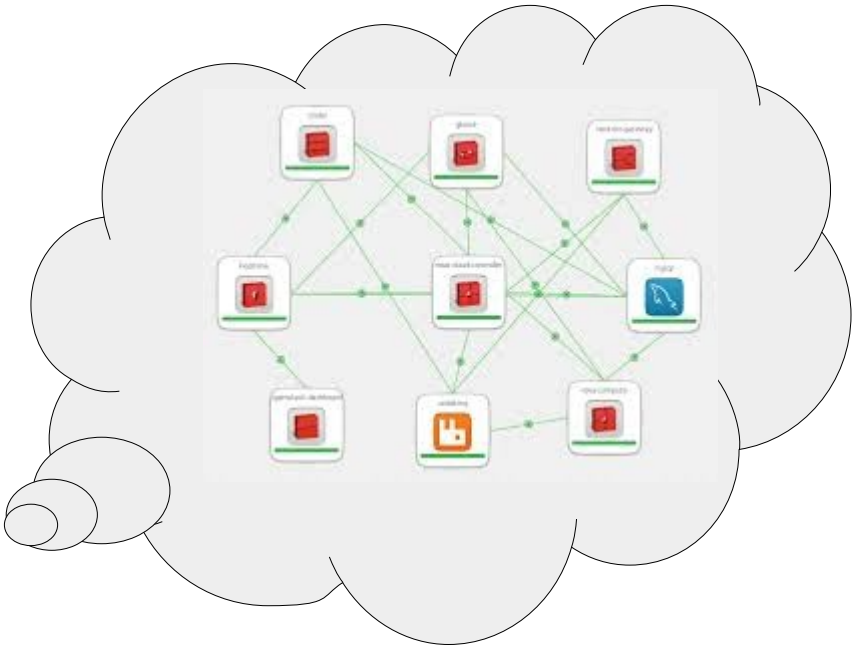
Zephyrus2: On the Fly Deployment Optimization using SMT and CP Technologies

Erika Ábrahám, Florian Corzilius, Einar Broch Johnsen, Gereon Kremer, and Jacopo Mauro

SETTA, 2016

Motivation

Automatize deployment

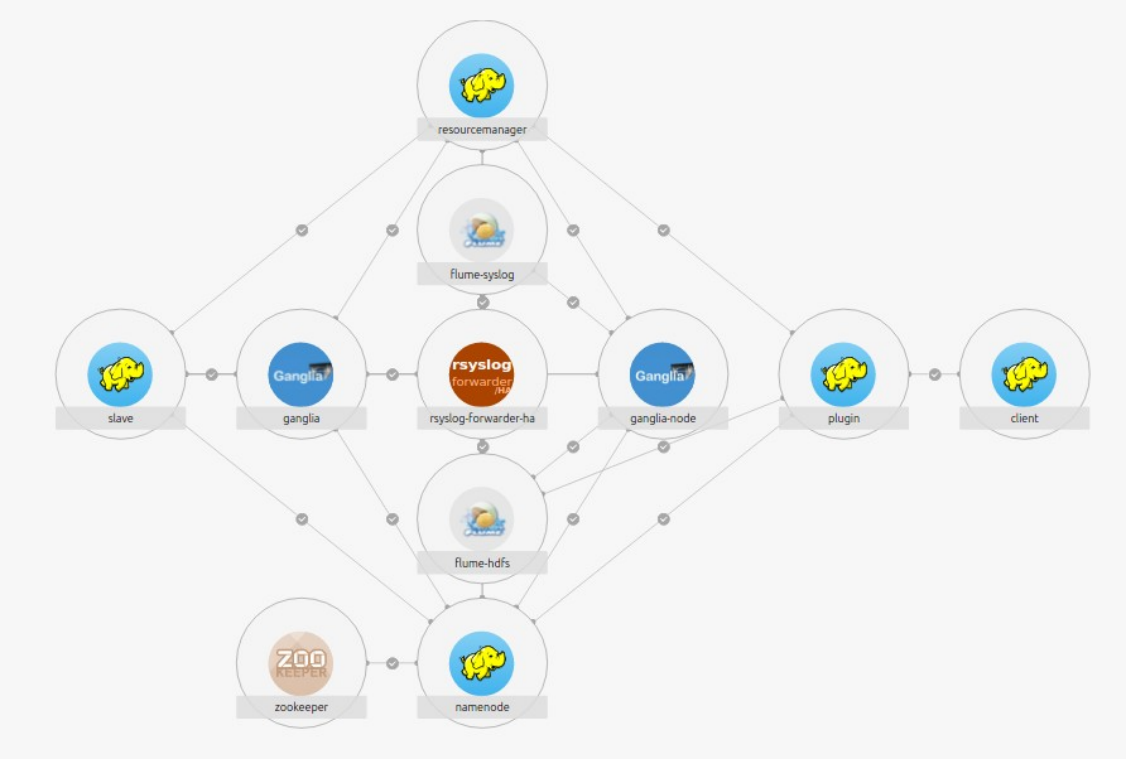


The dream

Automatize deployment

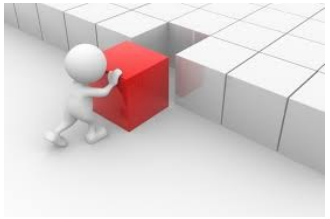
From Configuration to Deployment

TOSCA, JUJU (Canonical), Kubernetes, …



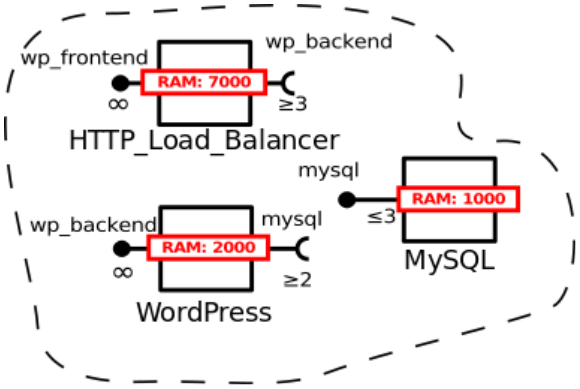
Challenges

- Cloud provides computational resources but
  - Which ones are best for us? E.g., amazon c4-xlarge or c4-2xlarge?
  - What components should be deployed and where?
  - How to connect the components?
- Zephyrus2 → Tool that helps in finding optimal configuration exploiting (cloud) computing resources



Zephyrus2

Deployable Components

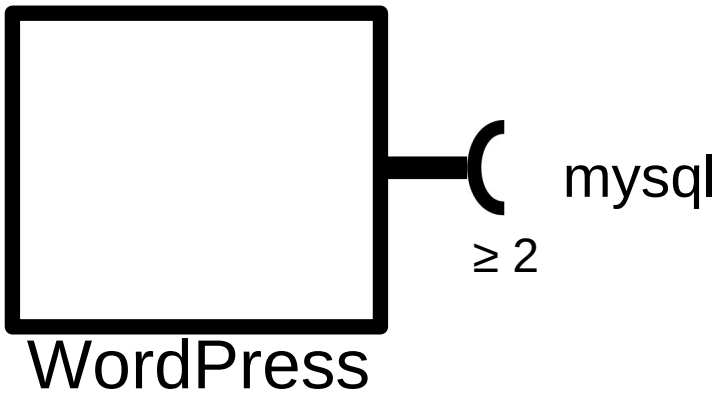


Universe of Components

Box with a name

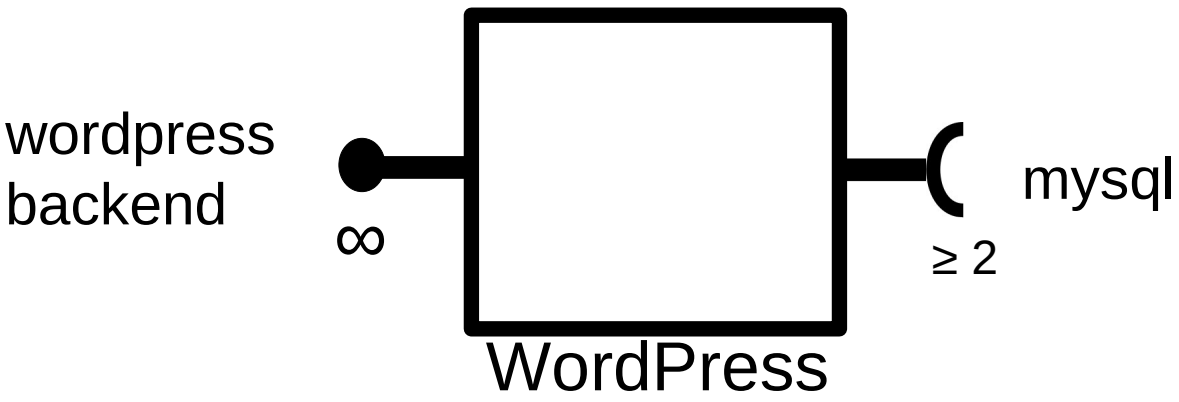
Universe of Components

Requirements



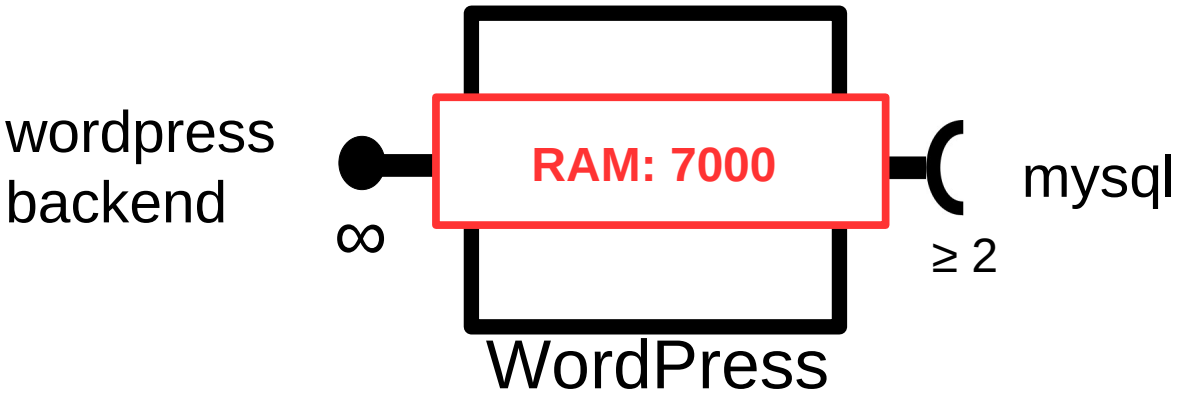
Universe of Components

Functionalities offered

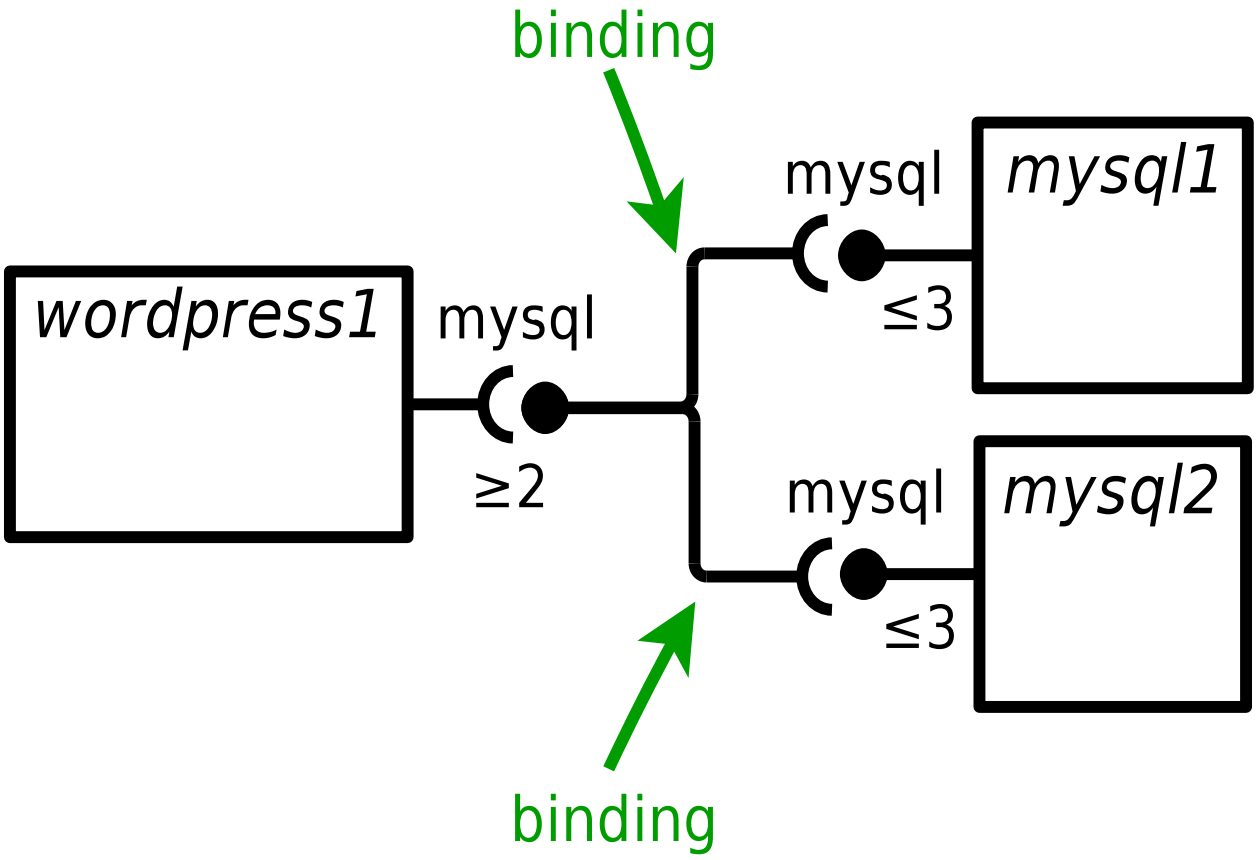


Universe of Components

Resource consumption

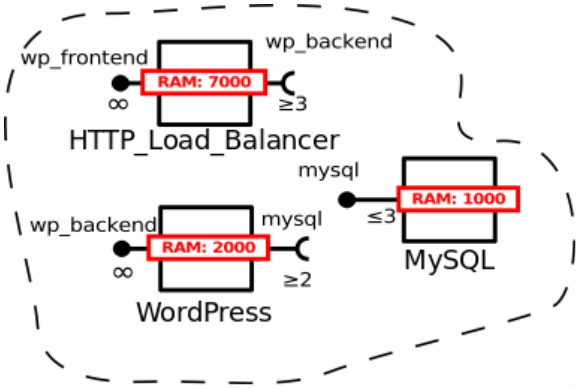


Bindings

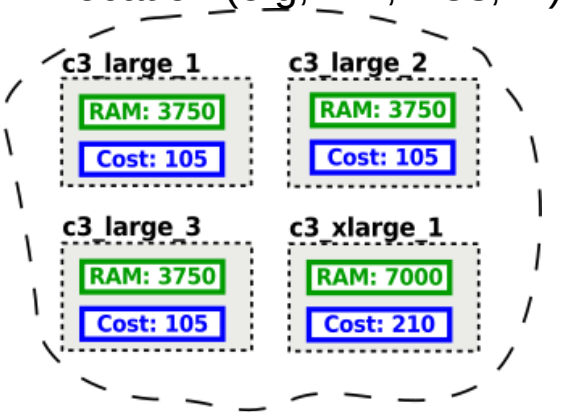

binding
mysql
mysql1
wordpress1
mysql
≤3
≥2
mysql
mysql2
≤3
binding

Zephyrus2

Deployable Components

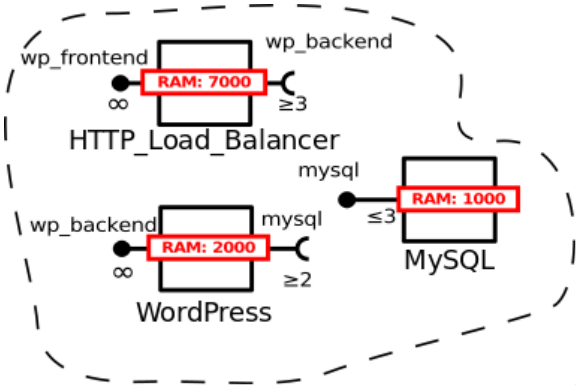


Location (e.g, VM, PCs, ...)



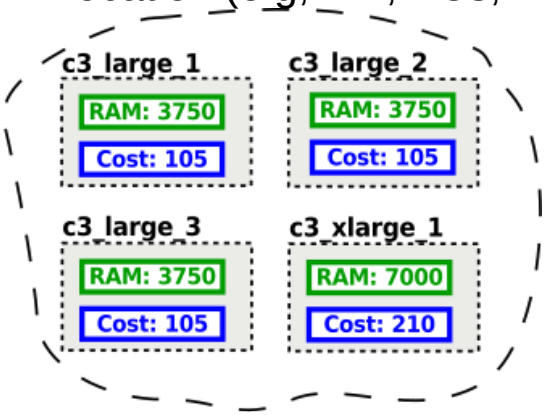
Zephyrus2

Deployable Components



User Desiderata

Location (e.g, VM, PCs, ...)

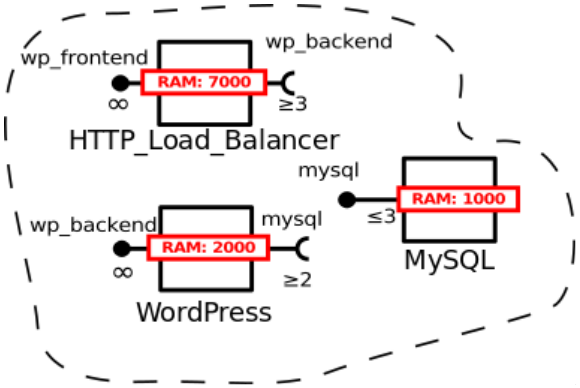


Specification : declarative

- Constraints describing VM (locations) & components
- Components in VM:
  - HTTP_Load_Balancer > 0 and c3_large[1].WordPress = 1
- Co-location:
  - forall ?x in locations: (?x.WordPress > 0 impl ?x.MySQL > 0)
- Distribution:
  - forall ?x in locations: (?x.HTTP_Load_Balancer > 0 impl
    (sum ?y in components: ?x.?y) = 1)
  - forall ?x in locations: (?x.MySQL < 2)

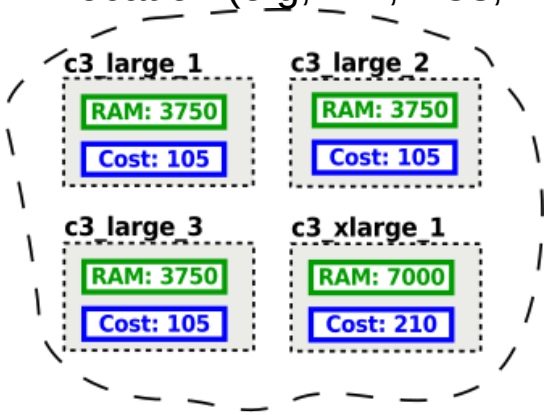
Zephyrus2

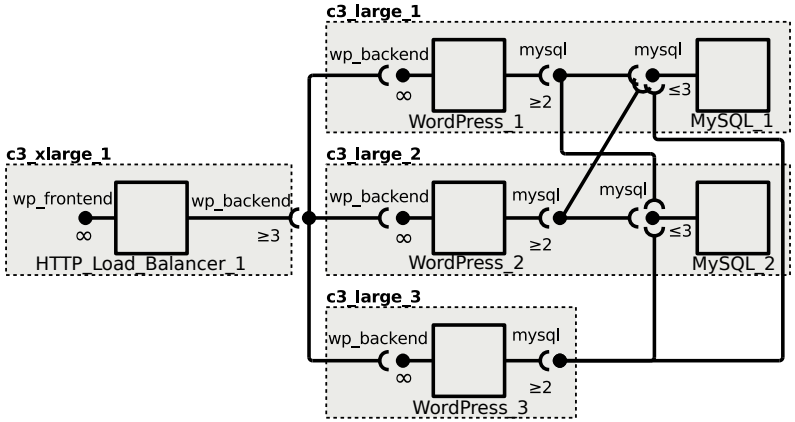
Deployable Components



User Desiderata

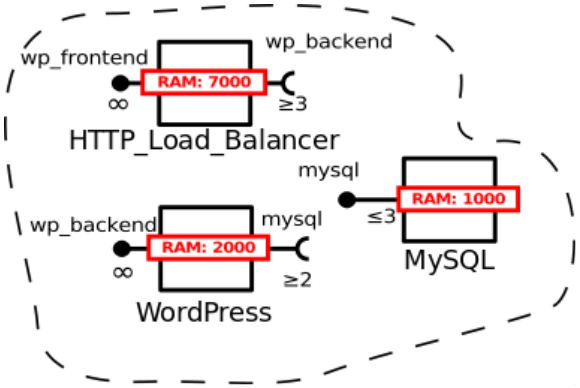
Location (e.g, VM, PCs, ...)





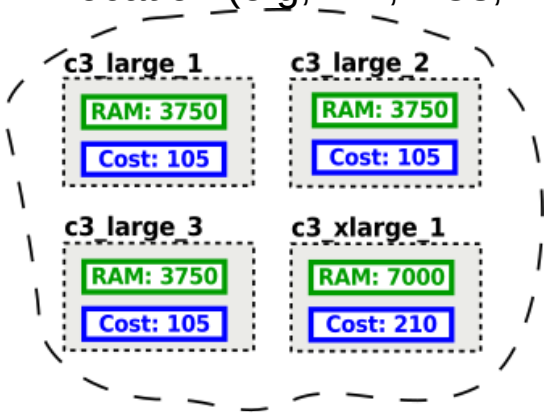
Zephyrus2

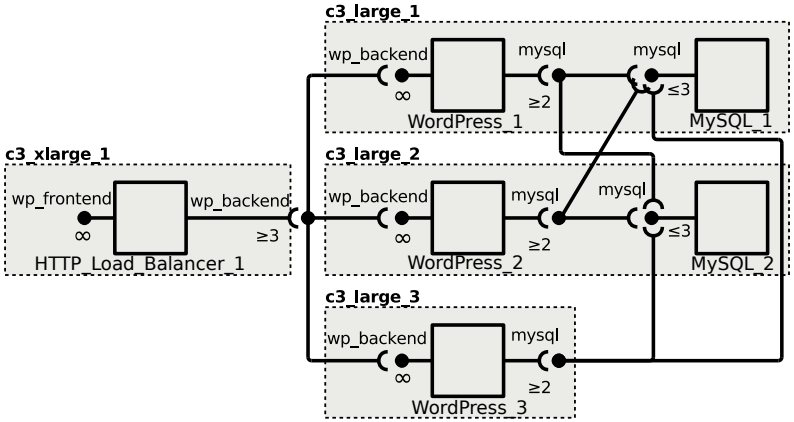
Deployable Components



User Desiderata

Location (e.g, VM, PCs, ...)





Optimization Criteria:

- less expensive,

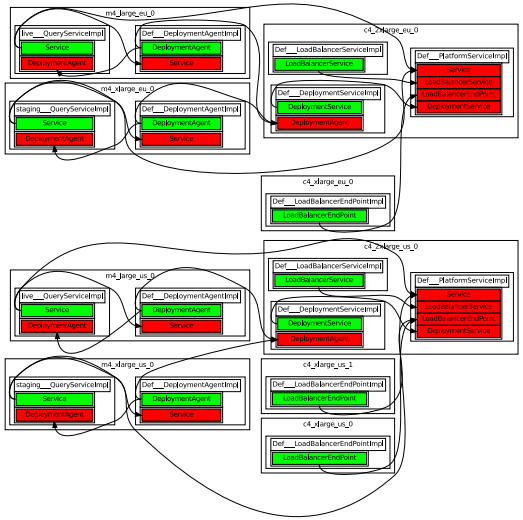
…

Internals

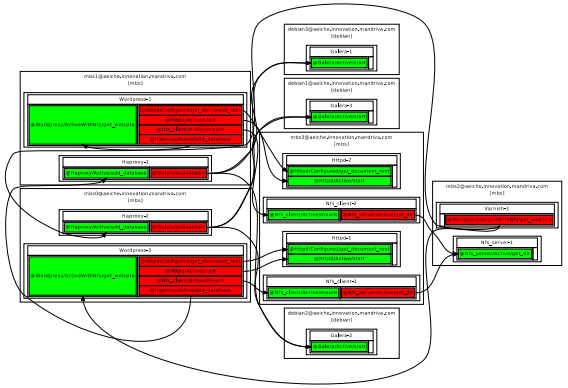
- Problem encoded into Constraint Optimization Problems
  - MiniZinc language → Use of state of the art constraint solvers
  - SMT solver → Z3 with new optimization function
- Open source:
  - https://bitbucket.org/jacopomauro/zephyrus2
- Deployable via Docker + Input and output in JSON

Complex industrial examples

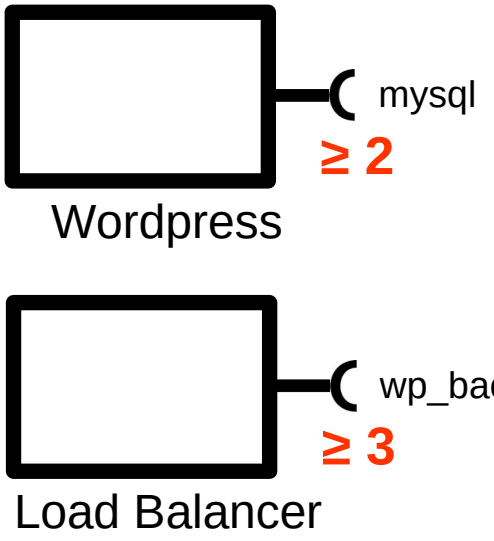
SDL

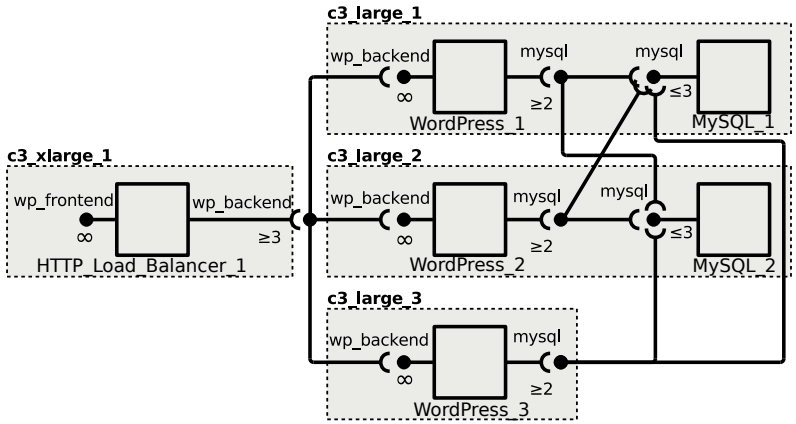


Mandriva



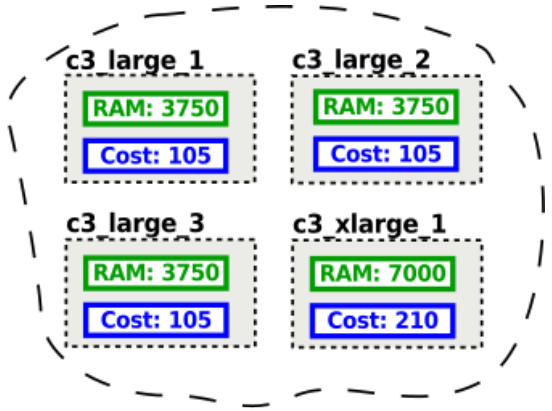
Experiments - Benchmarks





Parameters

- mysql_req: 6 … 12
- wp_req: 6 … 12
- vm_amount: 6 … 25

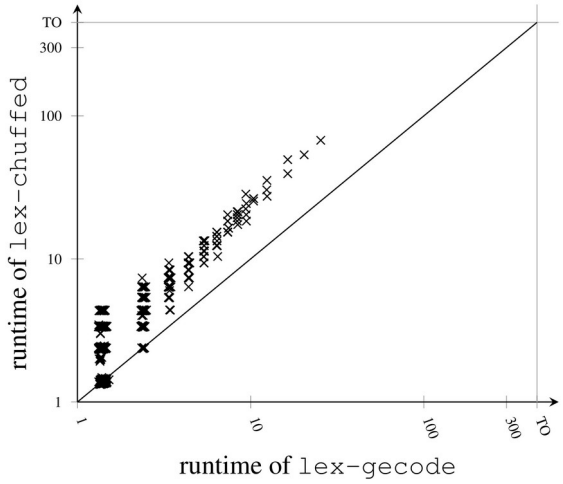


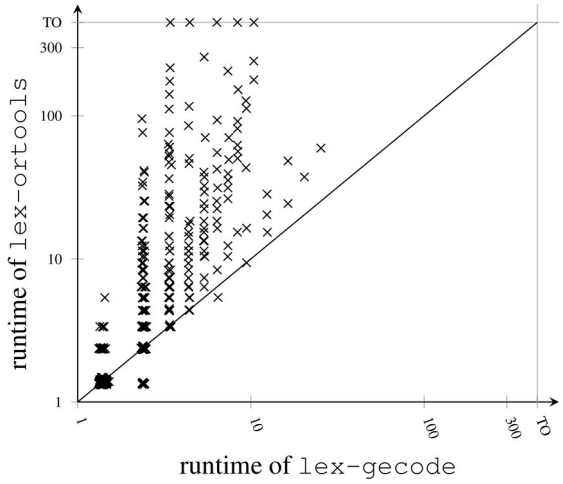
Experiments - Overview

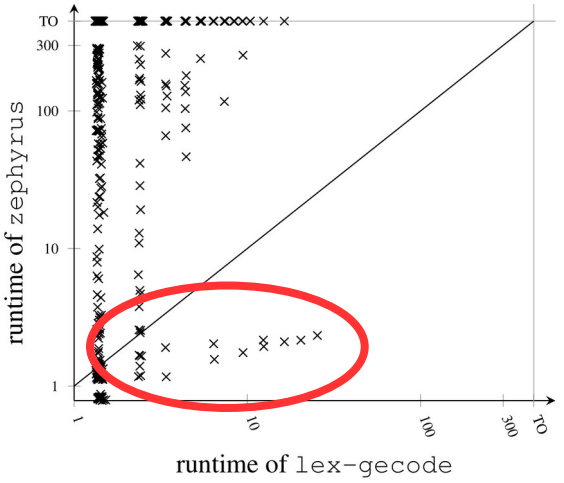
Solver	Solved	Timeout	Seconds
zephyrus	261 (27%)	719	67.81
chuffed	980 (100%)	0	4.45
Gecode	**980 (100%)**	**0**	**2.25**
OR-Tools	975 (99%)	5	7.13
Z3 (SMT)	960 (98%)	20	50.23

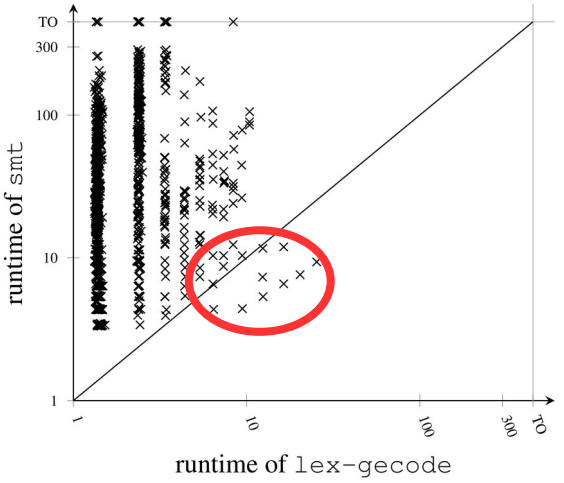
linearized, with redundant constraints, with symmetry breaking constraints

Experiments - Details









Experiments – Different Encodings

- Redundant constraints
  - Very beneficial for OR-Tools and Gecode
  - Irrelevant for chuffed and Z3
- Symmetry breaking
  - Beneficial for all solvers
  - Without: Z3 > OR-Tools > chuffed / Gecode

Z3 / SMT:

Robust, reasonably fast on suboptimal encoding

chuffed / Gecode:

Very efficient on properly encoded problems

Conclusions & Future Work

- Zepyrus2: state of the art configurator
  - Produce configuration with hundreds of components/virtual machines
  - Easily usable and deployable
- Future:
  - Better SMT encoding
  - Use portfolio techniques
  - Richer model support for components
  - More benchmarks