# Title: Auto-generation of Computer System Configurations Based on Input Constraints

**Inventors:** Zehra Sura/Watson/IBM, Anthony Hylick/Austin/IBM

# === Executive Summary ===

We disclose a tool that is formatted with: (1) a list of system building blocks (such as accelerator devices and interconnect types), and (2) interface rules for each system building block (that define valid and invalid connections between blocks). Each system building block has an associated set of metrics (such as price, performance, power requirements, etc). Then, for a given set of input constraints (such as maximum price or energy consumption, or minimum application needs/functionality), the tool automatically generates a set of computer system configurations that satisfy the input constraints. By appropriately specifying input constraints, the tool can be used by computer system designers to systematically explore the interesting portion of the system design space, or by customers to help determine the right system to buy for their specific needs, or by users deploying their applications in a cloud environment.

# === Deta**ils ===**

## -- Background --

Computer systems can be built by connecting various devices (such as devices used for storage, memory, and/or compute) using one or more interconnect types (such as PCIe, Ethernet, and other network switches), and various interconnect topologies.

Commodity off-the-shelf components for computing systems are built according to interfaces standardized in the industry, or include a clear definition of their interface properties, and can be used as building blocks when constructing a computing system.

There are metrics associated with individual components and the overall system, such as price, performance, energy consumption, etc.

Given some constraints on metrics or a prioritization of metrics, it is difficult to determine what computing system configuration is best suited for a given application scenario. The current trend towards using heterogeneous computing devices makes this problem even more difficult.

## -- Our Invention --

Our invention is a tool that automatically generates a set of computer system configurations based on input constraints. It uses a graphical representation to denote a computing system configuration.

The tool itself is initialized with the following information (which can be updated dynamically):

1. A list of system building blocks: each block has zero or more associated categories.
2. Interface rules for each system building block.

The tool supports a set of metrics such as price, power consumption, peak performance, memory capacity, etc. Each system building block or category of system building blocks has associated values defined for any number of these metrics. Interface rules allow the tool to distinguish valid and invalid connections between two or more building blocks. The rules can be defined using categories of system building blocks, types of individual system building blocks, number of system building blocks, and/or values of any associated metrics. A system building block inherits all properties/rules associated with its categories.

The tool takes as input a set of constraints that are formulated according to the application scenario. Examples of constraints include a maximum price or energy consumption value for the whole system, a minimum bandwidth between all devices in a certain category, a minimum/maximum amount of total storage capacity, etc. At least one of the input constraints is required to be a “bounding constraint”, i.e. a constraint that places an upper bound on the cardinality of the set of output computing system configurations.

The tool can optionally include a set of “internal” constraints to further limit the number of computing system configurations in the set output by the tool. Examples of such internal constraints are a maximum cardinality for the output set, or constraints defined on the graphical representation of the computing system configuration that reflect experiential knowledge of computing system design.

The tool uses known techniques from combinatorics, graph enumeration and graph isomorphism to map the given constraints to a set of computing system configurations, that is then produced as output.

Device properties are used to inform graph isomorphism.

# === Points of Novelty ===

1. A method

# === Related and Enabling Art ===

1. The Open Hardware