Constraint Systems

Global Constraints in Or-Tools

Some Global Constraints in Or-Tools

Let's see the signature for some global constraints in or-tools:

`X` refers always to a list of variables (or expressions)

- ALLDIFFERENT(X) $\longleftrightarrow$ `slv.AllDifferent(X)`
- ALLDIFFERENTEXCEPT(X) $\longleftrightarrow$ `slv.AllDifferentExcept(X, v)`
  - `v` is the escape value
- DISTRIBUTE(X, V, C) $\longleftrightarrow$ `slv.Distribute(X, V, C)`
  - `V` is the list of the values to constrain/count
  - `C` is a list with the cardinality variables for the values in `V`
- COUNT(X, V, C) $\longleftrightarrow$ `slv.Count(X, v, c)`
  - `v` is the value to count, `c` is either a cardinality variable or a value
- ATMOST(X) $\longleftrightarrow$ `slv.AtMost(X, v, u)`
  - `v` is the value to limit, `u` the maximum number of occurrences

Some Global Constraints in Or-Tools

Let's see the signature for some global constraints in or-tools:

`X` refers always to a list of variables (or expressions)

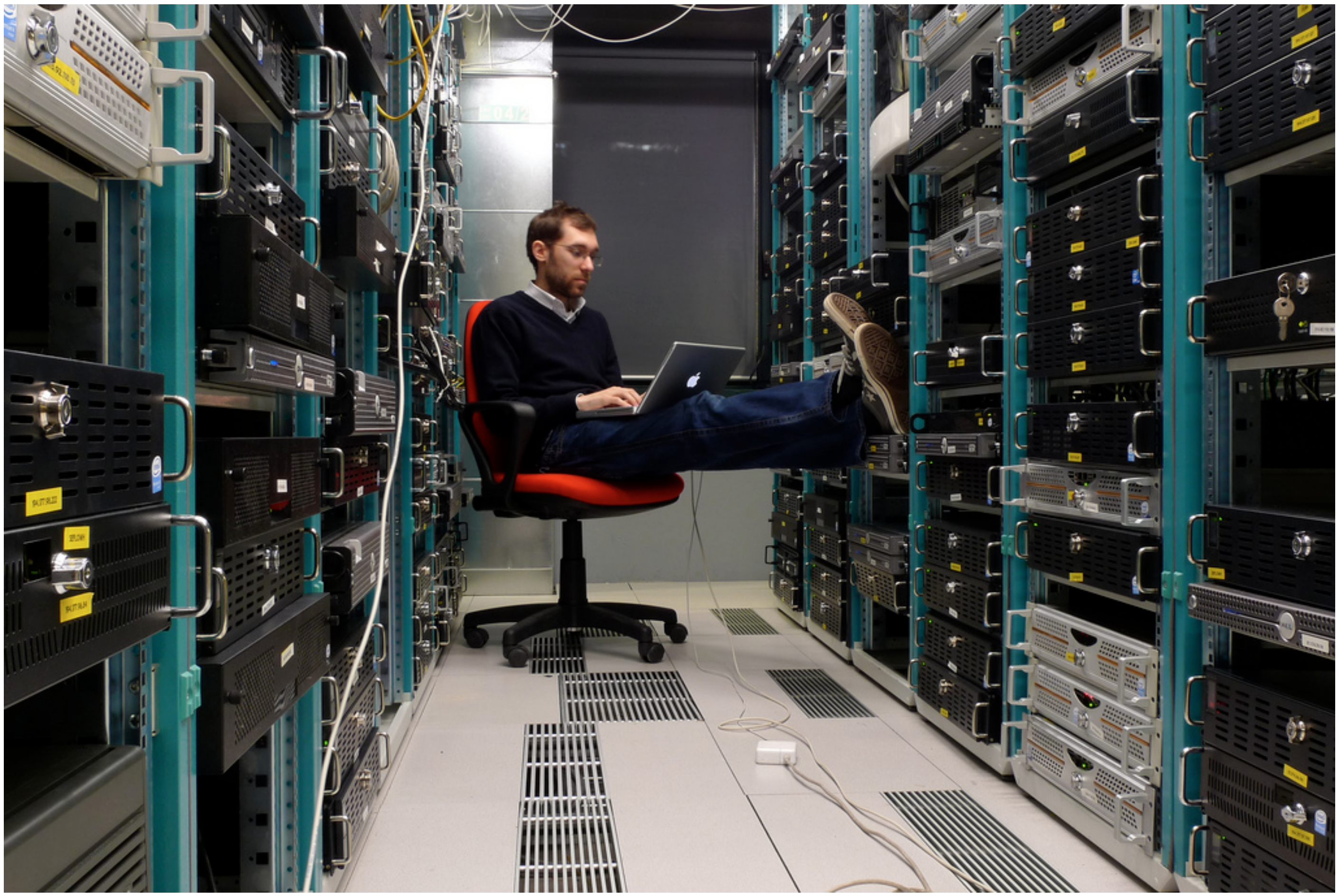
- $\mathrm{MIN}(z, X) \longleftrightarrow$ `slv.Min(X)`
  - `z` is implicit (`slv.Min` returns an expression object)
- $\mathrm{MAX}(z, X) \longleftrightarrow$ `slv.Max(X)`
  - `z` is implicit (same as above)
- $\mathrm{SUM}(z, X) \longleftrightarrow$ `slv.Sum(X)`
  - `z` is implicit (same as above)
- $\mathrm{ELEMENT}(z, V, x) \longleftrightarrow$ `var.IndexOf(V)`
  - `var` corresponds to x (i.e. the index)
  - `V` is the list to be indexed
  - `z` is implicit (same as above)

Constraint Systems

Lab 5 - Improving a Model

Server Consolidation

A data center hosts servers running a number of Virtual Machines



Server Consolidation

A data center hosts servers running a number of Virtual Machines

About the servers:

- There is a finite number of servers n_s
- Each server has a finite number of cores n_c
- All servers are identical

About the Virtual Machines:

- Each Virtual Machine i can run on a single server
- Each VM requires exclusive access to a number of cores c_i
- The VMs are grouped in services
- VMs within the same service should run on different servers

Server Consolidation

A data center hosts servers running a number of Virtual Machines

Goal:

- Pack the VMs on the smallest possible number of servers
  - This is called server consolidation
  - It's a common technique to reduce power consumption
- Build a CP model for this server consolidation problem
  - The start-kit contains instances + a basic working model
  - Try to improve the model as much as possible!
  - Use global constraints, break symmetries…
  - …Add bounds and redundant constraints
  - Only one rule: do not change the search strategy

It is possible to solve all instances in less than 30 seconds

Server Consolidation

The provided model is given by:

$$\min\ z = \sum_{j=0}^{n_s-1} (u_j > 0)$$

$$\begin{aligned}
\text{subject to: } u_j &= \sum_{i=0}^{n_{vm}-1} r_i(x_i = j) && \forall j = 0..n_s - 1 \\
u_j &\leq n_c && \forall j = 0..n_s - 1 \\
x_i &\neq x_j && \forall i, j = 0..n_{vm} - 1 \ :\ i < j, s_i = s_j \\
x_i &\in \{0..n_s - 1\} && \forall i = 0..n_{vm} - 1 \\
u_j &\in \{0..n_c\} && \forall j = 0..n_s - 1
\end{aligned}$$

- n_{vm} is the number of VMs, $x_i = j$ iff VM i is assigned to server j
- u_j is a variable/expression representing the core usage on server j
- s_i is the service for VM i, r_i the required number of cores