Harnessing over a Million CPU Cores to Solve a Single Hard Mixed Integer Programming Problem on a Supercomputer

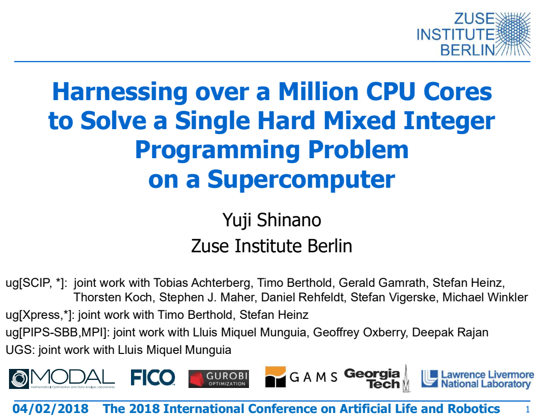
Yuji Shinano
Mathematical Optimization and Scientific Information, Zuse Institute Berlin,
Takustrasse 7, 14195 Berlin, Germany
E-mail: shinano@zib.de

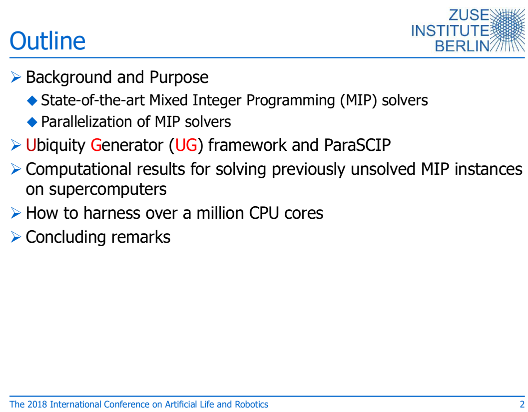
Abstract

The performance of mixed integer programming (MIP) solvers has improved tremendously in the last two decades and these solvers have been used to sol ve many real-word problems. ParaSCIP is the most successful parallel MIP solver in terms of solving previously unsolvable instances from the well-known benchmark instance set MIPLIB by using supercomputers. ParaSCIP has been developed by using the Ubiquity Generator (UG) framework, which is a general software package to parallelize any state-of-the-art branch-and-bound based solvers. ParaSCIP is a parallelized MIP solver of a single thread solver SCIP. Since Xpress is a multi-threaded solver and ParaSCIP can run at least 80,000 processes in parallel for solving a single MIP, ParaXpress could handle over a million CPU cores. In this talk, a ground design of the UG framework and its latest extensions to harness over a million CPU cores will be presented and preliminary computational results will be provided.

Keywords: mixed integer programming problem, massively parallel computing, SCIP, ParaSCIP, ParaXpress, UG

Slides





Background and Purpose

ZUSE INSTITUTE BERLIN

MIP (Mixed Integer Linear Programming)
- minimizes or maximizes a linear function
- is subject to linear constraints
- has integer and continuous variables

The most general form of combinatorial optimization problems

Many applications

The 2018 International Conference on Artificial Life and Robotics 3

Customer Applications

(2012 Gurobi Sales – 200+ new customers)

- Accounting
- Advertising
- Agriculture
- Airlines
- ATM provisioning
- Compilers
- Defense
- Electrical power
- Energy
- Finance
- Food service
- Forestry
- Gas distribution
- Government
- Internet applications
- Logistics/supply chain
- Medical
- Mining
- National research labs
- Online dating
- Portfolio management
- Railways
- Recycling
- Revenue management
- Semiconductor
- Shipping
- Social networking
- Sourcing
- Sports betting
- Sports scheduling
- Statistics
- Steel Manufacturing
- Telecommunications
- Transportation
- Utilities
- Workforce Management

Gurobi Optimization

© 2015 Gurobi Optimization

The 2018 International Conference on Artificial Life and Robotics 4

Background and Purpose

ZUSE INSTITUTE BERLIN

MIP (Mixed Integer Linear Programming)
- minimizes or maximizes a linear function
- is subject to linear constraints
- has integer and continuous variables

The most general form of combinatorial optimization problems

Many applications

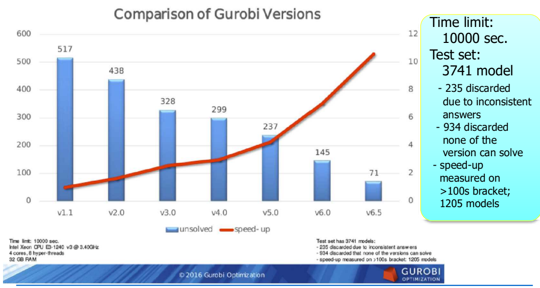
MIP solvability has been improving

The 2018 International Conference on Artificial Life and Robotics 5

Progress in a state-of-the-art MIP solver

ZUSE INSTITUTE BERLIN

Gurobi Keeps Getting Better



The 2018 International Conference on Artificial Life and Robotics 6

Background and Purpose

ZUSE INSTITUTE BERLIN

MIP (Mixed Integer Linear Programming)
- minimizes or maximizes a linear function
- is subject to linear constraints
- has integer and continuous variables

The most general form of combinatorial optimization problems

Many applications

MIP solvability has been improving

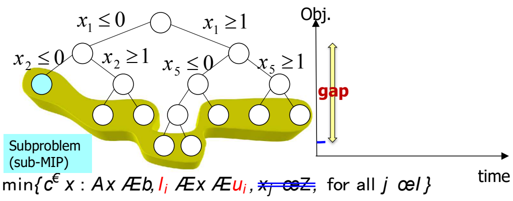
Development of a massively parallel MIP solver
- can solve instances that cannot be solved by state-of-the-art MIP solvers
- keeps catching up performance improvements of state-of-the-art MIP solvers

The 2018 International Conference on Artificial Life and Robotics 7

Parallelization of MIP solvers

ZUSE INSTITUTE BERLIN

Branch-and-bound looks suitable for parallelization
- MIP solvers: LP based Branch-and-cut algorithm

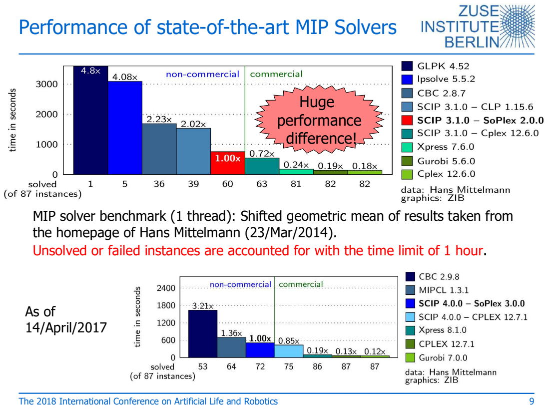


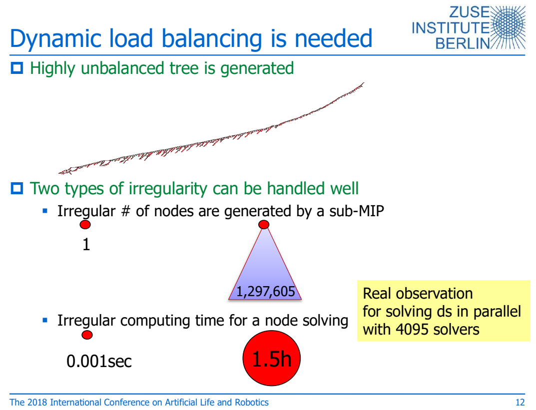
$\min\{c^T x : Ax \le b, l_i \le x \le u_i,$ ~~$x_j \in \mathbb{Z}$~~, for all $j \in I\}$

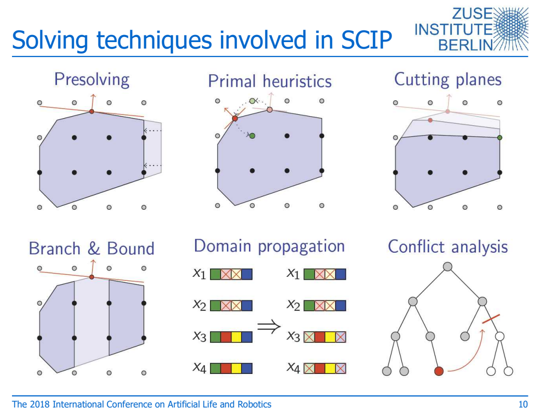
Subproblems (sub-MIPs) can be processed independently

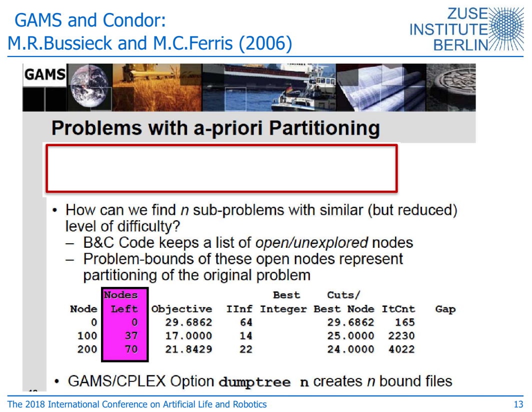
Utilize the large number of processors for solving extremely hard MIP instances (**previously unsolved problem instances from MIPLIB**)

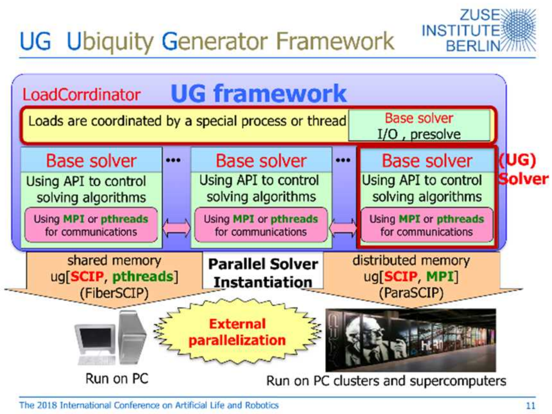
The 2018 International Conference on Artificial Life and Robotics 8

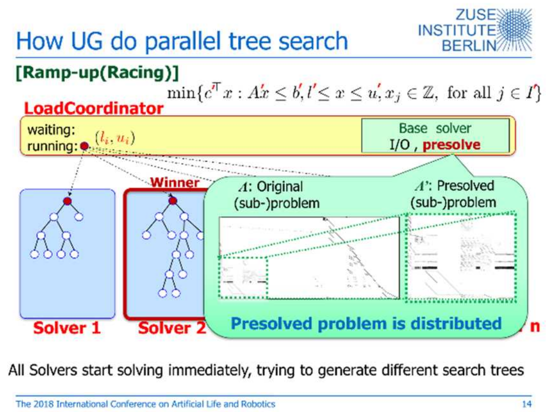

Performance of state-of-the-art MIP Solvers
ZUSE INSTITUTE BERLIN
non-commercial
commercial
Huge performance difference!
time in seconds
4.8x
4.08x
2.23x
2.02x
1.00x
0.72x
0.24x
0.19x
0.18x
solved (of 87 instances)
GLPK 4.52
lpsolve 5.5.2
CBC 2.8.7
SCIP 3.1.0 – CLP 1.15.6
SCIP 3.1.0 – SoPlex 2.0.0
SCIP 3.1.0 – Cplex 12.6.0
Xpress 7.6.0
Gurobi 5.6.0
Cplex 12.6.0
data: Hans Mittelmann
graphics: ZIB
MIP solver benchmark (1 thread): Shifted geometric mean of results taken from the homepage of Hans Mittelmann (23/Mar/2014).
Unsolved or failed instances are accounted for with the time limit of 1 hour.
As of 14/April/2017
3.21x
1.36x
1.00x
0.85x
0.19x
0.13x
0.12x
CBC 2.9.8
MIPCL 1.3.1
SCIP 4.0.0 – SoPlex 3.0.0
SCIP 4.0.0 – CPLEX 12.7.1
Xpress 8.1.0
CPLEX 12.7.1
Gurobi 7.0.0
The 2018 International Conference on Artificial Life and Robotics
9

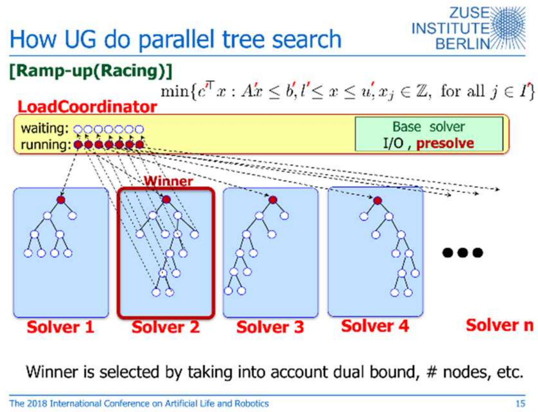

Dynamic load balancing is needed
ZUSE INSTITUTE BERLIN
Highly unbalanced tree is generated
Two types of irregularity can be handled well
Irregular # of nodes are generated by a sub-MIP
1
1,297,605
Real observation for solving ds in parallel with 4095 solvers
Irregular computing time for a node solving
0.001sec
1.5h
The 2018 International Conference on Artificial Life and Robotics
12

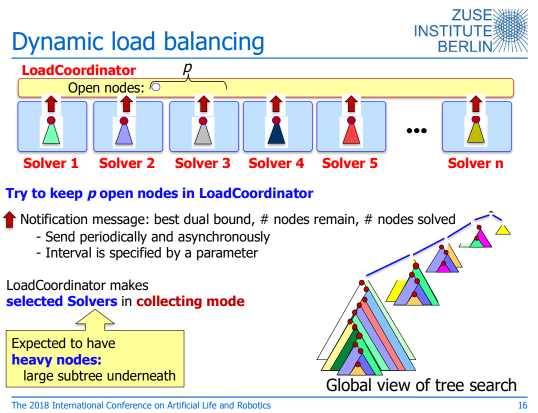

Solving techniques involved in SCIP
ZUSE INSTITUTE BERLIN
Presolving
Primal heuristics
Cutting planes
Branch & Bound
Domain propagation
Conflict analysis
The 2018 International Conference on Artificial Life and Robotics
10

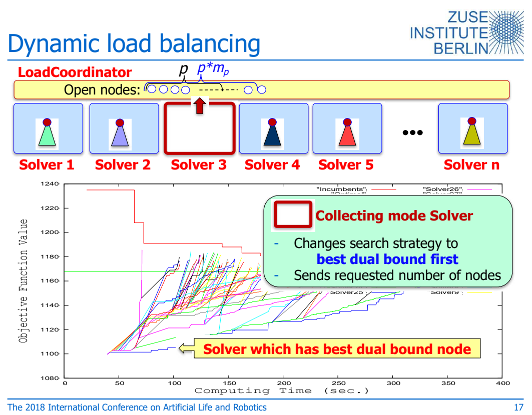

GAMS and Condor: M.R.Bussieck and M.C.Ferris (2006)
ZUSE INSTITUTE BERLIN
GAMS
Problems with a-priori Partitioning
How can we find n sub-problems with similar (but reduced) level of difficulty?
B&C Code keeps a list of open/unexplored nodes
Problem-bounds of these open nodes represent partitioning of the original problem
Node | Nodes Left | Objective | IInf | Best Integer | Cuts/ Best Node | ItCnt | Gap
0 | 0 | 29.6862 | 64 | | 29.6862 | 165
100 | 37 | 17.0000 | 14 | | 25.0000 | 2230
200 | 70 | 21.8429 | 22 | | 24.0000 | 4022
GAMS/CPLEX Option dumptree n creates n bound files
The 2018 International Conference on Artificial Life and Robotics
13

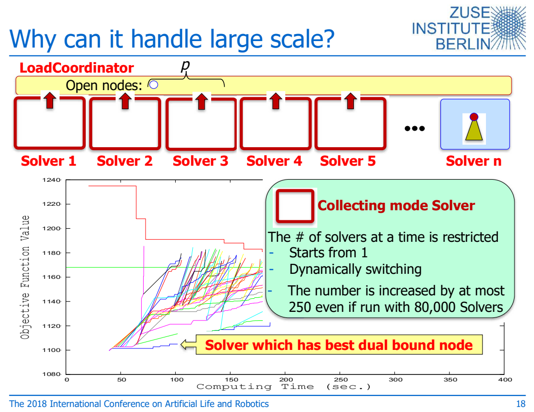

UG Ubiquity Generator Framework
ZUSE INSTITUTE BERLIN
LoadCorrdinator
UG framework
Loads are coordinated by a special process or thread
Base solver I/O , presolve
Base solver
Using API to control solving algorithms
Using MPI or pthreads for communications
(UG) Solver
shared memory ug[SCIP, pthreads] (FiberSCIP)
Parallel Solver Instantiation
distributed memory ug[SCIP, MPI] (ParaSCIP)
External parallelization
Run on PC
Run on PC clusters and supercomputers
The 2018 International Conference on Artificial Life and Robotics
11

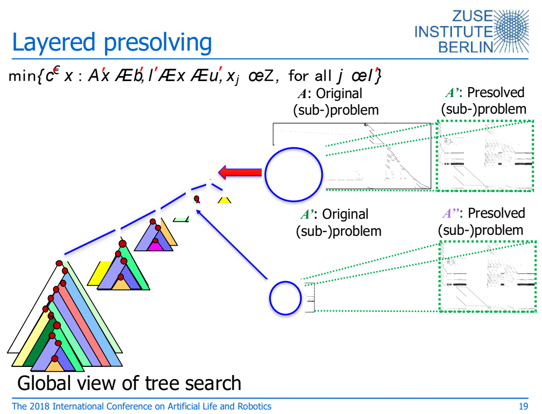

How UG do parallel tree search
ZUSE INSTITUTE BERLIN
[Ramp-up(Racing)]
$\min\{c^T x : A'x \le b', l' \le x \le u', x_j \in \mathbb{Z}, \text{ for all } j \in I'\}$
LoadCoordinator
waiting:
running:
Base solver I/O , presolve
Winner
A: Original (sub-)problem
A': Presolved (sub-)problem
Solver 1
Solver 2
Presolved problem is distributed
All Solvers start solving immediately, trying to generate different search trees
The 2018 International Conference on Artificial Life and Robotics
14

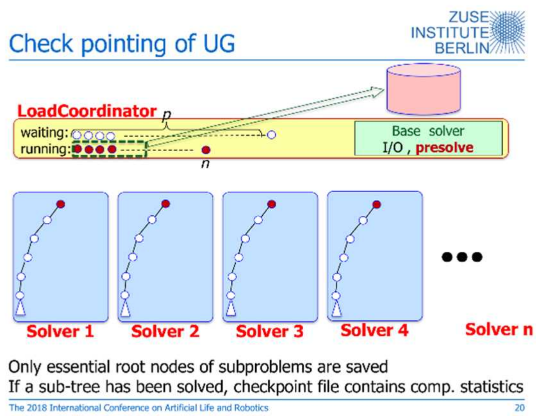

How UG do parallel tree search
ZUSE INSTITUTE BERLIN
[Ramp-up(Racing)]
LoadCoordinator
waiting:
running:
Base solver
I/O , presolve
Winner
Solver 1
Solver 2
Solver 3
Solver 4
Solver n
Winner is selected by taking into account dual bound, # nodes, etc.
The 2018 International Conference on Artificial Life and Robotics
15

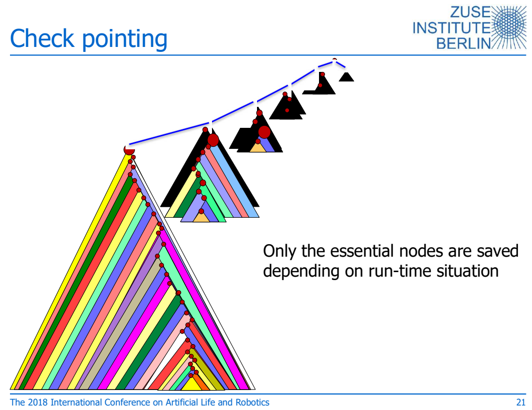

Dynamic load balancing
ZUSE INSTITUTE BERLIN
LoadCoordinator
Open nodes:
Solver 1
Solver 2
Solver 3
Solver 4
Solver 5
Solver n
Try to keep p open nodes in LoadCoordinator
Notification message: best dual bound, # nodes remain, # nodes solved
- Send periodically and asynchronously
- Interval is specified by a parameter
LoadCoordinator makes
selected Solvers in collecting mode
Expected to have
heavy nodes:
large subtree underneath
Global view of tree search
The 2018 International Conference on Artificial Life and Robotics
16

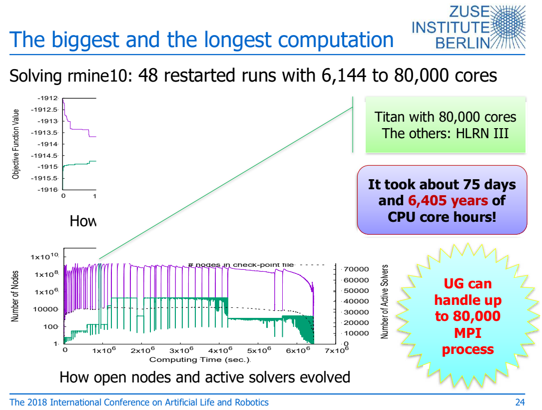

Dynamic load balancing
ZUSE INSTITUTE BERLIN
LoadCoordinator
Open nodes:
Solver 1
Solver 2
Solver 3
Solver 4
Solver 5
Solver n
Collecting mode Solver
- Changes search strategy to
best dual bound first
- Sends requested number of nodes
Solver which has best dual bound node
Objective Function Value
Computing Time (sec.)
The 2018 International Conference on Artificial Life and Robotics
17

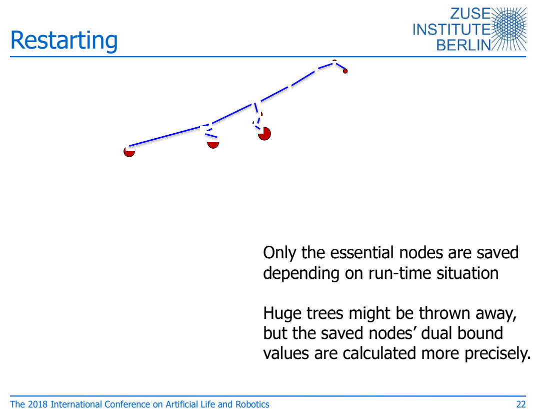

Why can it handle large scale?
ZUSE INSTITUTE BERLIN
LoadCoordinator
Open nodes:
Solver 1
Solver 2
Solver 3
Solver 4
Solver 5
Solver n
Collecting mode Solver
The # of solvers at a time is restricted
- Starts from 1
- Dynamically switching
- The number is increased by at most
250 even if run with 80,000 Solvers
Solver which has best dual bound node
Objective Function Value
Computing Time (sec.)
The 2018 International Conference on Artificial Life and Robotics
18

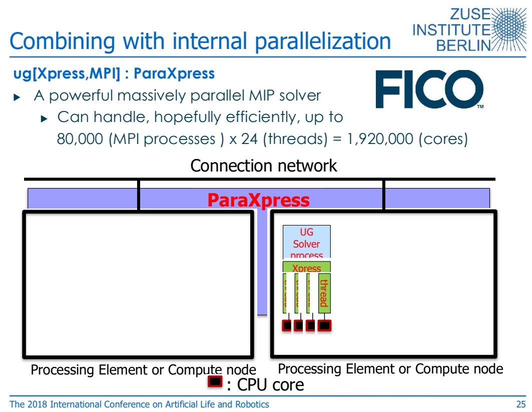

Layered presolving
ZUSE INSTITUTE BERLIN
A: Original
(sub-)problem
A': Presolved
(sub-)problem
A': Original
(sub-)problem
A'': Presolved
(sub-)problem
Global view of tree search
The 2018 International Conference on Artificial Life and Robotics
19

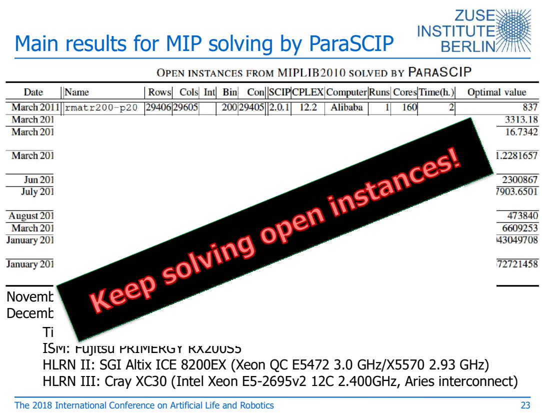

Check pointing of UG
ZUSE INSTITUTE BERLIN
LoadCoordinator
waiting:
running:
Base solver
I/O , presolve
Solver 1
Solver 2
Solver 3
Solver 4
Solver n
Only essential root nodes of subproblems are saved
If a sub-tree has been solved, checkpoint file contains comp. statistics
The 2018 International Conference on Artificial Life and Robotics
20

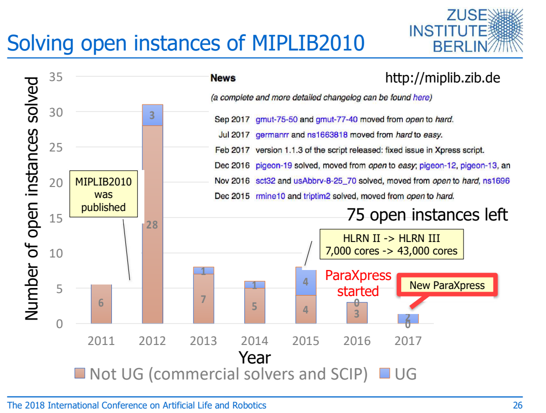

Check pointing
ZUSE INSTITUTE BERLIN
Only the essential nodes are saved depending on run-time situation
The 2018 International Conference on Artificial Life and Robotics
21

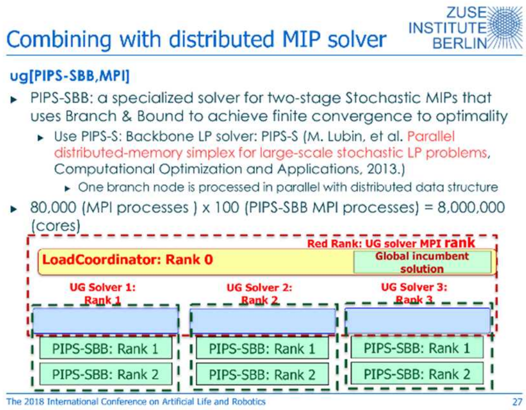

The biggest and the longest computation
ZUSE INSTITUTE BERLIN
Solving rmine10: 48 restarted runs with 6,144 to 80,000 cores
Titan with 80,000 cores
The others: HLRN III
It took about 75 days and 6,405 years of CPU core hours!
UG can handle up to 80,000 MPI process
Objective Function Value
Number of Nodes
Number of Active Solvers
Computing Time (sec.)
How open nodes and active solvers evolved
The 2018 International Conference on Artificial Life and Robotics
24

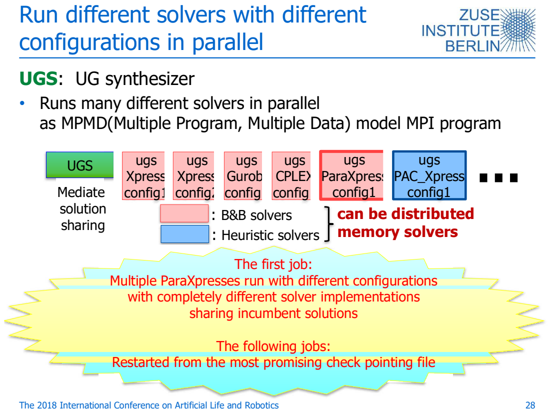

Restarting
ZUSE INSTITUTE BERLIN
Only the essential nodes are saved depending on run-time situation
Huge trees might be thrown away, but the saved nodes' dual bound values are calculated more precisely.
The 2018 International Conference on Artificial Life and Robotics
22


Combining with internal parallelization
ZUSE INSTITUTE BERLIN
ug[Xpress,MPI] : ParaXpress
FICO
A powerful massively parallel MIP solver
Can handle, hopefully efficiently, up to 80,000 (MPI processes) x 24 (threads) = 1,920,000 (cores)
Connection network
ParaXpress
UG Solver process
Xpress
thread
Processing Element or Compute node
Processing Element or Compute node
: CPU core
The 2018 International Conference on Artificial Life and Robotics
25


Main results for MIP solving by ParaSCIP
ZUSE INSTITUTE BERLIN
OPEN INSTANCES FROM MIPLIB2010 SOLVED BY PARASCIP
Date | Name | Rows | Cols | Int | Bin | Con | SCIP | CPLEX | Computer | Runs | Cores | Time(h.) | Optimal value
March 2011 | rmatr200-p20 | 29406 | 29605 | | 200 | 29405 | 2.0.1 | 12.2 | Alibaba | 1 | 160 | 2 | 837
Keep solving open instances!
HLRN II: SGI Altix ICE 8200EX (Xeon QC E5472 3.0 GHz/X5570 2.93 GHz)
HLRN III: Cray XC30 (Intel Xeon E5-2695v2 12C 2.400GHz, Aries interconnect)
The 2018 International Conference on Artificial Life and Robotics
23


Solving open instances of MIPLIB2010
ZUSE INSTITUTE BERLIN
News
http://miplib.zib.de
(a complete and more detailed changelog can be found here)
Sep 2017 gmut-75-50 and gmut-77-40 moved from open to hard.
Jul 2017 germanrr and ns1663818 moved from hard to easy.
Feb 2017 version 1.1.3 of the script released: fixed issue in Xpress script.
Dec 2015 rmine10 and triptim2 solved, moved from open to hard.
75 open instances left
MIPLIB2010 was published
HLRN II -> HLRN III
7,000 cores -> 43,000 cores
ParaXpress started
New ParaXpress
Number of open instances solved
Year
Not UG (commercial solvers and SCIP)
UG
The 2018 International Conference on Artificial Life and Robotics
26





Conclusions

UG is a general framework to parallelize any kind of state-of-the art branch-and-bound based solver.

ug[SCIP,* is tool to develop parallel general branch-and-cut solvers. Customized SCIP solver can be parallelized with least effort.

ug[SCIP-Jack,*] is solver for Steiner Tree Problems and its variants, namely only the solver which can run on a distributed memory computing environment (solved three open benchmark instances.

ug[Xpress, MPI](=ParaXpress) and ug[PIPS-SBB, MPI] is ready to run on over a million CPU cores.

UGS is another general framework to configure a parallel solver that can realize any combination of algorithm portfolio and racing.

References

1. Y.Shinano: T*he Ubiquity Generator Framework: 7 Years of Progress in Parallelizing Branch-and-Bound*, ZIB-Report (17-60), 2017.
2. Y.Shinano, S.Heinz, S.Vigerske,M.Winker: FiberSCIP - A shared memory parallelization of SCIP, INFORMS *Journal on Computing, 30(1):11–30, 2018*
3.Y.Shinano, T.Achterberg, T.Berthold, S.Heinz, T.Koch, M.Winker: Solving Hard MIPLIB2003 Problems with ParaSCIP on Supercomputers: An Update, *Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International,* pp.1552 – 1561, 2014
4. Y.Shinano, T.Achterberg, T.Berthold, S.Heinz, T.Koch, M.Winker: Solving Open MIP Instances with ParaSCIP on Supercomputers using up to 80,000 Cores, *Proc. of 30th IEEE International Parallel & Distributed Processing Symposium*, pp.770-779, 2016
5. Y.Shinano, T.Berthold, S.Heinz, A First Implementation of ParaXpress: Combining Internal and External Parallelization to Solve MIPs onSupercomputers, to appear *ICMS 2016 proceedings*.
6. Lluis-Miquel Munguia, Geoffrey Oxberry, Deepak Rajan, Yuji Shinano: *Parallel PIPS-SBB: Multi-Level Parallelism For Stochastic Mixed-Integer Programs*, ZIB-Report (17-58), 2017.
7. Gerald Gamrath, Thorsten Koch, Stephen Maher, Daniel Rehfeldt, Yuji Shinano: SCIP-Jack – A solver for STP and variants with parallelization extensions, *Mathematical Programming Computation*, 9(2), pp. 231-296, 2017