

The 6th International Conference
“Distributed Computing and Grid-technologies
in Science and Education”

**Customizable system for coarse-grained
parallel schemes in branch-and-bound
algorithm for discrete optimization**

Sergey Smirnov
IITP RAS

Задача линейного программирования

$$f(x) = \sum_{j=1}^n c_j x_j = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \rightarrow \min$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \quad (i = 1, 2, \dots, m)$$

$$x_j \geq 0 \quad (j = 1, 2, \dots, n)$$

Частично целочисленная задача:

часть $X_i \in Z$

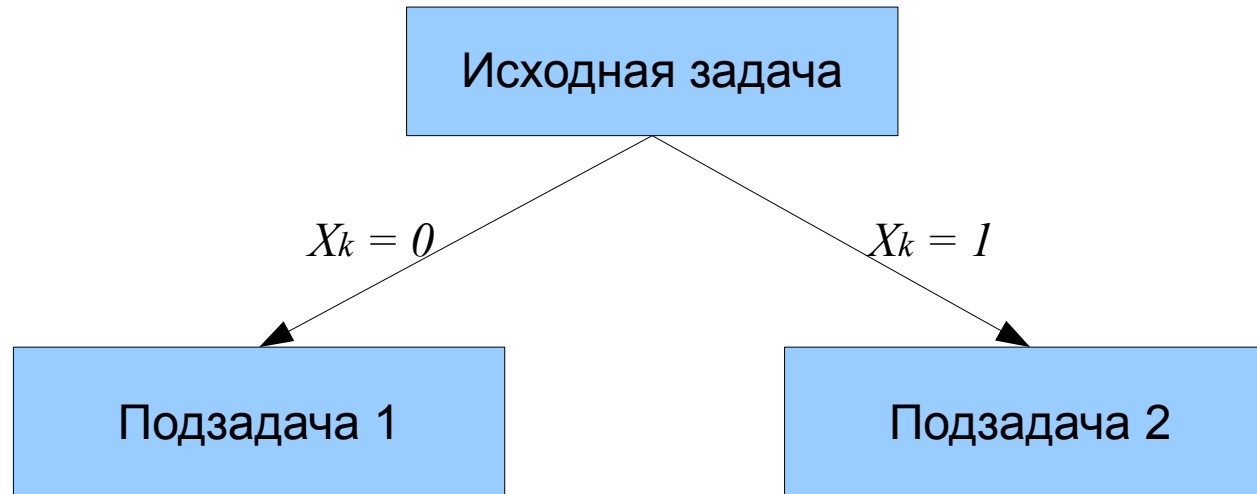
Решается вариантами метода ветвей и границ

Общие сведения о пакете CBC

- Cbc (Coin-or branch and cut) — пакет для решения частично целочисленных задач линейного программирования с открытым исходным кодом
- Может использоваться и как библиотека, и как автономное приложение.
- Как автономное приложение, принимает данные в форматах MPS, AMPL и др.
- <https://projects.coin-or.org/Cbc>

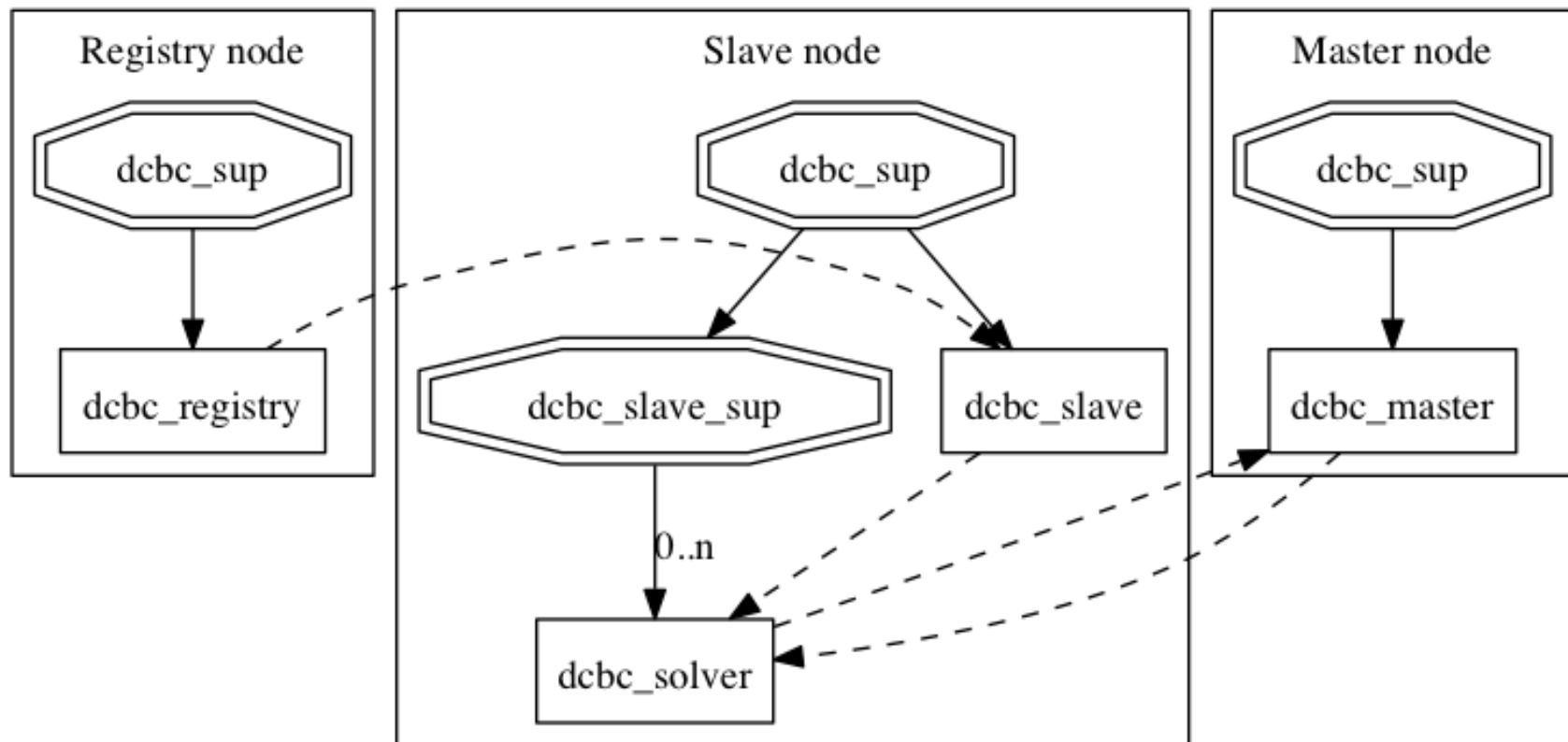
Деление на подзадачи

- Исходная задача задана в виде AMPL-стаба
- На ее основе генерируем набор подзадач, фиксируя отдельные целочисленные переменные
- Деление было реализовано в виде отдельного приложения `nlmod`
- Если зафиксировать M переменных, получаем 2^M подзадач

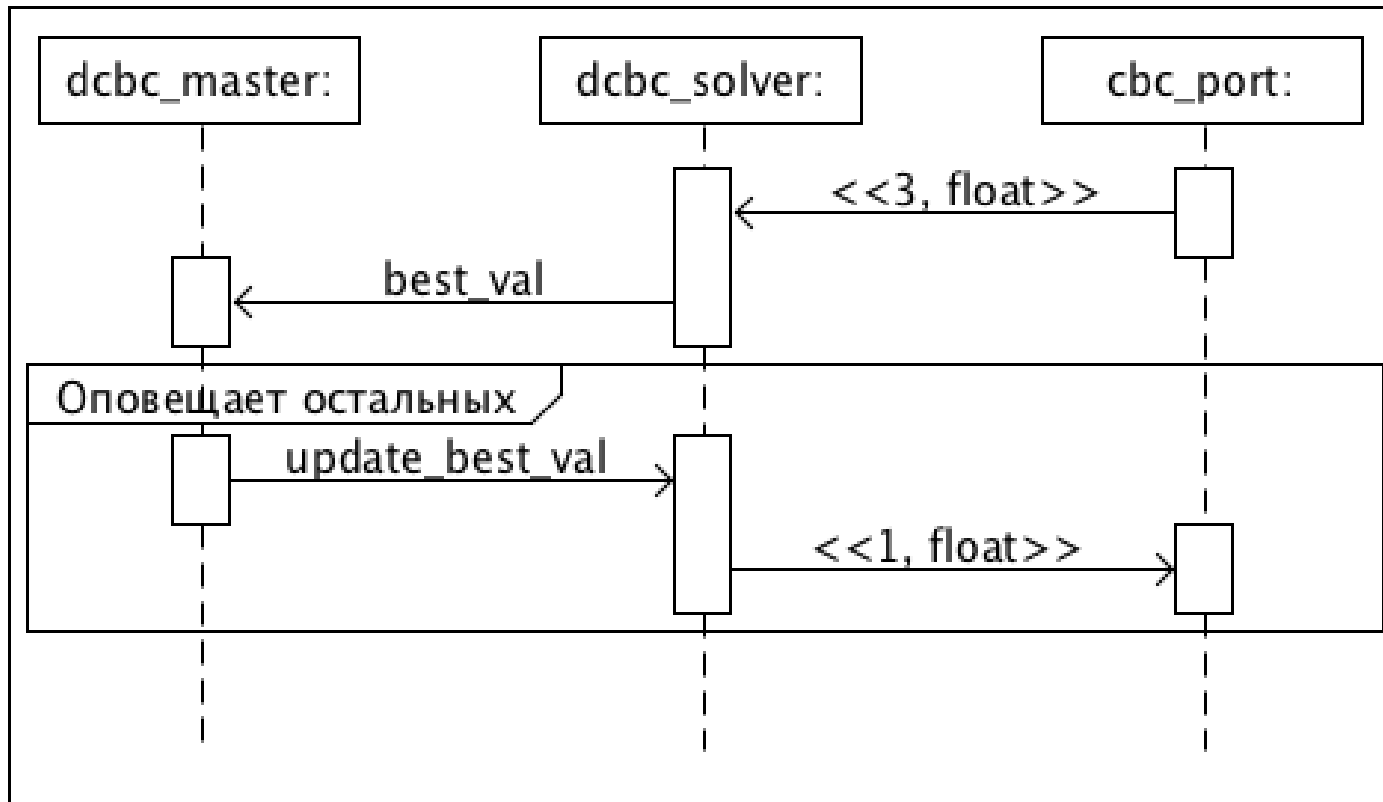


Distributed CBC

- Управление решением реализовано распределенным приложением на Erlang
- Решение подзадач ведут однопоточные экземпляры CBC
- <https://github.com/ssmir/dcbc>



Обмен рекордами



Вычислительные эксперименты

На базе задачи о коммивояжере с N городами

M — число переменных, зафиксированных в подзадачах

T — время, затраченное на решение задачи

T_n — если делили $n! \bmod$, T_a — если делили в AMPL

M = 0		
N	T, мин.	Ts, мин.
80	5,3	1,6
90	20,3	6,0
100	623,8	1,0
110	>10000	74,5

N = 90	
M	T, мин.
0 (1)	20,3
2 (4)	16,0
3 (8)	4,4
4 (16)	8,5
5 (32)	8,6

N = 100	
M	T, мин.
0 (1)	623,8
3 (8)	201,9
6 (64)	1627,0

N	M	T_n , мин.	T_a , мин.
80	4 (16)	1,5	1,9
90	5 (32)	8,6	11,1
100	6 (64)	1627,0	228,9
110	7 (128)	?	1211,6

Указанные времена работы получены на основе единичных запусков

Вычислительные средства:

8 потоков на 2 x Intel Xeon E5620 @ 2.40GHz

4 потока на Intel Core i7-2600K @ 3.40GHz

Выводы

- В случае задачи коммивояжера, DCVC позволяет получить заметное ускорение по сравнению с однопоточным SVC
- DCVC позволяет решать задачи, на которых «падает» многопоточный SVC
- Задачу коммивояжера DCVC решает медленнее однопоточного SCIP

Спасибо за внимание!
Вопросы?

Сергей Смирнов