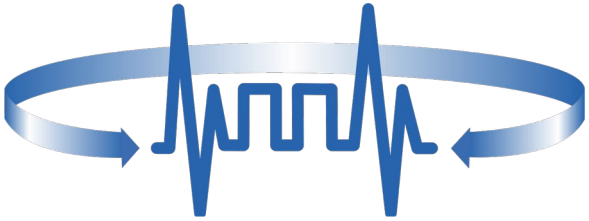
The 9th International Conference "Distributed Computing and Grid Technologies in Science and Education" (GRID'2021)

Concurrently employing resources of several supercomputers with ParaSCIP solver by Everest platform

Sergey Smirnov, Vladimir Voloshinov,
Oleg Sukhoroslov

Center for Distributed Computing,
Institute for Information Transmission Problems
of the Russian Academy of Sciences (Kharkevich Institute)



This work is supported by the Russian Science Foundation (project 20-07-00701).

Discrete Mathematical Programming (MILP & MINLP)

$$f_o(x) \to \min_x,$$
$$x = (x_B, x_C) \in Q, x_B \in \{0,1\}^{n_B}, x_C \in \mathbb{R}^{n_C}$$

(P)

$$Q = \left\{ f_i(x_B, x_C) \leqslant 0 (i \in \mathbf{I}), g_j(x_B, x_C) = 0 (j \in \mathbf{J}) \right\}$$
$$= \text{may be something else } \ldots$$

Branch-and-Cut algorithm is usually used: Branch-and-Bound (B&B) + cutting-plane method.

VERY briefly, B&B based on two interacting procedures:

Building the Search Tree

Recursive decomposition of feasible domain (Q), e.g. by fixing some x_B variables in accordance with some rules

Pruning Branch & Get Incumbents

Get lower bounds of obj. value for domain subsets;

search feasible solutions $x' \in Q$ and keep the best ones, aka incumbents $f_o(x')$

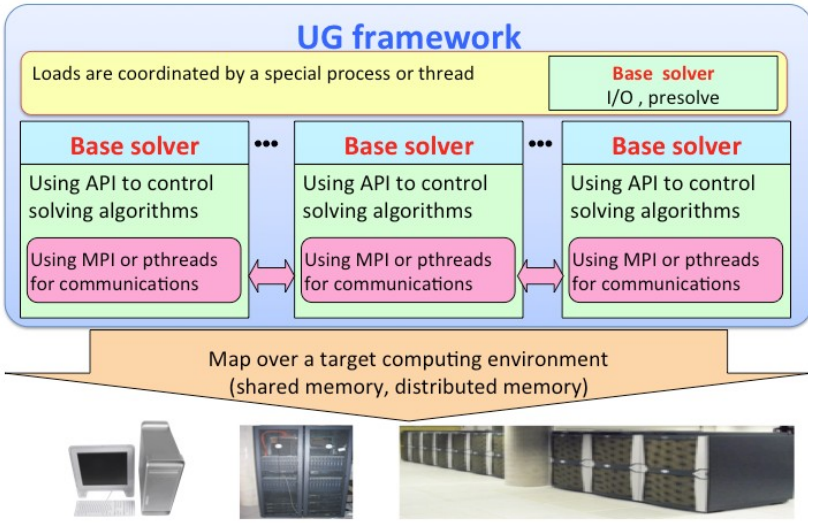
ParaSCIP, Zuse Institute Berlin

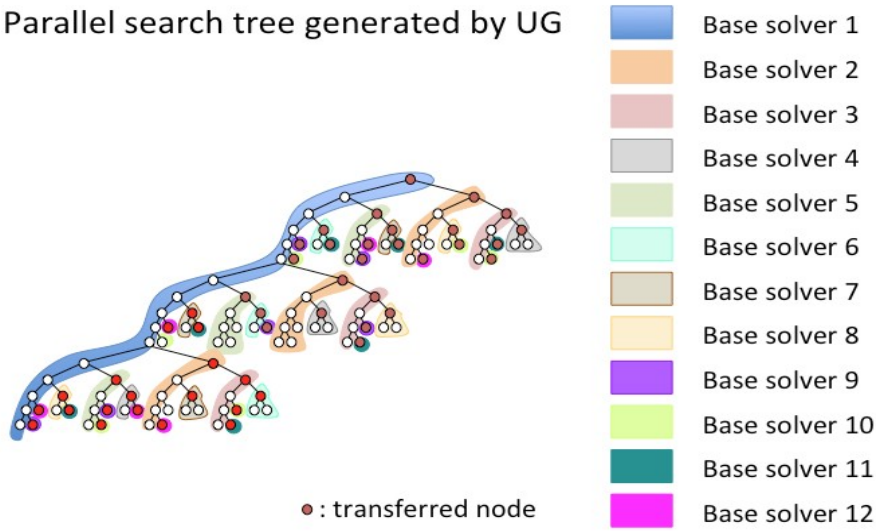
Parallel implementation of B&B via SCIP and MPI for High-Performance Computing environments, http://ug.zib.de/

UG (**Ubiquity Generator**) is a framework to parallelize B&B solvers in a distributed or shared memory computing environment.

ParaSCIP = UG[SCIP, MPI], *FiberSCIP=UG[SCIP, **Pthreads**]*, *ParaXpress*=UG[Xpress, MPI],…

Yuji Shinano, Tobias Achterberg, Timo Berthold, Stefan Heinz, Thorsten Koch, *ParaSCIP -- a parallel extension of SCIP*, **2012**





ParaSCIP, Zuse Institute Berlin (2)

Success story of solving open instances from MIPLIB2010 on:

North-German Supercomputing Alliance (Zuse Institute), Germany:

- **HLRN-II**, ~12 000 cores, https://www.hlrn.de/home/view/System2
- **HLRN-III**, ~40 000 cores, SGI Cray, https://-*-/System3

Experiments with 1024 – 12000 cores, 1 – 200 hours

Oak Ridge National Laboratory, USA

- Titan, Cray XK7, ~500000 cores, http://www.olcf.ornl.gov/titan

Experiments with 80 000 cores.

Small experience solving nonlinear problems, MILP - basically

Our input: HPC4/HPC5, NRC "Kurchatov Institute", ~22 000 cores, T–Platforms (5 in Russia Top50)

Experiments with MINLP: Thomson problems (N=5),

Flat Torus Packing problem N=9 – open conjecture has been proved (it took 128 cores * ~16 hours = 2048 CPU*hours)

Why several supercomputers?

- Problems:
  - Jobs for 20-30 nodes and 400-720 CPUs crash on HPC4
  - With memory demanding problems we have to allocate full nodes but not utilize some of the CPUs
  - Hard to allocate large jobs

Running on Multiple Clusters

- Run ParaSCIP on multiple clusters unchanged
  - Supercomputers' queues are managed by SLURM
  - Supercomputers are behind firewalls, nodes are not accessible from the outside
  - MPICH-G2, PACX-MPI, QCG-OMPI, etc. do not seem to be suitable
- Use checkpoint files
- Alternative communication mechanism for UG framework
- Integration of ParaSCIP and DDBNB

Programmatic implementation of DomainDecompositionB&B

- DDBNB, https://github.com/distcomp/ddbnb
- Basic “ingredients”:
- High-level optimization modeling tools to perform decomposition:

  AMPL, A Modeling Language for Math. Program., ampl.com

  Pyomo (free), PYthon Optimization Modeling, pyomo.org, AMPL-Compatible (!)

- B&B solvers, AMPL-compatible, with open API:
  - CBC, COIN-OR Branc-and-Cut, https://github.com/coin-or/Cbc;
  - SCIP, Solve Constraint Integer Problem, http://scip.zib.de, MIQCP

- Web-based platform, Everest, http://everest.distcomp.org provides:

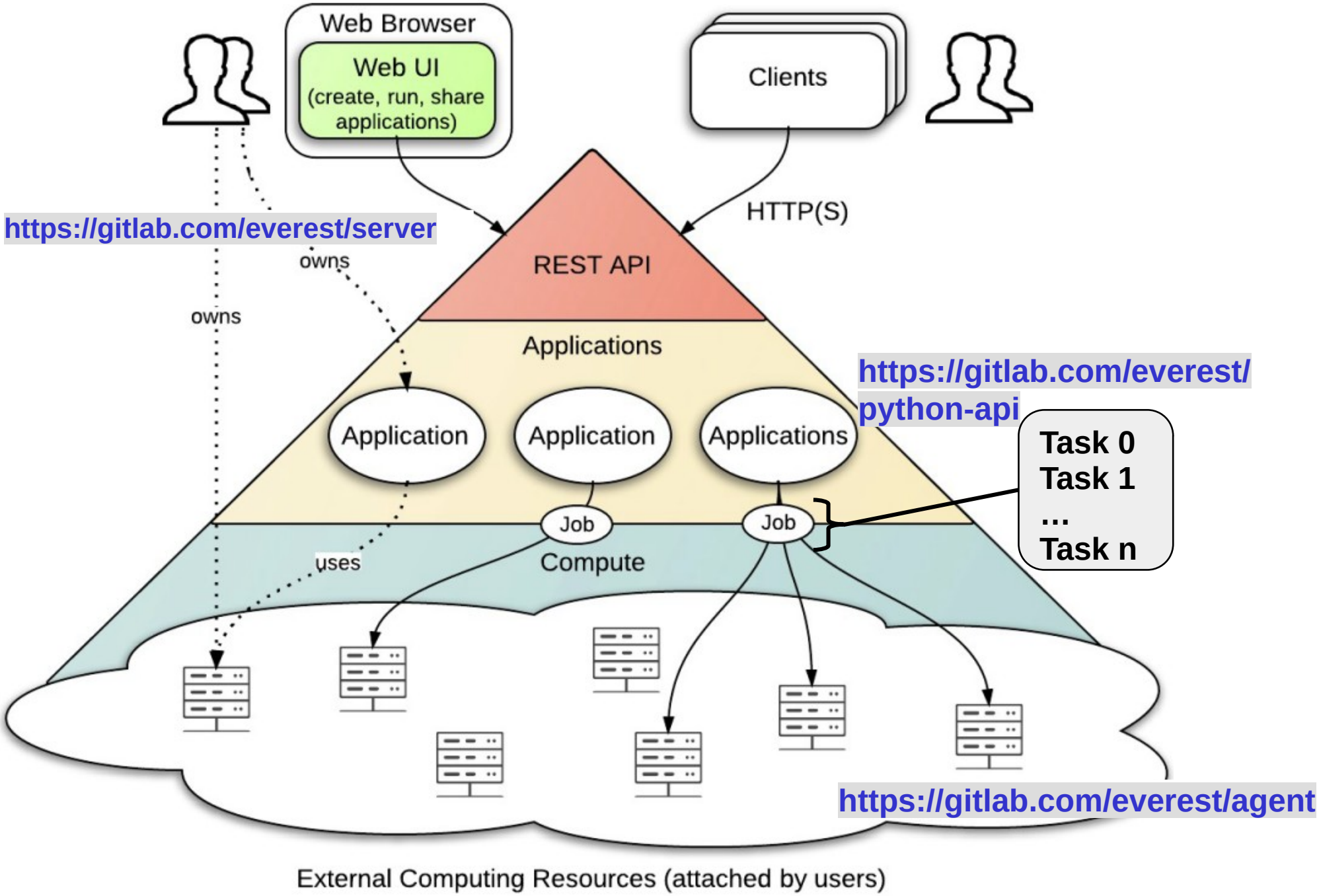
  integration of solvers installed on heterogeneous resources;

  generic service to run a pack of predefined tasks (subproblems);

  generic communication mechanism to exchange incumbents.

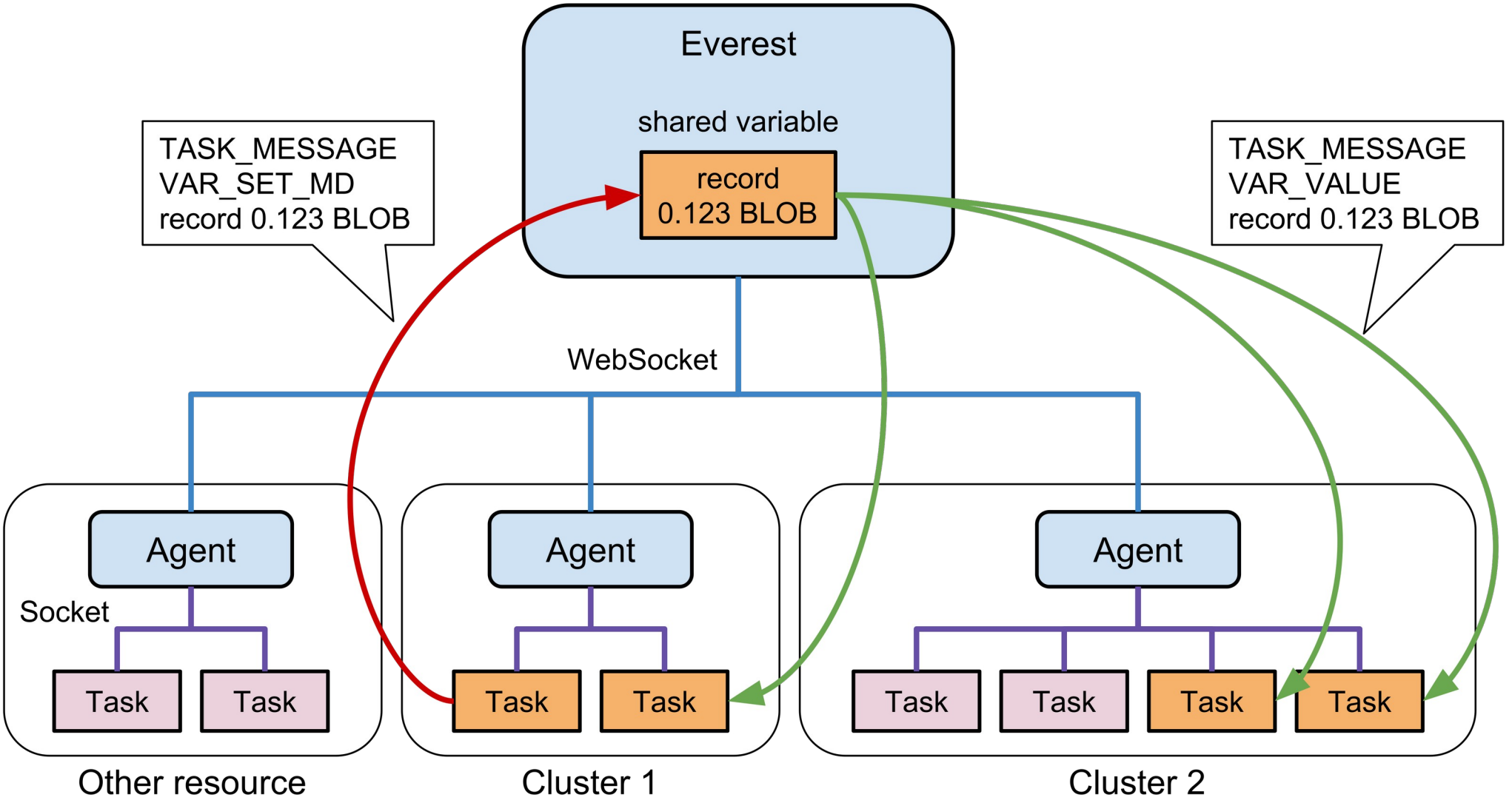
Everest web-based platform, everest.distcomp.org

Describe/Develop/Deploy REST-services representing existing applications



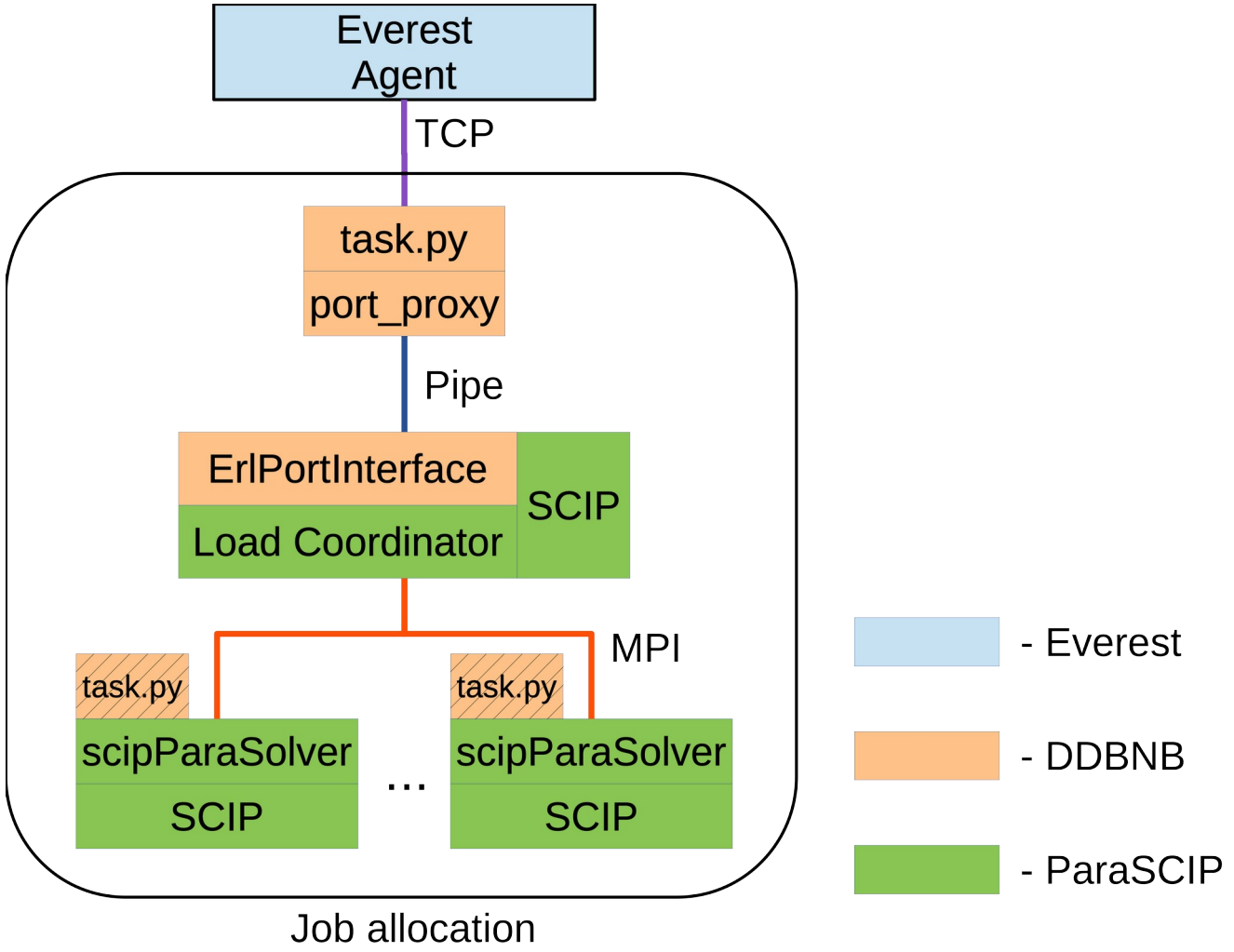
Message exchange via shared variables in Everest

Send message = update shared variable => “multicast” incumbent value



For DDBNB with ParaSCIP: each Task is running ParaSCIP solver on cluster and processes MILP/MINLP subproblem

Interaction with solver



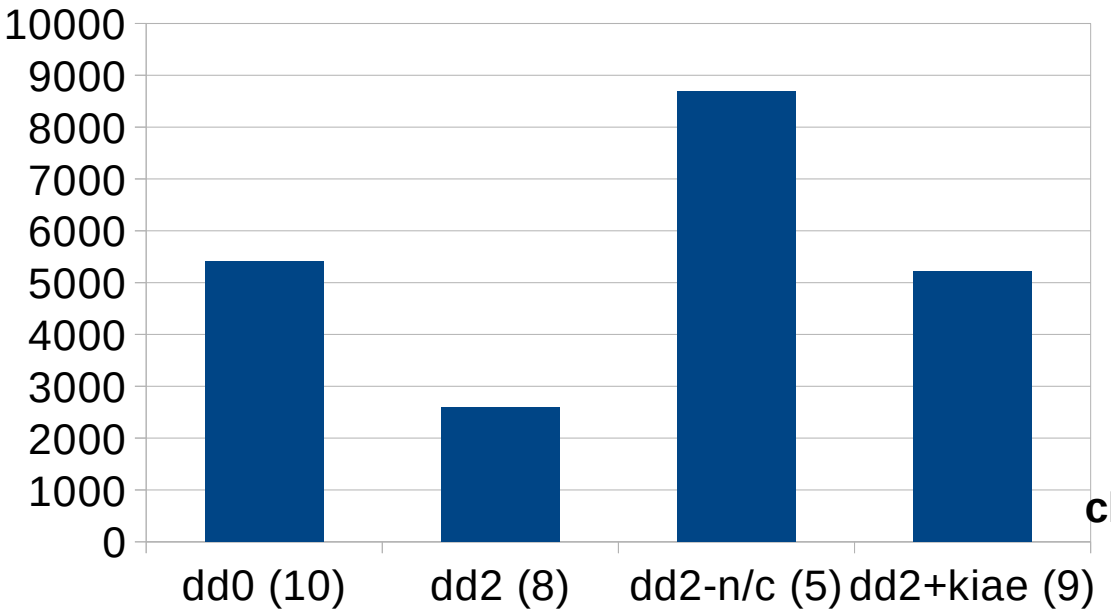
- Everest: task execution and exchange of incumbents
- In ParaSCIP there is a single distinguished Load Coordinator (LC) process, other processes are workers (solver)
- task.py/port_proxy – solver adapter passing incumbents to Everest and back

ParaSCIP (UG) Modification

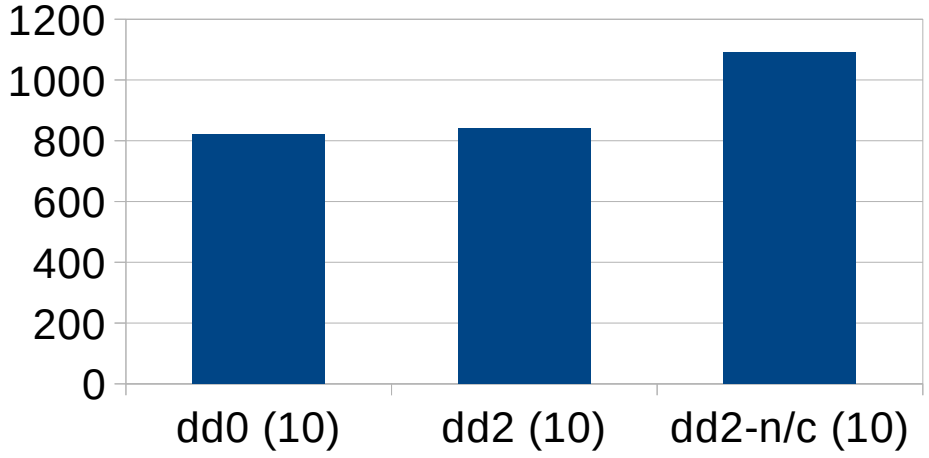
- Exchange not only objective function values but also decision variable values
- Convert solution coordinates from original problem coordinates to transformed (presolved) ones and back
- Fix a crash when loading a solution from file
- MPI_THREAD_MULTIPLE is not available on every supercomputer
- Had to read solution coming from the outside in the LC message loop's thread

Computing Experiment (1)

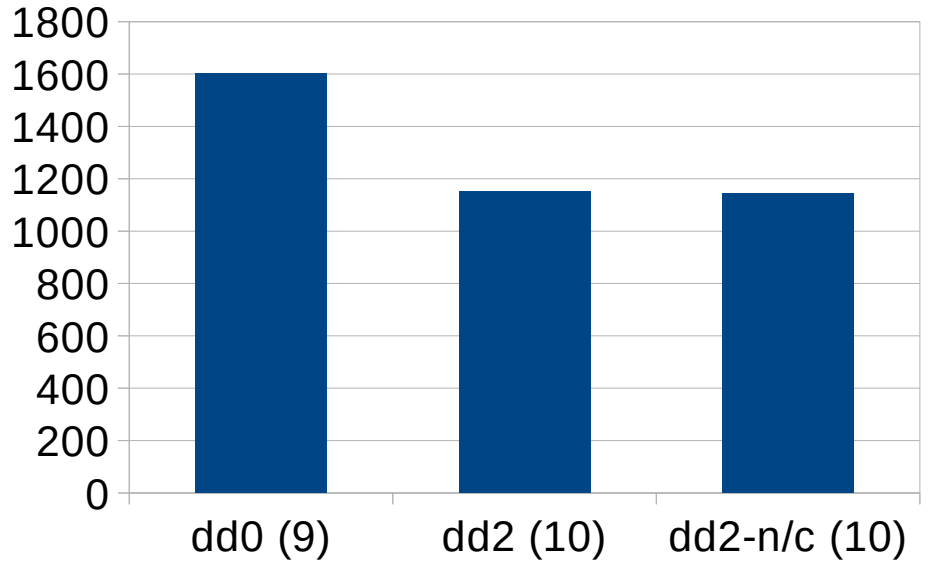
ch150, 32 processes per task, HSE cluster



bier127, 16 processes per task, HSE cluster



ch130, 16 processes per task, HSE cluster



- dd0 – original problem
- dd2 – 4 subproblems (ParaSCIPs running concurrently)
- n/c – no exchange of incumbents
- +kiae – two tasks on HSE cluster, two – on HPC5 in NRC «Kurchatov Institute»
- Number of runs in brackets

- Traveling salesman problem instances from TSPLIB
- Running time deviation is quite large for some instances

Computing Experiment (2)

ch150, 32 processes per ParaSCIP

dd0	dd2	dd2-n/c	dd2+kiae
12993	1908	8480	7178
3232	1357	8692	1706
5495	2708	10430	6660
7607	3968	4291	5223
1322	2967	8842	9998
11650	1402		1717
3203	2503		5314
3244	2679		3273
6644			4918
5333			
median			
5414	2591	8692	5223
stdev			
3792	868	2290	2672

bier127, 16 processes ...

dd0	dd2	dd2-n/c
837	847	1099
823	932	1019
817	932	1035
817	822	913
857	832	1007
1325	872	1247
993	807	3510
814	817	1313
801	882	1080
815	806	1137
median		
820	840	1090
stdev		
163	48	773

ch130, 16 processes ...

dd0	dd2	dd2-n/c
1604	889	1142
1670	1122	1127
1164	1082	1437
985	1181	1182
1787	1899	1144
1405	1075	1062
1778	2050	1614
2487	955	1809
1305	1475	953
	1345	1127
median		
1604	1152	1143
stdev		
470	386	284

Conclusion

- Implemented reading and writing incumbent solutions in the ParaSCIP solver
- Solver adapter does not crash
- DDBNB and Everest were modified to allow exchanging incumbents along with decision variable values
- Solving time is reduced for some of the instances and there is no slowdown for others

Thank you!

Sergey Smirnov,
IITP RAS (Kharkevich Institute)
sasmir@gmail.com