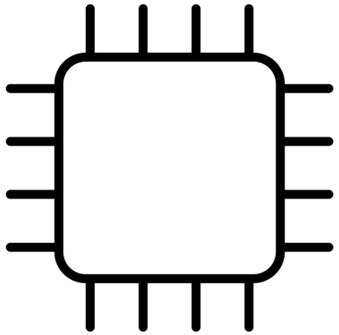
CPU



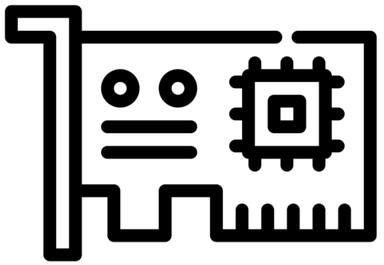
Original System

e.g., Firewall

CPU



\+



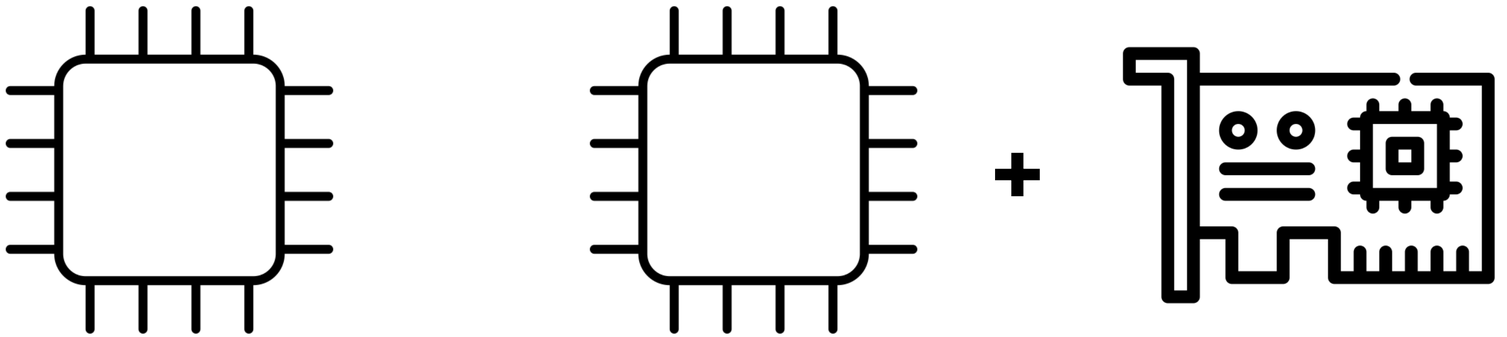
Accelerator

Proposed System

e.g., Firewall with SmartNIC

Proposed system uses an accelerator in addition to the CPU

Common story in networking and systems literature...

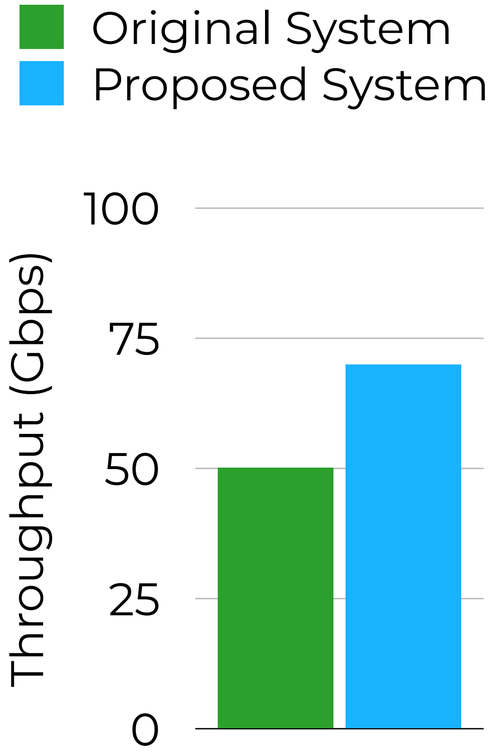


Original System

e.g., Firewall

Proposed System

e.g., Firewall with SmartNIC

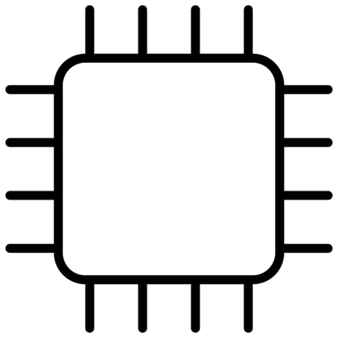


Proposed system performs better than the original system

Is the proposed system really better than the original system?

What if we scale the original system by giving it more CPU cores?

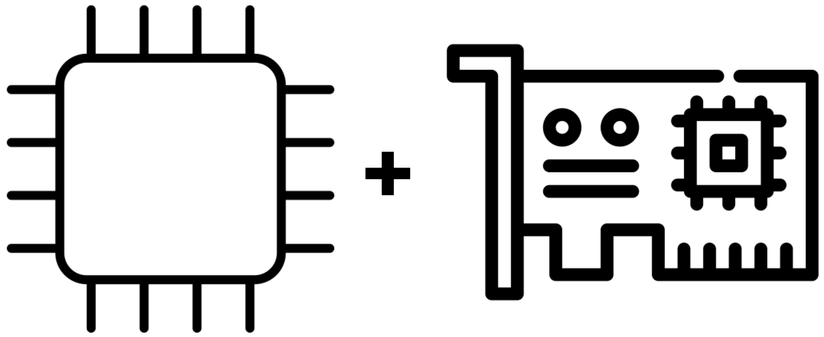
CPU



Original System

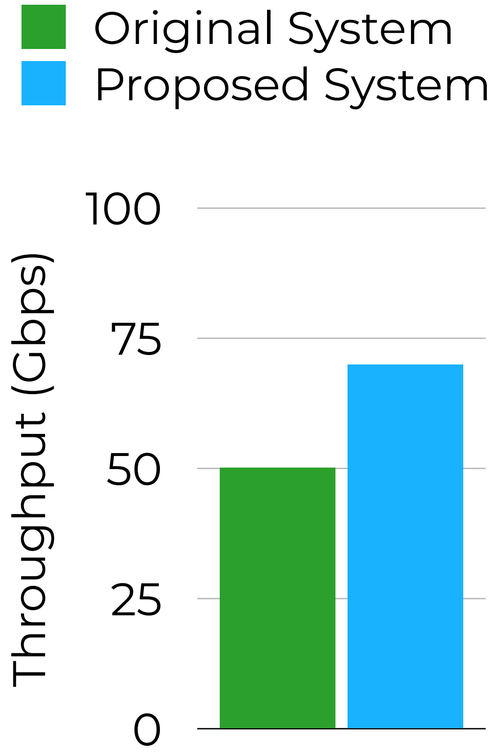
e.g., Firewall

CPU **Accelerator**

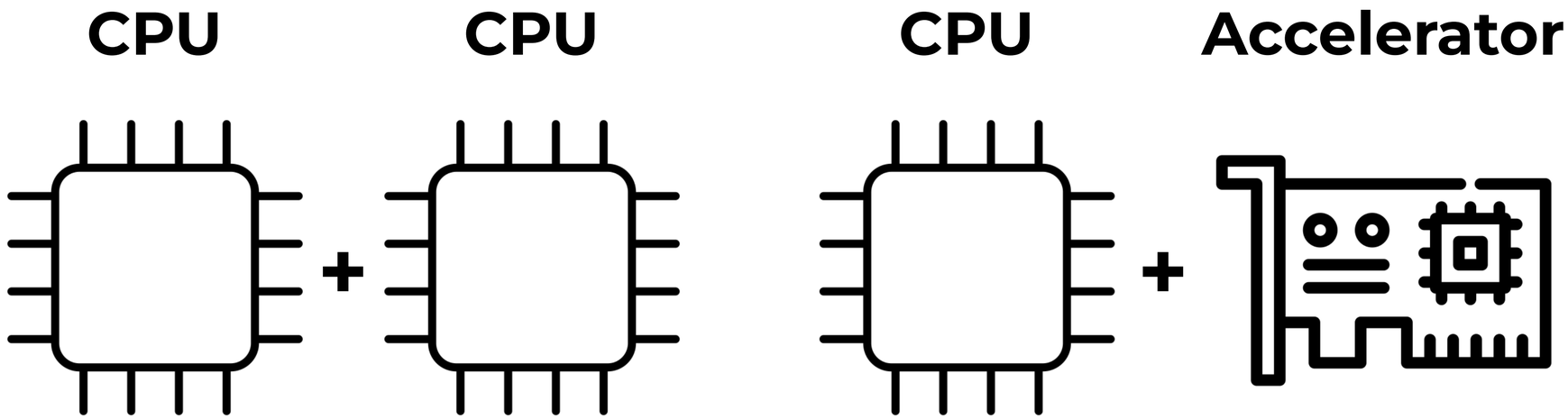


Proposed System

e.g., Firewall with SmartNIC



What if we scale the original system by giving it more CPU cores?

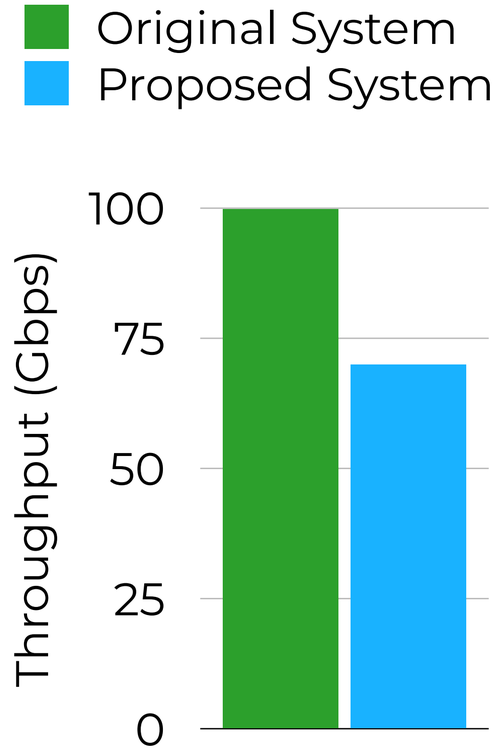


Original System

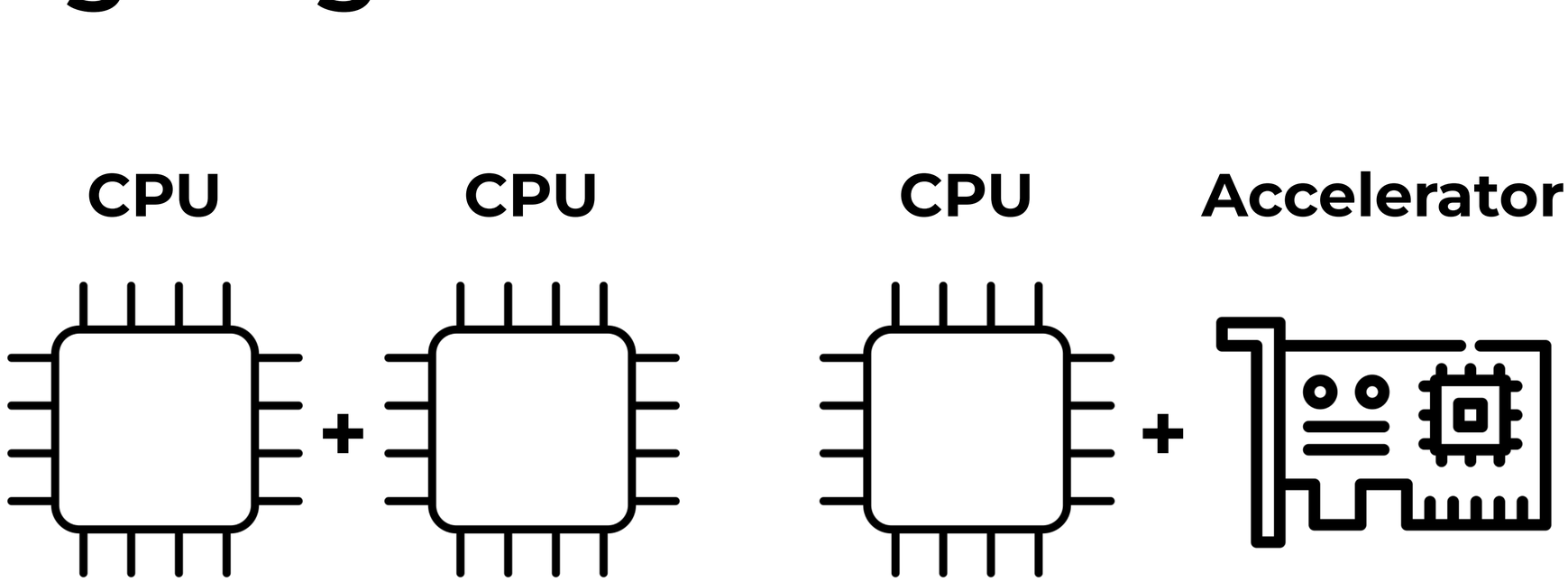
e.g., Firewall

Proposed System

e.g., Firewall with SmartNIC



What if we scale the original system by giving it more CPU cores?

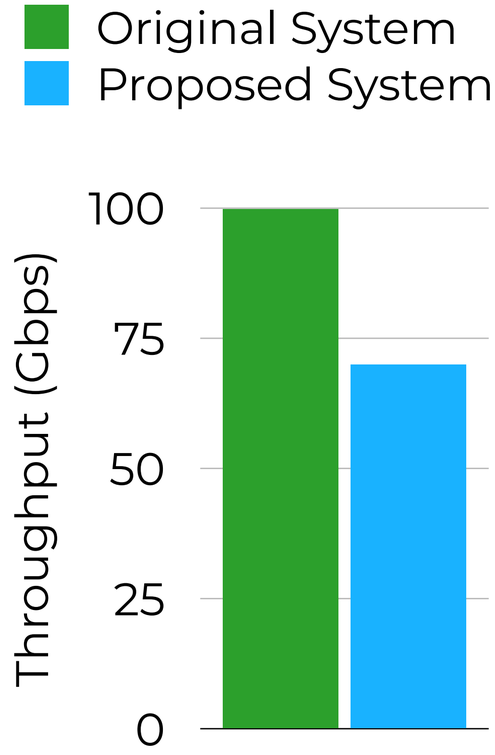


Original System

e.g., Firewall

Proposed System

e.g., Firewall with SmartNIC



Why is an accelerator any better than these additional cores?

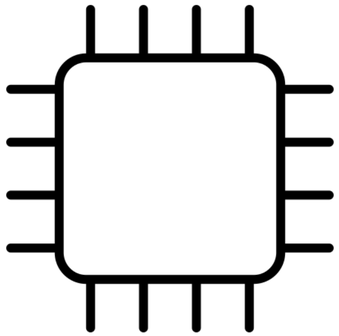
Our main message

Systems evaluation with heterogeneous hardware should consider, and report, not only performance but also cost

Traditional systems evaluation is unidimensional and can ignore cost

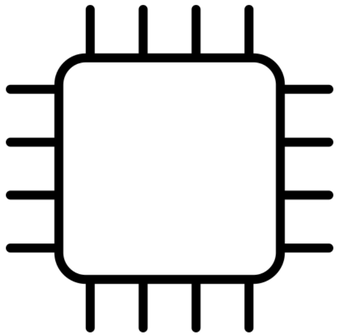
Traditional systems evaluation is unidimensional and can ignore cost

CPU



Original System

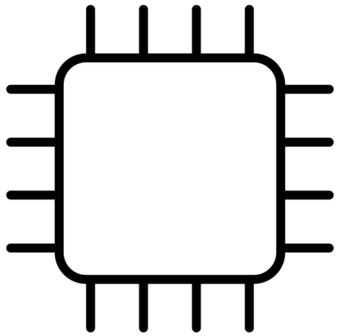
CPU



Proposed System

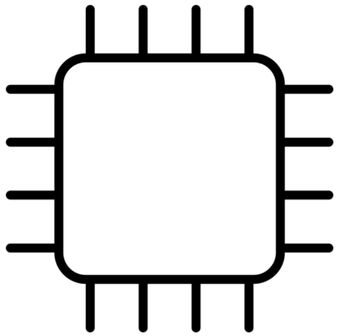
Traditional systems evaluation is unidimensional and can ignore cost

CPU



Original System

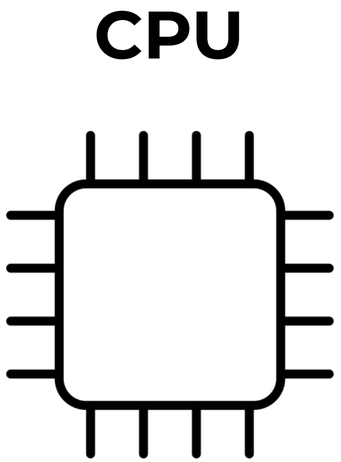
CPU



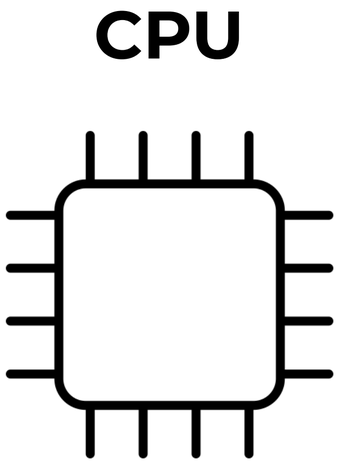
Proposed System

Both systems use the same hardware

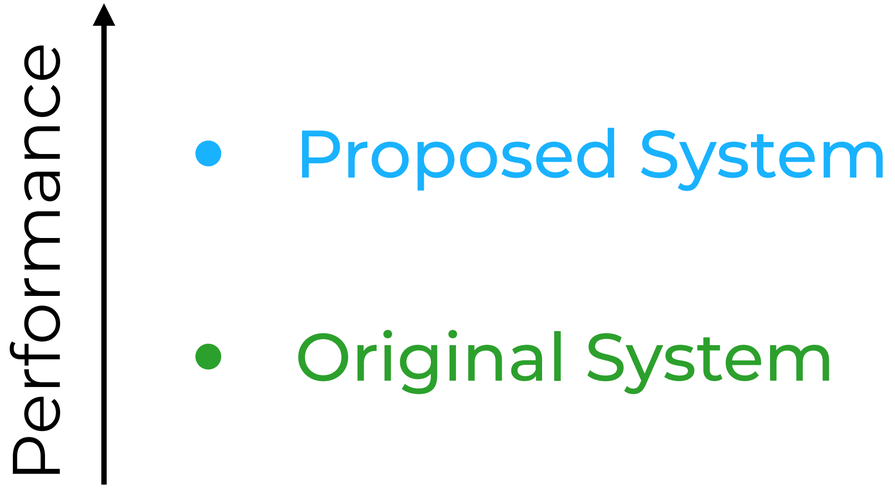
Traditional systems evaluation is unidimensional and can ignore cost



Original System

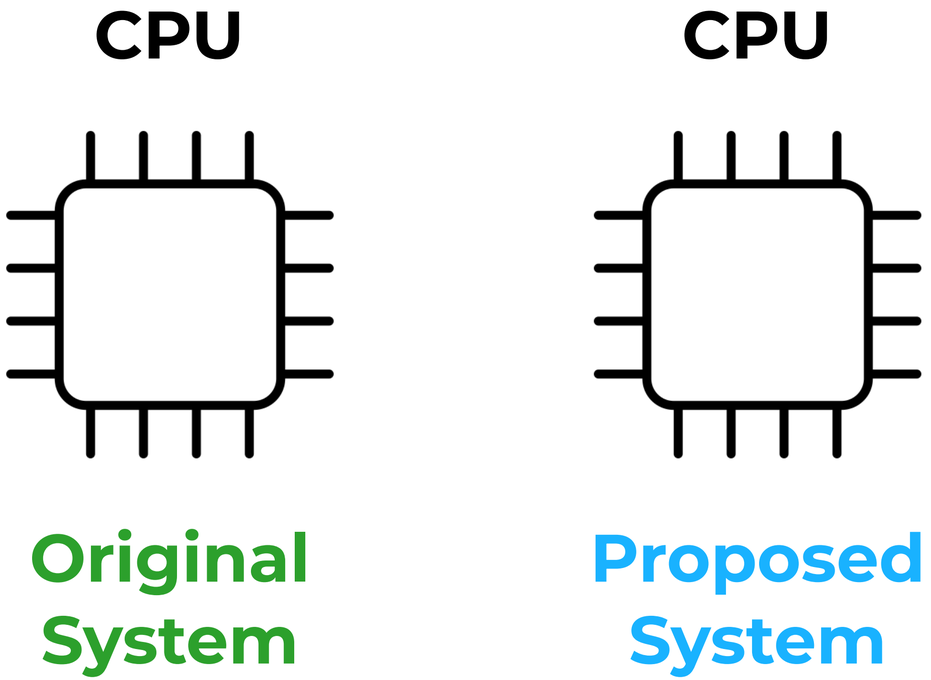


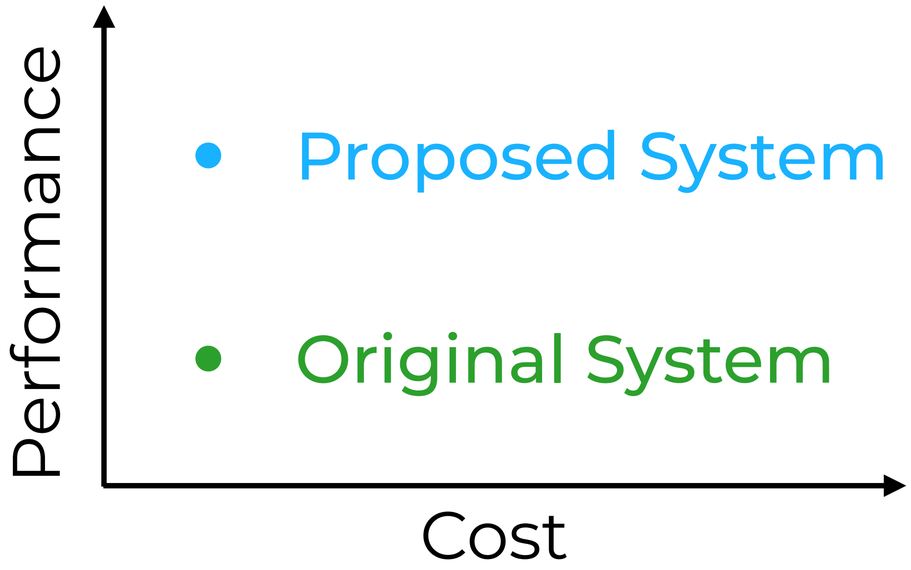
Proposed System



Both systems use the same hardware

Traditional systems evaluation is unidimensional and can ignore cost

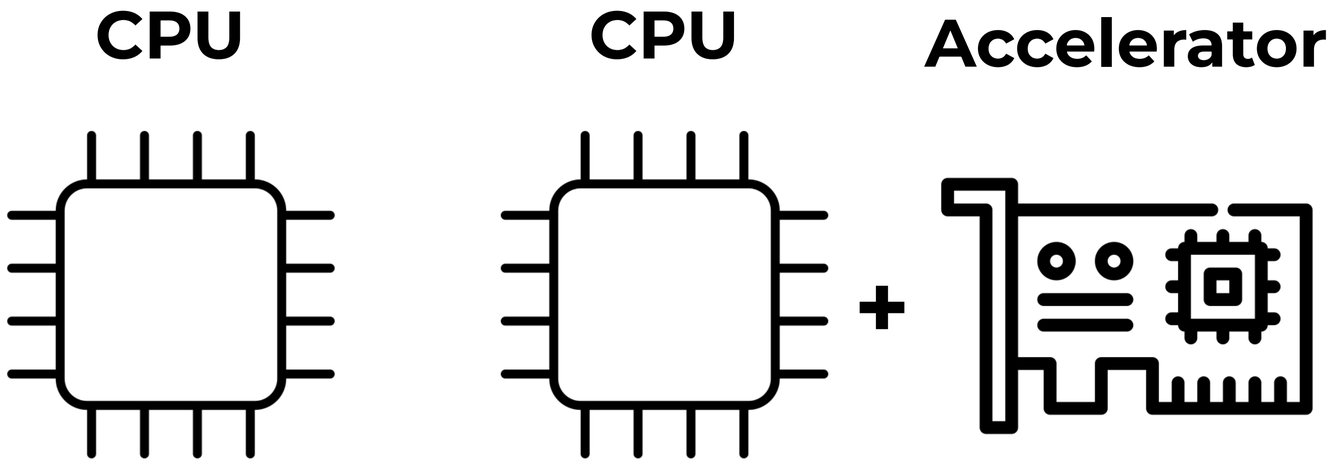


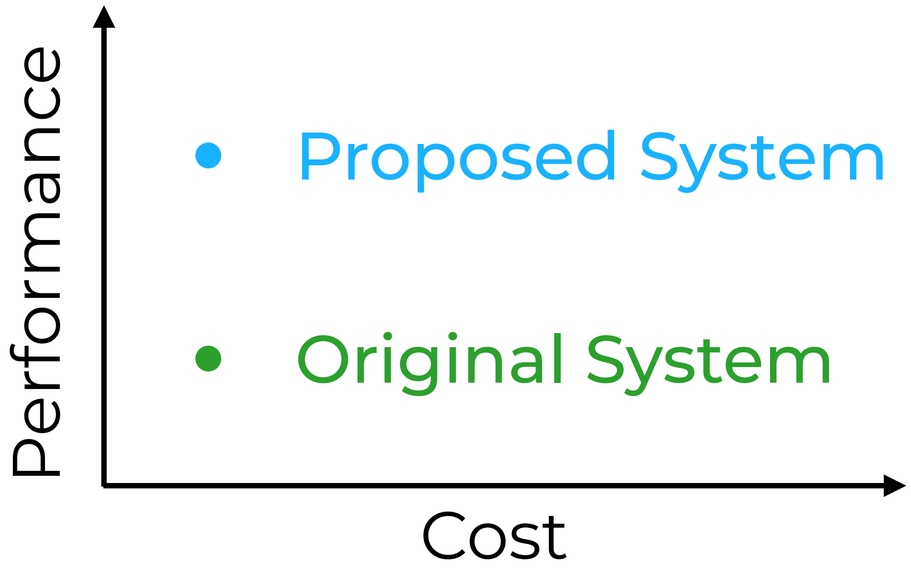


Both systems use the same hardware

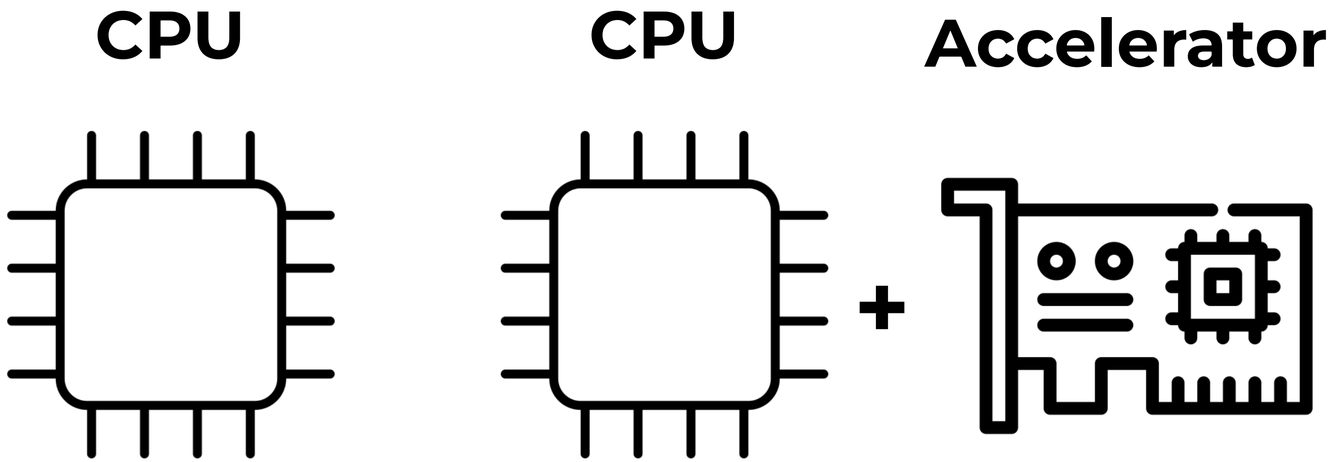
and share the same cost

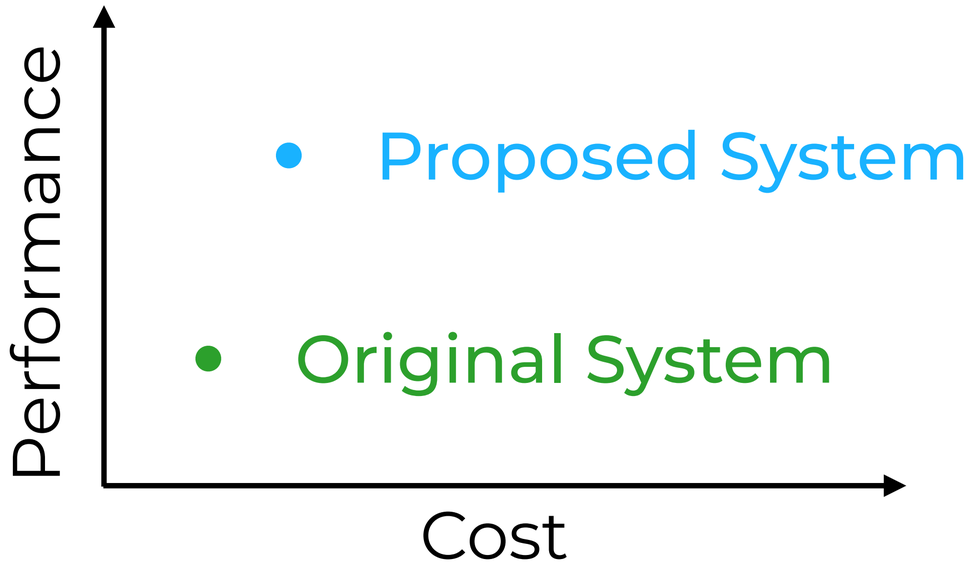
But systems with different hardware have different costs



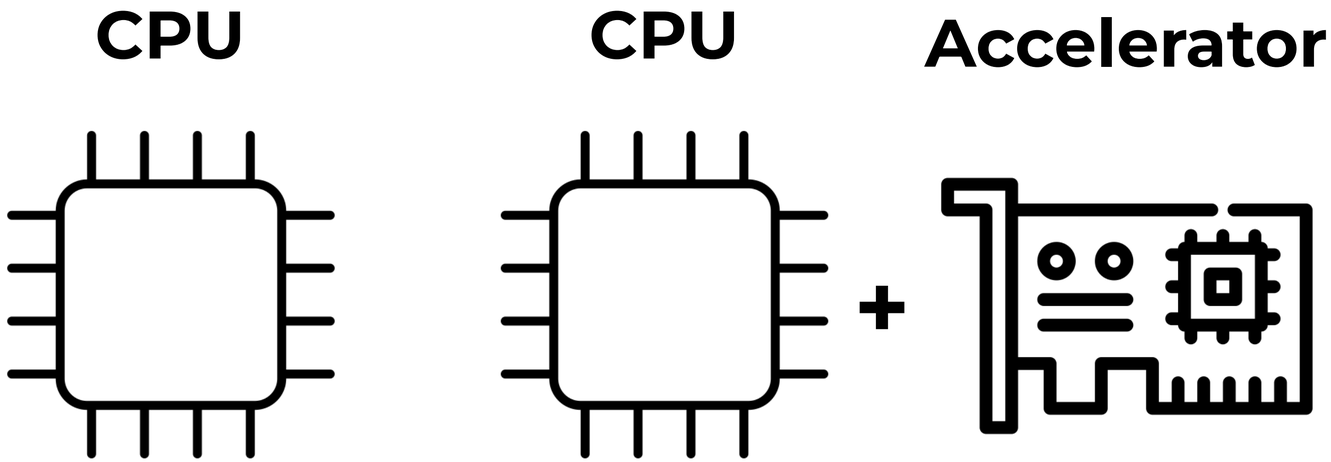


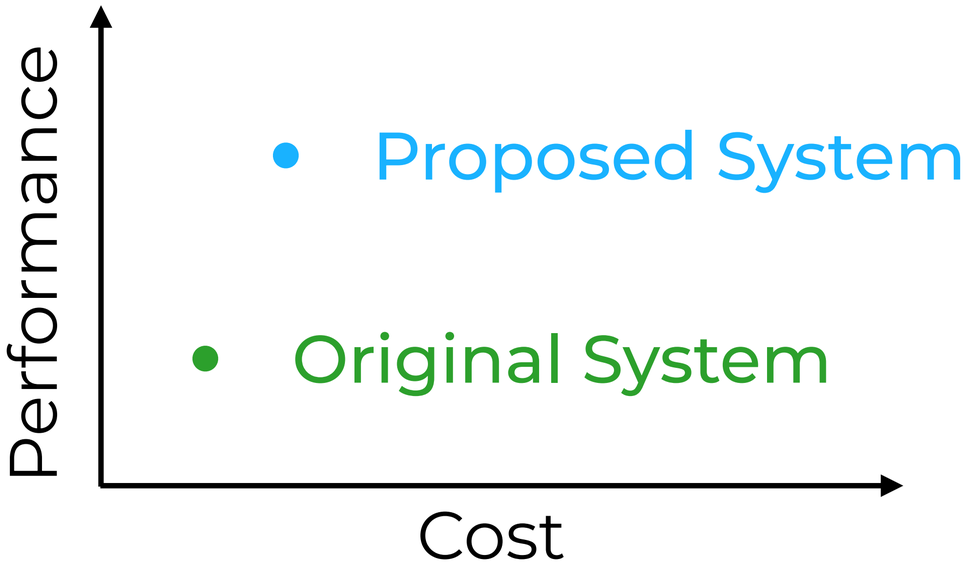
But systems with different hardware have different costs





But systems with different hardware have different costs





The evaluation can no longer be made unidimensional

① How to pick good cost metrics?

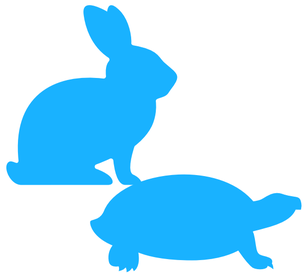
① How to pick good cost metrics?

② How to incorporate cost when evaluating systems with heterogeneous hardware?

① How to pick good cost metrics?

② How to incorporate cost when evaluating systems with heterogeneous hardware?

Principles for Choosing Cost Metrics



Context Independent

Quantifiable

End-to-end coverage

Costs should be Context-Independent

Costs should be Context-Independent

Problem: Many cost metrics are context-dependent

Costs should be Context-Independent

Problem: Many cost metrics are context-dependent

Their values depend on **when** they are measured and by **whom**

Costs should be Context-Independent

Problem: Many cost metrics are context-dependent

Their values depend on **when** they are measured and by **whom**

Context Dependent

Hardware Price ($)

Total Cost of Ownership ($)

Carbon Footprint (CO_2e)

Costs should be Context-Independent

Problem: Many cost metrics are context-dependent

Their values depend on **when** they are measured and by **whom**

Context Dependent	Context Independent
Hardware Price ($)	Power (W)
Total Cost of Ownership ($)	Silicon Die Area (mm^2)
Carbon Footprint (CO_2e)	Heat dissipation (BTU/h)

Costs should be Quantifiable

Costs should be Quantifiable

Problem: Many useful costs are not quantifiable

Costs should be Quantifiable

Problem: Many useful costs are not quantifiable

We have no good tools or standard procedures for measuring them

Costs should be Quantifiable

Problem: Many useful costs are not quantifiable

We have no good tools or standard procedures for measuring them

Unquantifiable

Programming Complexity

Manageability

Costs should be Quantifiable

Problem: Many useful costs are not quantifiable

We have no good tools or standard procedures for measuring them

Unquantifiable	Quantifiable
Programming Complexity	Power (W)
Manageability	Hardware Price ($)

Costs should have End-to-End Coverage

Costs should have End-to-End Coverage

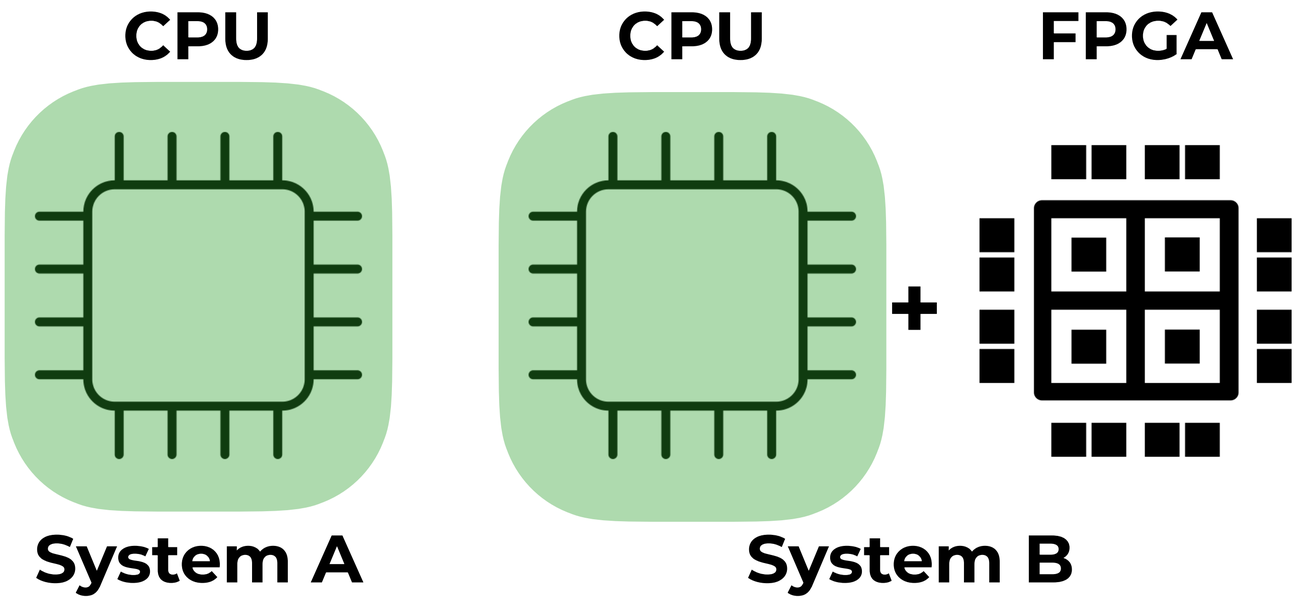
Problem: Some cost metrics fail to cover all the hardware components

Costs should have End-to-End Coverage

Problem: Some cost metrics fail to cover all the hardware components

Cost Metric

Number of CPU cores

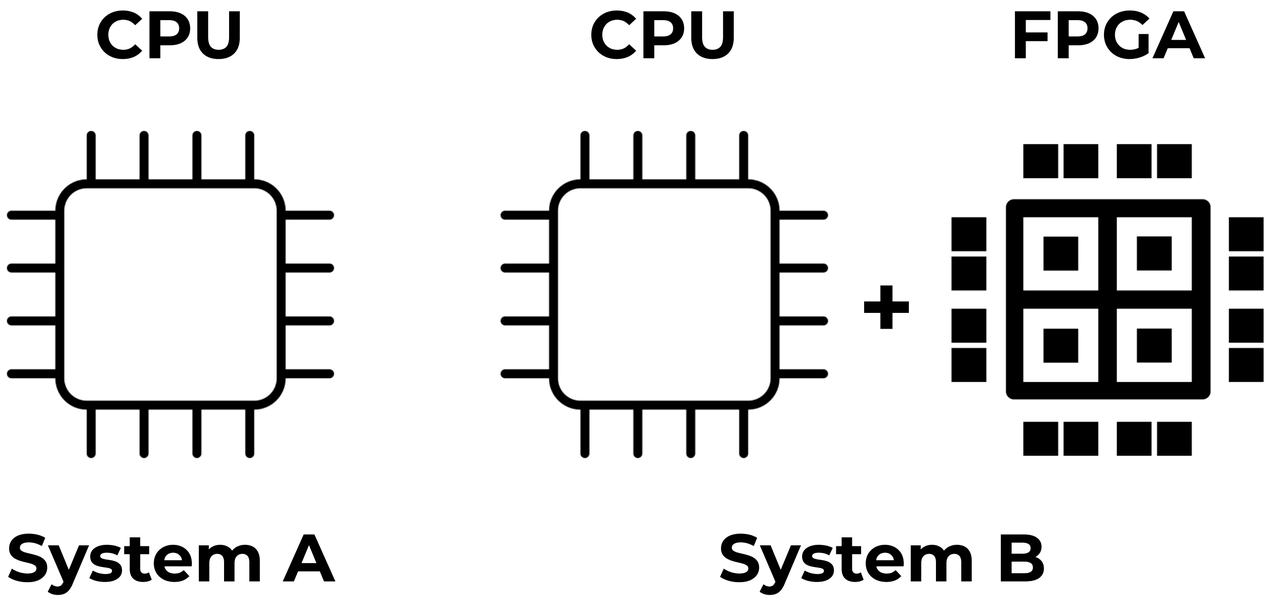


Costs should have End-to-End Coverage

Problem: Some cost metrics fail to cover all the hardware components

Cost Metric

~~Number of CPU cores~~



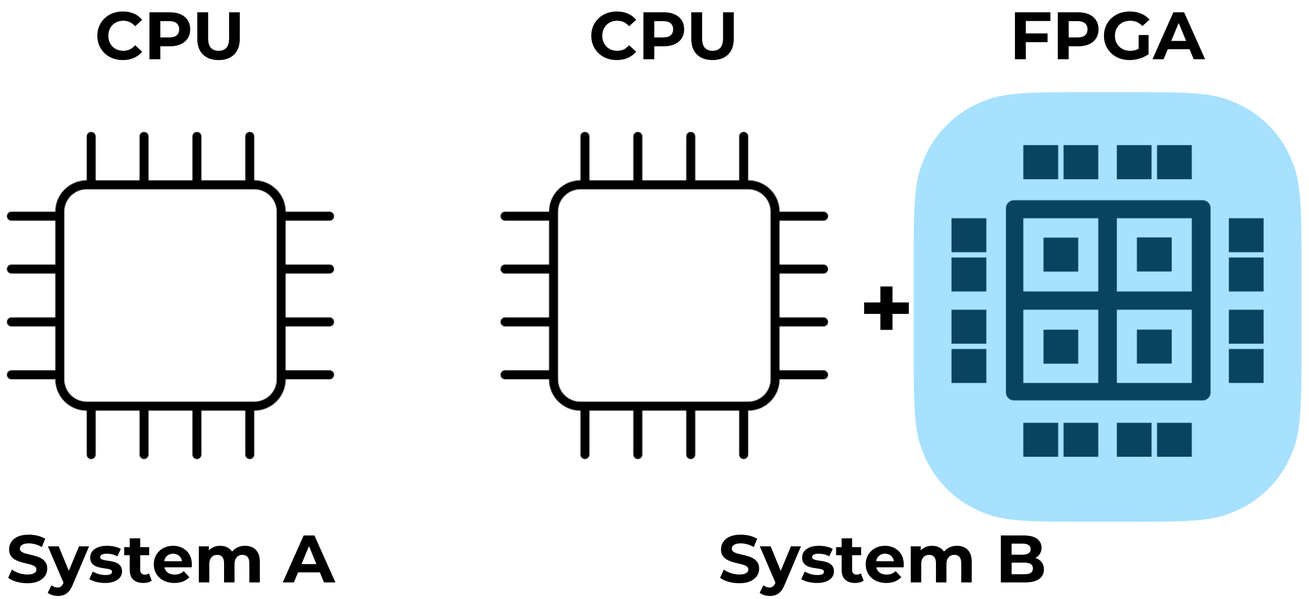
Costs should have End-to-End Coverage

Problem: Some cost metrics fail to cover all the hardware components

Cost Metric

~~Number of CPU cores~~

Number of Lookup Tables



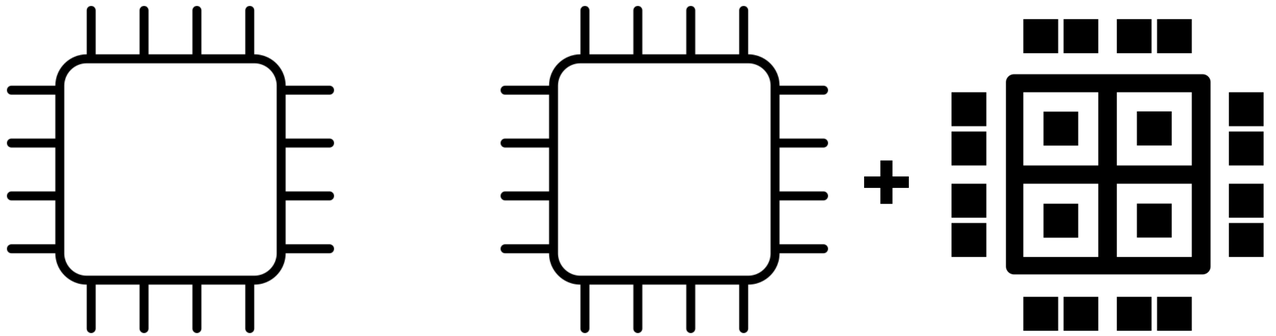
Costs should have End-to-End Coverage

Problem: Some cost metrics fail to cover all the hardware components

Cost Metric

~~Number of CPU cores~~

~~Number of Lookup Tables~~



System A **System B**

Costs should have End-to-End Coverage

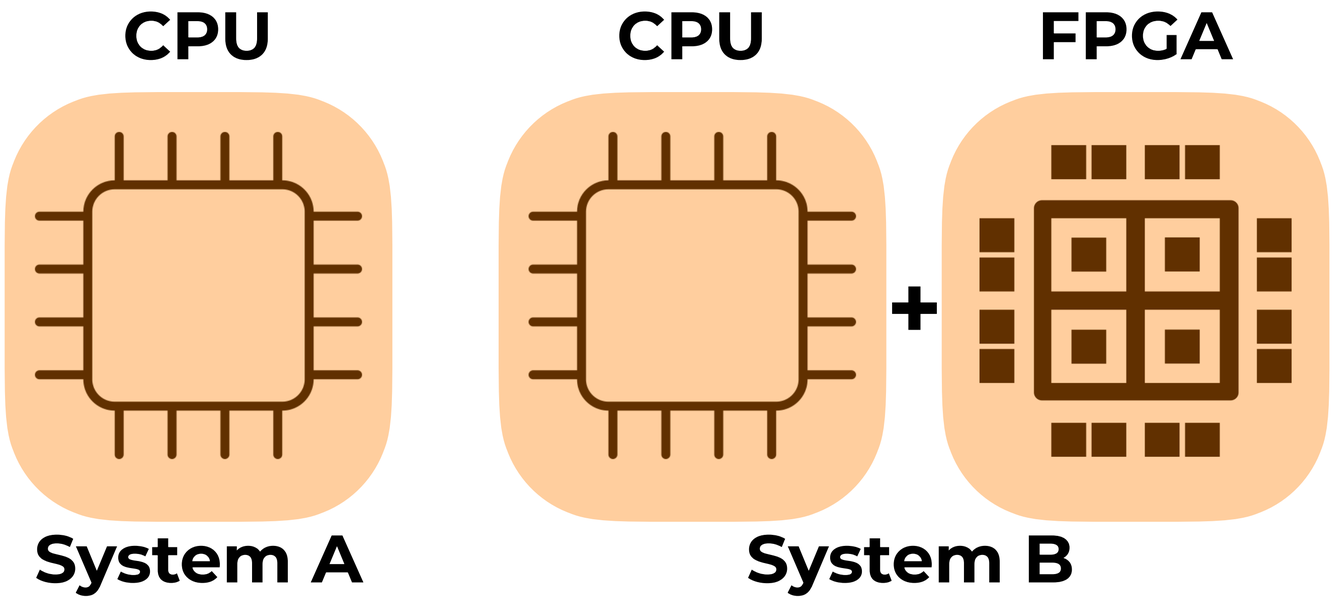
Problem: Some cost metrics fail to cover all the hardware components

Cost Metric

~~Number of CPU cores~~

~~Number of Lookup Tables~~

Power



Costs should have End-to-End Coverage

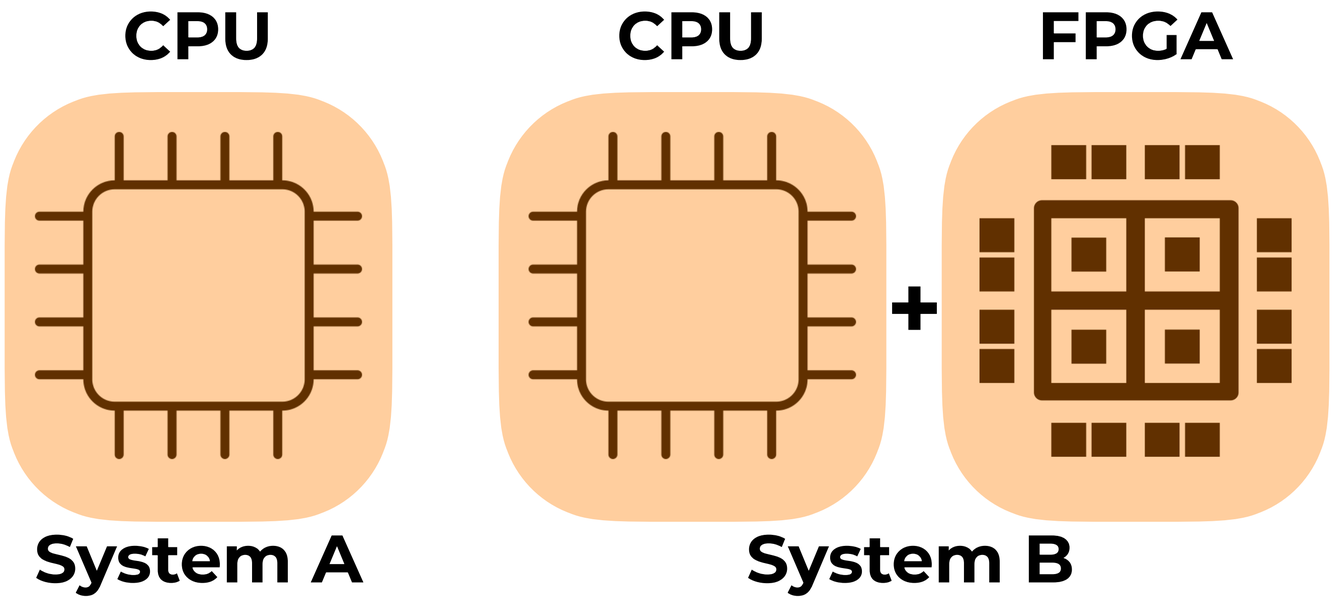
Problem: Some cost metrics fail to cover all the hardware components

Cost Metric

~~Number of CPU cores~~

~~Number of Lookup Tables~~

Power ✓



Costs should have End-to-End Coverage

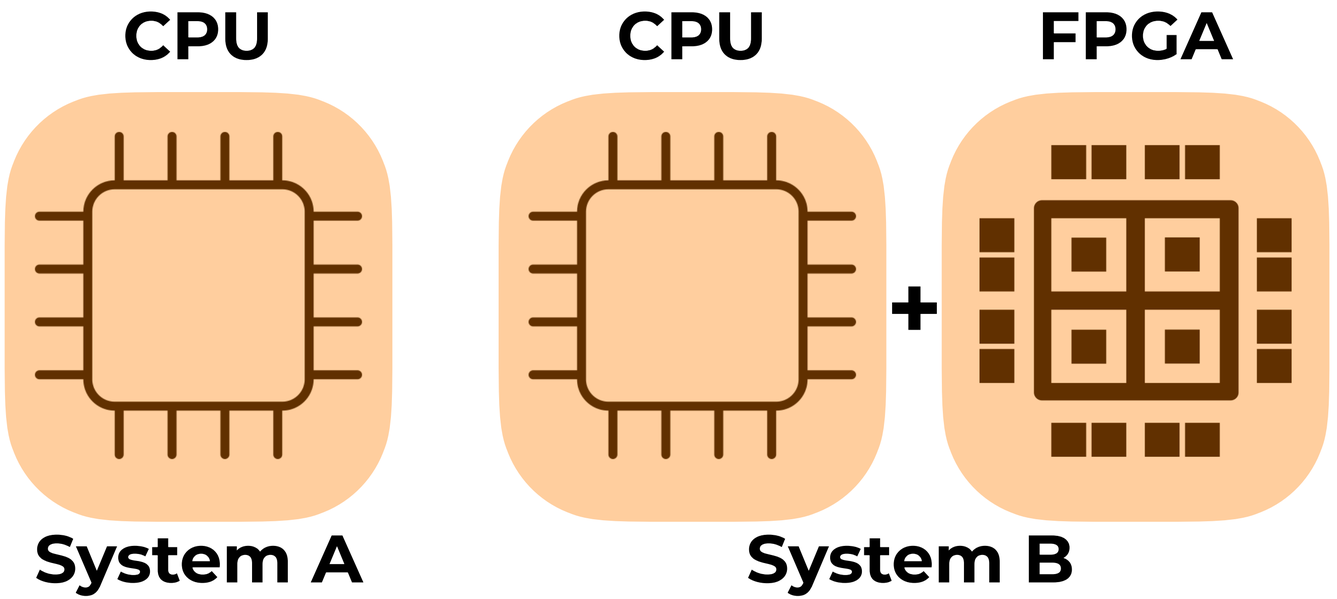
Problem: Some cost metrics fail to cover all the hardware components

Cost Metric

~~Number of CPU cores~~

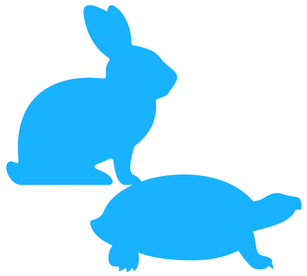
~~Number of Lookup Tables~~

Power ✓



Power is a good cost metric as it covers all the hardware components for all systems being evaluated

Principles for Choosing Cost Metrics



Context Independent

Quantifiable

End-to-end coverage

Conclusion and Discussion

Conclusion and Discussion

Evaluations with heterogeneous hardware must be **multidimensional**, considering **both performance and cost**

Conclusion and Discussion

Evaluations with heterogeneous hardware must be **multidimensional**, considering **both performance and cost**

Cost metrics should be: **context independent**, **quantifiable**, and have **end-to-end coverage**

Conclusion and Discussion

Evaluations with heterogeneous hardware must be **multidimensional**, considering **both performance and cost**

Cost metrics should be: **context independent**, **quantifiable**, and have **end-to-end coverage**

Potential Practical Issues

Conclusion and Discussion

Evaluations with heterogeneous hardware must be **multidimensional**, considering **both performance and cost**

Cost metrics should be: **context independent**, **quantifiable**, and have **end-to-end coverage**

Potential Practical Issues

Cost information is often sensitive: companies may not be willing to reveal it

Conclusion and Discussion

Evaluations with heterogeneous hardware must be **multidimensional**, considering **both performance and cost**

Cost metrics should be: **context independent**, **quantifiable**, and have **end-to-end coverage**

Potential Practical Issues

Cost information is often sensitive: companies may not be willing to reveal it

TCO is the main cost metric used in industry so why are we rejecting it? Are there context-independent alternatives?