What’s New in Gurobi 6.5

GUROBI
OPTIMIZATION

Gurobi 6.5 – Overview

- A quick review of some “simpler” enhancements
- A more extensive look at several addition enhancements
  - Python modeling
  - Variable hints
  - API recorder and replay
  - Gurobi Instant Cloud
  - MIQCP: what’s behind the big performance improvements
- **Performance Improvements** (with an emphasis on MIP performance)
  - Internal benchmarks
  - External (competitive) benchmarks

Some “Simpler” Enhancements

- APIs
  - IIS support in MATLAB
  - R interface extensions
- Licensing
  - Password protection for token servers
  - Single-use licenses without a token server
  - New command to provide location of current license file
    - `gurobi_cl --license`
- Distributed
  - Distributed MIP logging
- Platforms
  - Support for clang++ on Mac (libc++)
  - Support for Visual Studio 2015
  - Compute Server encryption routines moved to a separate library
- Other
  - BarX attribute to query the best barrier iterate
  - OPB file format reader

Several Additional Enhancements

Python Modeling

- Expression building is more than 4X faster

- Lazy update mode
  - Set new `UpdateMode` parameter to 1
  - Refer to new variables and constraints without calling `Model.update()`

- More control over when Gurobi environments are created/released
  - Default environment not created until first used
  - Released with new `disposeDefaultEnv()` method

- Interactive examples for commonly faced business problems
  - http://www.gurobi.com/resources/examples/example-models-overview
  - http://examples.gurobi.com/facility-location/

Variable Hints

- Provide hints to the solver about which variable should take which value
- Guides heuristics and branching
- VarHintVal attribute:
  - Specifies a value for a variable
- VarHintPri attribute:
  - Specifies a level of confidence in this particular variable value
- Comparison to MIP starts:
  - MIP start is used to provide an initial feasible solution to the solver
    - Is evaluated prior to starting the solution process
    - Provides incumbent if feasible
    - Does not influence solution process if it is not feasible
  - Variable Hints guide the search
    - High quality hints should lead to a high quality solution quickly
      - Either through heuristics or through branching
    - Affects the whole solution process

API Recorder

- Setting `Record` parameter to `1` will produce `recording000.grbr` file
  - Tracks all Gurobi API calls
- Use `gurobi_cl recording000.grbr` to replay execution
  - Replay Gurobi execution independently from your own application
- Use cases:
  - Debug performance issues
    - Measures time spent in API calls (e.g., model building) and algorithms (solving)
  - Identify cases where your program leaks Gurobi models or environments
    - Lists number of models and environments that were never freed by your program
  - Relay exact sequence of commands your program issues to Gurobi
    - Assists technical support in case you run into a question or issue that is difficult to reproduce
    - Just send recording file, instead of having to send the whole application

Gurobi Instant Cloud

Gurobi Cloud Offering

- Gurobi has had a Cloud offering for over 5 years
  - Pay for just what you use
  - No software or hardware to purchase or configure
  - Ideal for short term or sporadic use, or irregular/large peak usage

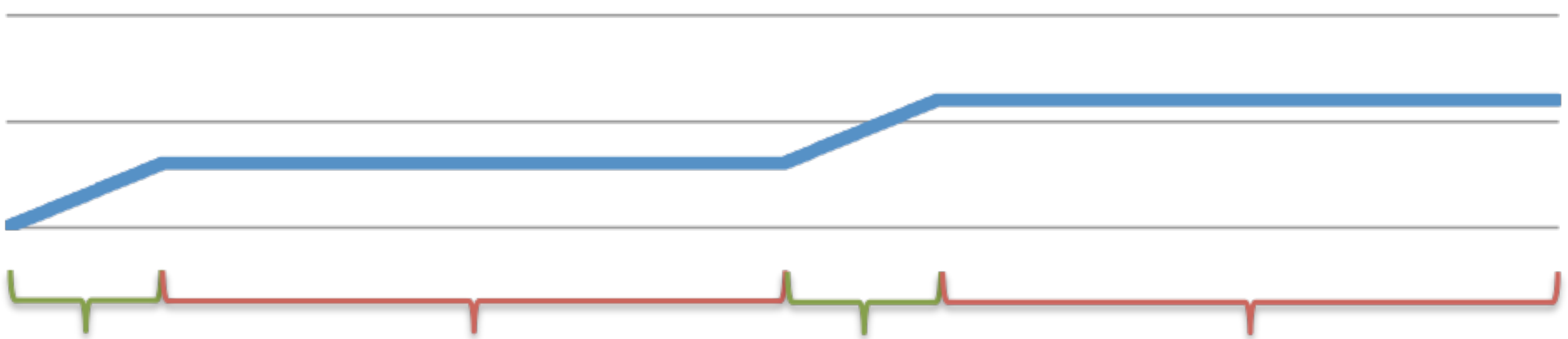
- What's new with the Gurobi Instant Cloud?
  - Simplified pricing and a reduced cost per hour
  - Much easier to launch and use Cloud instances

New Cloud Pricing Model

	Light License	Full License
Hourly Price	$10/hr	$20/hr
Monthly Price Cap	$1500	$3000

- Price cap lets you run a machine 24 hrs/day for ~750hrs in a month

Gurobi License cost



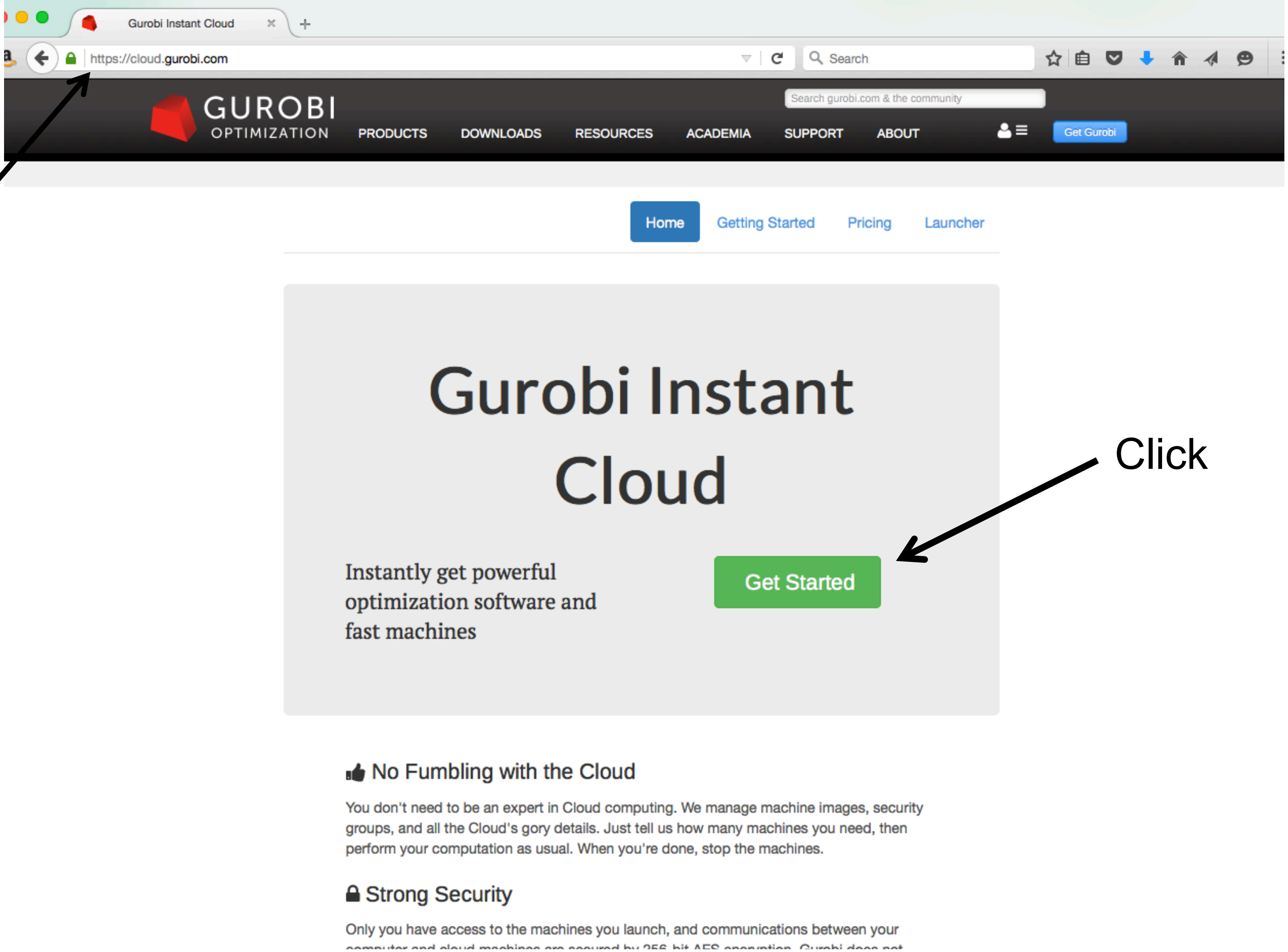
The first 150 hours are billed based on usage

The next 600 hours, until 750 hours, are included at no charge

The next 150 hours are billed based on usage

The next 600 hours, until 1500 hours, are included at no charge

Using the Instant Cloud: `cloud.gurobi.com`



Gurobi Instant Cloud

Instantly get powerful optimization software and fast machines

No Fumbling with the Cloud

You don't need to be an expert in Cloud computing. We manage machine images, security groups, and all the Cloud's gory details. Just tell us how many machines you need, then perform your computation as usual. When you're done, stop the machines.

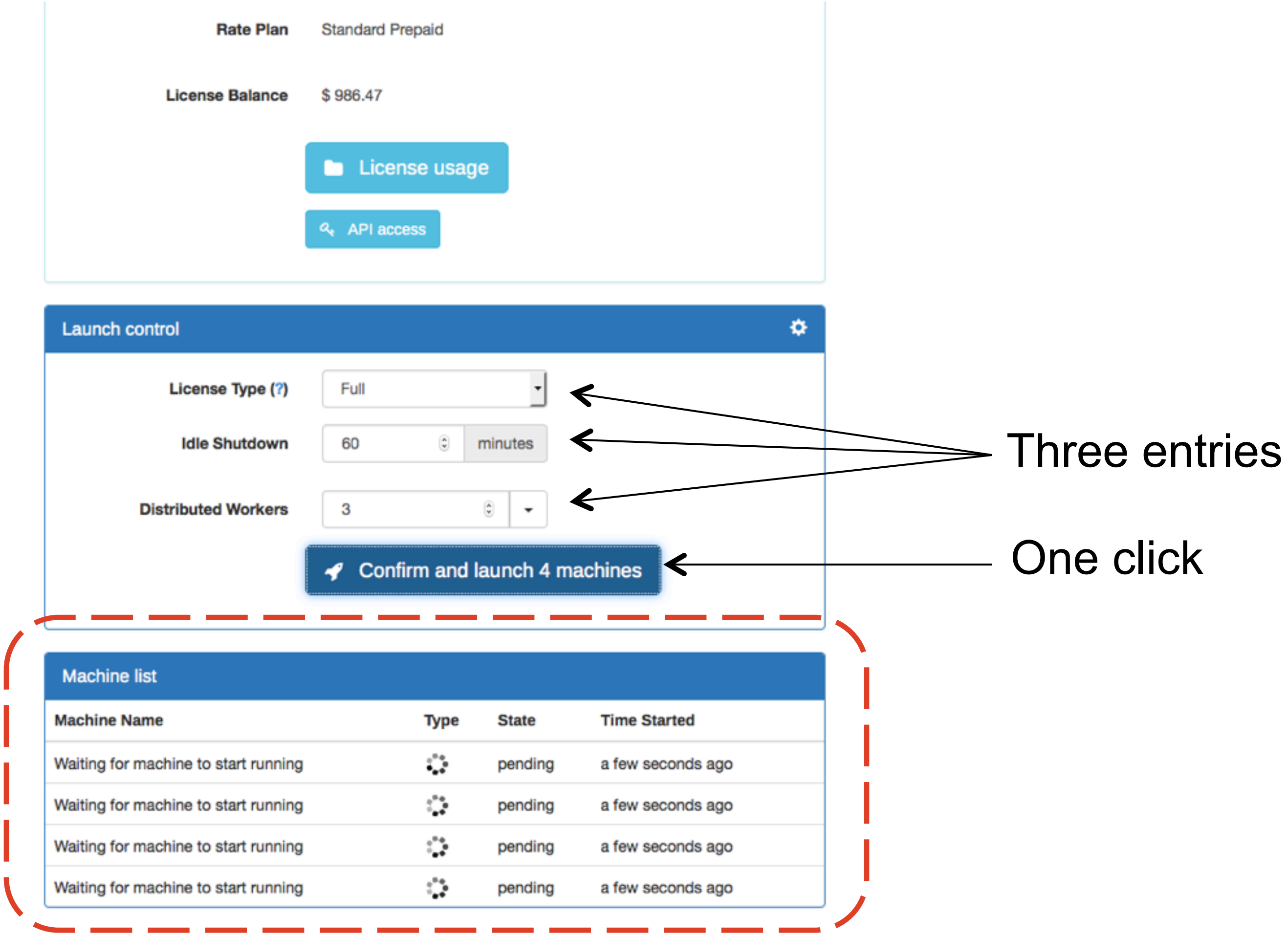
Strong Security

Only you have access to the machines you launch, and communications between your computer and cloud machines are secured by 256-bit AES encryption. Gurobi does not

URL

Click

Instantiating Machines



Rate Plan: Standard Prepaid

License Balance: $ 986.47

License Type (?): Full

Idle Shutdown: 60 minutes

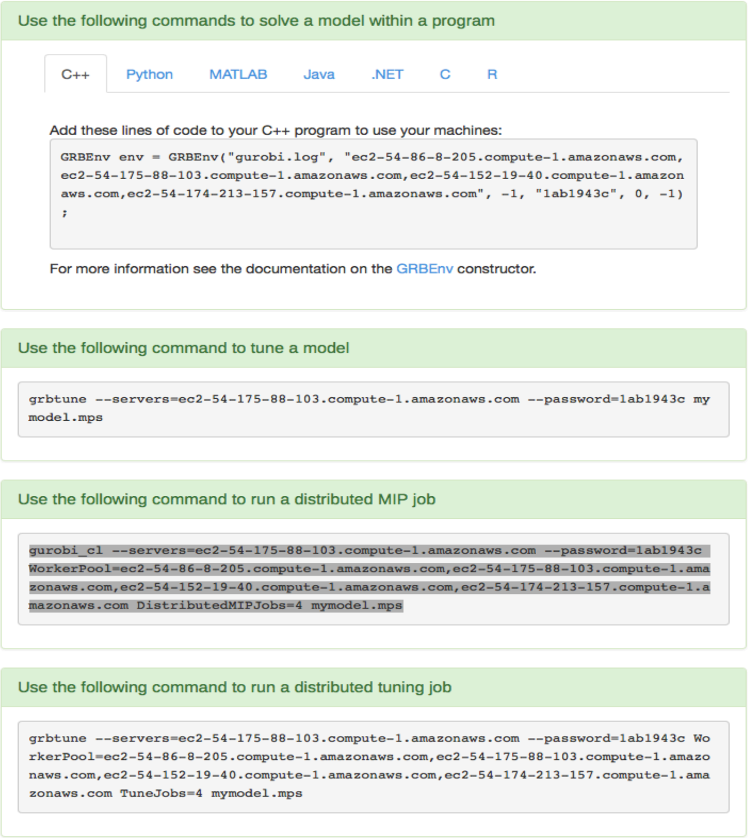
Distributed Workers: 3

Machine Name	Type	State	Time Started
Waiting for machine to start running		pending	a few seconds ago
Waiting for machine to start running		pending	a few seconds ago
Waiting for machine to start running		pending	a few seconds ago
Waiting for machine to start running		pending	a few seconds ago

Three entries

One click

Solving a Model



Use the following commands to solve a model within a program

Add these lines of code to your C++ program to use your machines:

```
GRBEnv env = GRBEnv("gurobi.log", "ec2-54-86-8-205.compute-1.amazonaws.com,
ec2-54-175-88-103.compute-1.amazonaws.com,ec2-54-152-19-40.compute-1.amazon
aws.com,ec2-54-174-213-157.compute-1.amazonaws.com", -1, "1ab1943c", 0, -1)
;
```

For more information see the documentation on the GRBEnv constructor.

Use the following command to tune a model

```
grbtune --servers=ec2-54-175-88-103.compute-1.amazonaws.com --password=1ab1943c my
model.mps
```

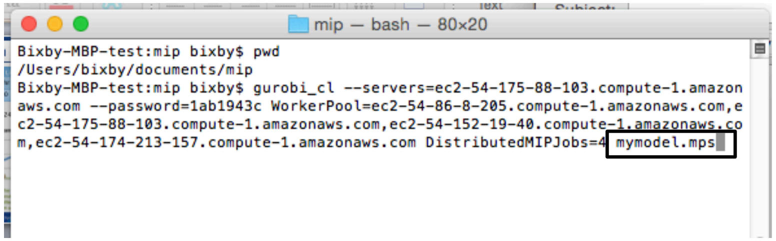
Use the following command to run a distributed MIP job

```
gurobi_cl --servers=ec2-54-175-88-103.compute-1.amazonaws.com --password=1ab1943c
WorkerPool=ec2-54-86-8-205.compute-1.amazonaws.com,ec2-54-175-88-103.compute-1.ama
zonaws.com,ec2-54-152-19-40.compute-1.amazonaws.com,ec2-54-174-213-157.compute-1.a
mazonaws.com DistributedMIPJobs=4 mymodel.mps
```

Use the following command to run a distributed tuning job

```
grbtune --servers=ec2-54-175-88-103.compute-1.amazonaws.com --password=1ab1943c Wo
rkerPool=ec2-54-86-8-205.compute-1.amazonaws.com,ec2-54-175-88-103.compute-1.amazo
naws.com,ec2-54-152-19-40.compute-1.amazonaws.com,ec2-54-174-213-157.compute-1.ama
zonaws.com TuneJobs=4 mymodel.mps
```

Copy predefined command



```
Bixby-MBP-test:mip bixby$ pwd
/Users/bixby/documents/mip
Bixby-MBP-test:mip bixby$ gurobi_cl --servers=ec2-54-175-88-103.compute-1.amazon
aws.com --password=1ab1943c WorkerPool=ec2-54-86-8-205.compute-1.amazonaws.com,e
c2-54-175-88-103.compute-1.amazonaws.com,ec2-54-152-19-40.compute-1.amazonaws.co
m,ec2-54-174-213-157.compute-1.amazonaws.com DistributedMIPJobs=4 mymodel.mps
```

Paste into a terminal

MIQCP

MIQCP

- Mittelmann benchmarks (> 1.0 means Gurobi is faster)

Benchmark	CPLEX	XPRESS	MOSEK
MISOCP	2.46X	4.82X	10.5X

Where does the MIQCP performance come from?

Change	>1s				>100s			
	#	Wins	Losses	Speedup	#	Wins	Losses	Speedup
cone disaggr.	116	40	0	1.79x	25	16	0	4.43x
branching thr.	106	30	6	1.39x	24	10	6	2.34x
impr. presolve	114	12	2	1.16x	34	5	0	1.40x
impr. OA cuts	110	48	25	1.51x	46	27	7	2.30x

- Cone disaggregation
  - See Vielma, Dunning, Huchette, Lubin (2015)
- Branching threshold
  - Changed 10^{-5} to 10^{-8}
- Improved presolve
  - Detect one particular structure to improve bound strengthening
- Improved outer approximaxtion cuts
  - See Günlük and Linderoth (2011)

Cone Disaggregation

- Reformulation
  - Replace the large cone

    $x_1^2 + x_2^2 + \ldots + x_n^2 \leq y^2$ (1)

    $y \geq 0$

    with

    $x_i^2 \leq z_i y, \;\; i = 1, 2, \ldots, n$ (2) n rotated cones

    $z_1 + z_2 + \ldots + z_n \leq y$ (3) 1 linear constraint

    $y \geq 0, z_1, z_2, \ldots, z_n \geq 0$ n new variables

- See Vielma, Dunning, Huchette, Lubin (2015) on Extended Formulations

Gurobi 6.5 Performance Benchmarks

GUROBI
OPTIMIZATION

Two Kinds of Benchmarks

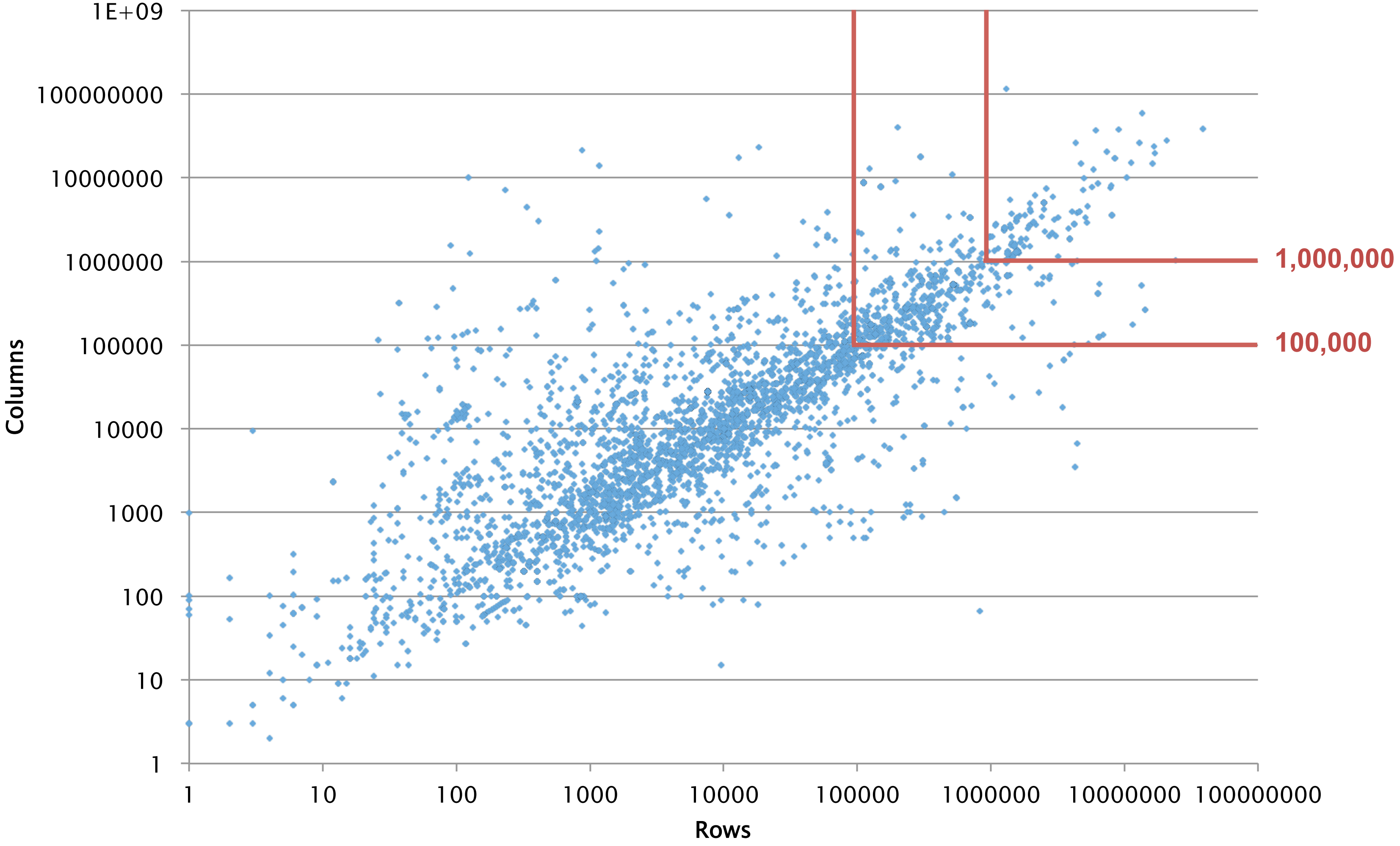
- Internal benchmarks
  - Most important: compare Gurobi version-over-version
  - Based on internal Library of some 10,000 models

- External, competitive benchmarks
  - Conducted by Hans Mittelmann, Arizona State University
    - http://plato.asu.edu/bench.html
  - For MIP largely based upon MIPLIB 2010

Internal Benchmarks

Gurobi MIP Library

(3842 models)



Performance Improvements in Gurobi 6.5

Problem Class	>1s				>100s			
	#	Wins	Losses	Speedup	#	Wins	Losses	Speedup
LP: concur.	409	122	50	1.16x	148	55	30	1.32x
primal	390	111	66	1.03x	169	61	34	1.02x
dual	362	103	57	1.11x	131	47	27	1.24x
barrier	408	120	41	1.12x	133	61	23	1.25x
QCP/SOCP	78	15	11	1.16x	20	6	2	1.97x
MILP	1851	982	407	1.37x	832	514	215	1.72x
MIQP	95	57	22	1.99x	36	24	9	4.98x
MIQCP	190	130	44	2.78x	86	67	17	6.49x

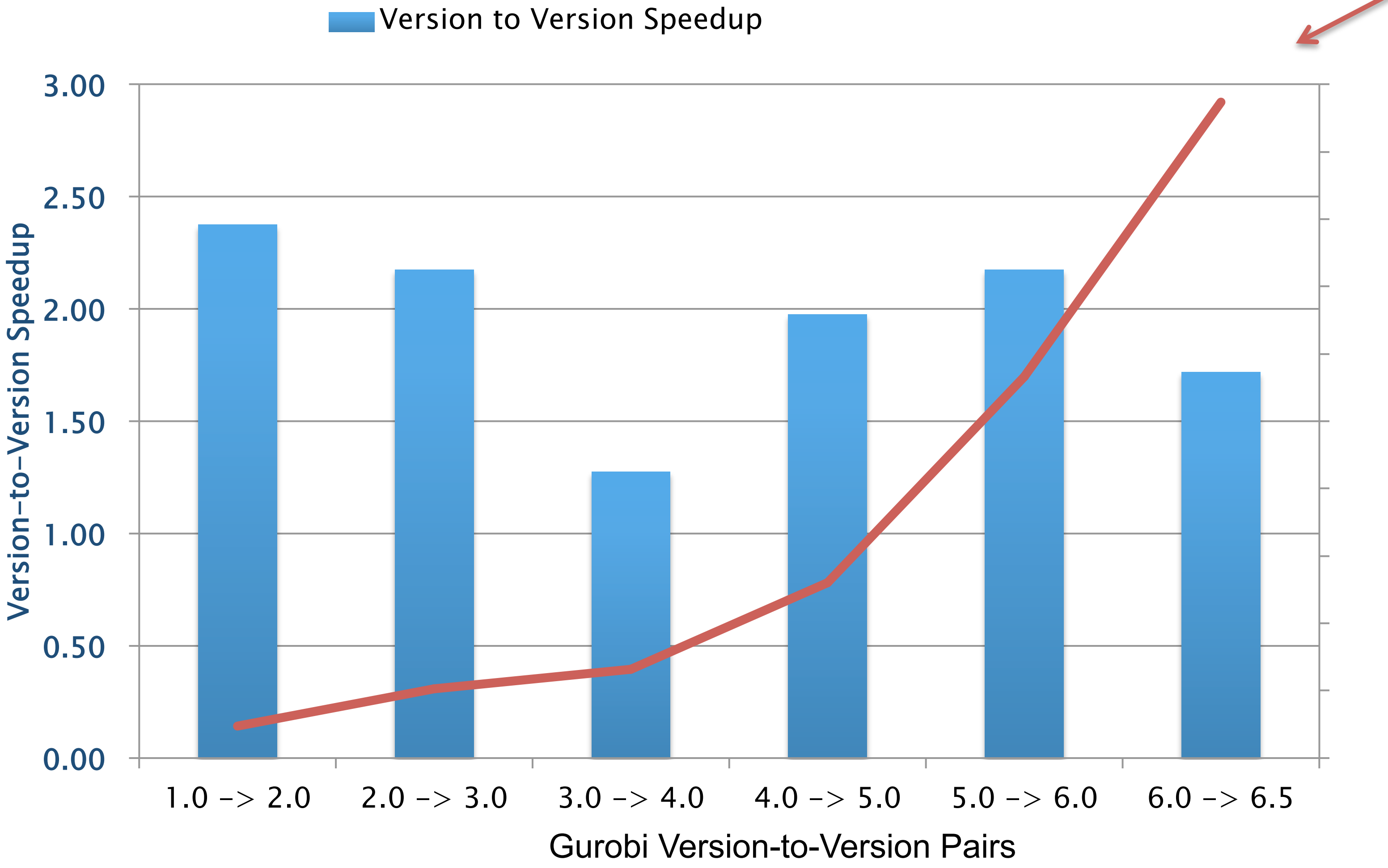
- Gurobi 6.0 vs. 6.5: > 1.00x means that Gurobi 6.5 is faster than Gurobi 6.0
- MIQP: big speedup from 5 models of a single source
- QP not included: only 16 models > 1s

Performance Improvements in Gurobi 6.5

Problem Class	>1s				>100s			
	#	Wins	Losses	Speedup	#	Wins	Losses	Speedup
LP	409	122	50	1.16x	148	55	30	1.32x
primal	390	111	66	1.03x	169	61	34	1.02x
dual	362	103	57	1.11x	131	47	27	1.24x
barrier	408	120	41	1.12x	133	61	23	1.25x
QCP/SOCP	78	15	11	1.16x	20	6	2	1.97x
MILP	1851	982	407	1.37x	832	514	215	1.72x
MIQP	95	57	22	1.99x	36	24	9	4.98x
MIQCP	190	130	44	2.78x	86	67	17	6.49x

- Gurobi 6.0 vs. 6.5: > 1.00x means that Gurobi 6.5 is faster
- MIQP: big speedup from 5 models of a single source
- QP not included: only 16 models > 1s

Gurobi Keeps Getting Better



48.7x improvement (six years)

Where do these improvements come from?

A taxonomy of improvements: Part I

Cuts		24.1%
Improved MIR aggregation	11.2%	
Improved node cut selection	5.1%	
More sophisticated root-cut filtering and abort criterion	4.1%	
More aggressive implied-bound cuts	1.2%	
More aggressive sub-MIP cuts	0.8%	
Presolve		15.6%
Improvements in probing	7.0%	
Improved sparse presolve	3.8%	
Merging parallel integer columns with arbitrary scalars	1.4%	
Disconnected components in presolve	1.3%	
More stable non-zero cancellation	0.7%	
Aggregating symmetric continuous variables	0.6%	
Branching		7.7%
Replaced 10-5 threshold by 10-8	2.6%	
Follow-on branching	2.4%	
Using reduced costs as pseudo-costs	1.3%	
Modified threshold in implications-based tie-breaking	1.2%	

A taxonomy of improvements: Part II

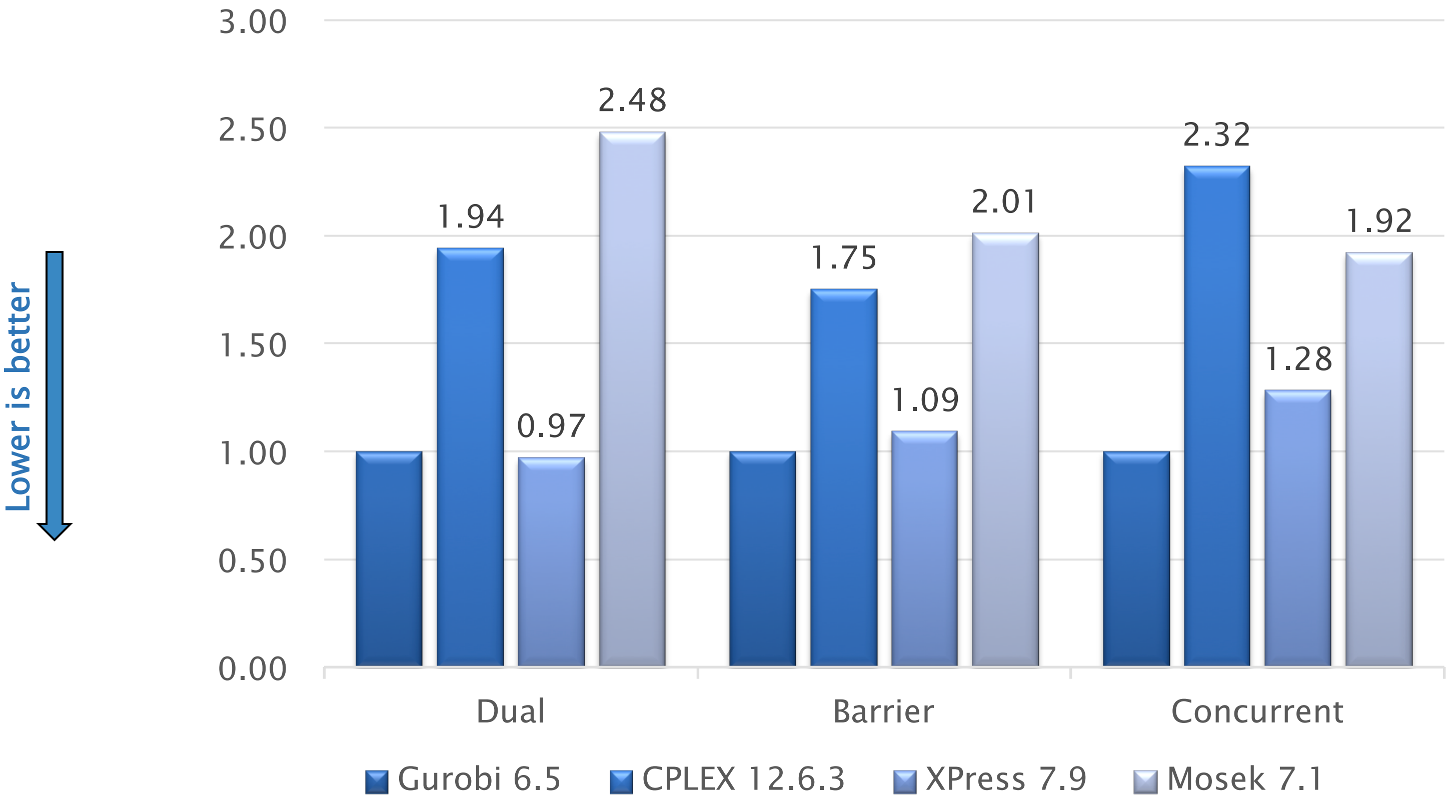
MIP/LP integration		7.5%
◦ Adjusting pi to get stronger reduced costs	3.8%	
◦ Improvements in simplex pricing	3.6%	
Heuristics		3.5%
◦ New heuristic running in parallel to the root node	2.4%	
◦ Randomization in fix-and-dive heuristics	1.1%	
Node presolve		3.9%
◦ Improved conflict analysis	1.8%	
◦ More node bound strengthening	1.4%	
◦ Slightly faster propagation	0.7%	
Compiler		2.0%
◦ Switched to Intel 2016 compiler	2.0%	

External Benchmarks

(Hans Mittelmann: http://plato.asu.edu/bench.html)

LP Benchmarks

LP Solve Times



- Complete test data available here: http://plato.asu.edu/ftp/lpcom.html
- 44 models, time limit 25000 sec., 8 threads

MILP Benchmarks

Mittelmann (MIPLIB 2010): MIP Solve Times

- Gurobi 6.5 vs. Competition: Solve times (> 1.0 means Gurobi faster)

Benchmark	CPLEX				XPRESS			
	P=1	P=4	P=12	P=32	P=1	P=4	P=12	P=32
Optimality	1.16X	1.45X	1.11X	1.10X	1.32X	1.25X	1.36X	1.33X
Feasibility	–	1.14X	–	–	–	4.62X	–	–
Infeasibility	–	0.98X	–	–	–	1.67X	–	–
"Easy" Optimality	–	–	1.11X	–	–	–	1.99X	–

Mittelmann (MIPLIB 2010): MIP Solve Times

- Gurobi 6.5 vs. Competition: Solve times (> 1.0 means Gurobi faster)

Benchmark	CPLEX				XPRESS			
	P=1	P=4	P=12	P=32	P=1	P=4	P=12	P=32
Optimality	1.21X				1.34X			
Feasibility	–	1.14X	–	–	–	4.62X	–	–
Infeasibility	–	0.98X	–	–	–	1.67X	–	–
“Easy” Optimality	–	–	1.11X	–	–	–	1.99X	–

Mittelmann: MILP Solvability

Gurobi 6.5 vs. Competition: Hit time limit

Benchmark	Gurobi	CPLEX	XPRESS
Optimality	2	4	3
Feasibility	0	0	3
Infeasibility	0	0	0
"Easy" -- Optimality	11	12	39

QP Benchmarks

Mittelmann: QP Solve Times

- Gurobi 6.5 vs. Competition: Solve times
  (> 1.0 means Gurobi is faster)

Benchmark	CPLEX	XPRESS	MOSEK
MIQP	1.33X	1.36X	–
MIQCP	1.47X	1.61X	–
MISOCP	2.46X	4.82X	10.5X
SOCP	2.32X	1.22X	0.97X

Q&A

GUROBI
OPTIMIZATION

Thank You

For more information please contact
info@gurobi.com

GUROBI
OPTIMIZATION