



Link-Time Optimization on PlayStation®4

Yunzhong Gao

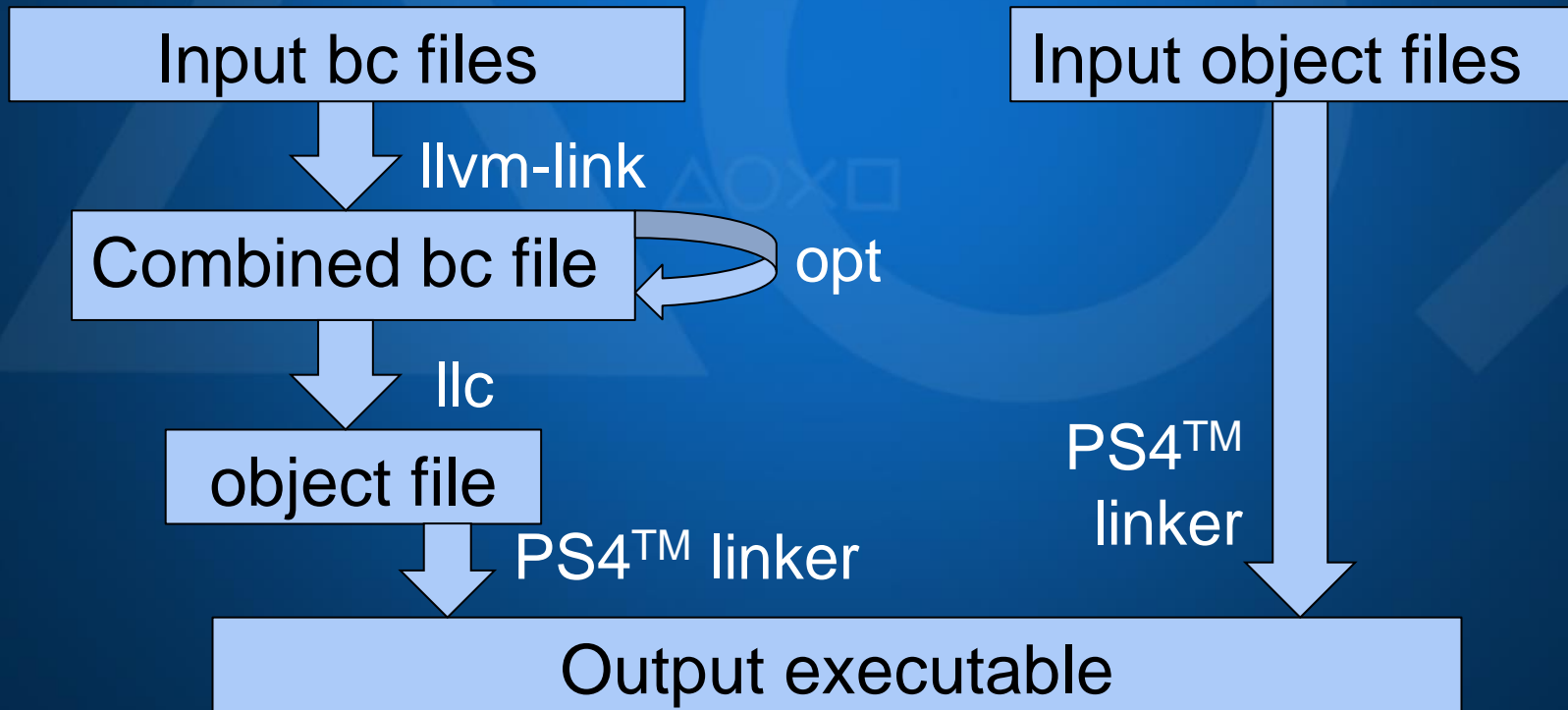
Sony Computer Entertainment

LLVM Dev Meeting, 28-29 Oct 2014



Last Year...

- LTO without linker





Past work in LTO...

- LOTS of contribution from the community

- Thank you!



- Reference: see Rafael's talk at Euro LLVM 2014



In the Linker...

- Proprietary linker



In the Linker...

- Proprietary linker
- Uses tools/LTO interface
 - `lto_module_is_object_file()`



In the Linker...

Input bc files

Input object files



In the Linker...

Input bc files

Input object files

Scan Phase

for each bc file:

- lto_module_get_symbol_name()
- lto_module_get_symbol_attribute()
- lto_module_get_deplib()

- lto_codegen_add_module()



In the Linker...

Input bc files

Input object files

Scan Phase





In the Linker...

Input bc files

Input object files

Scan Phase

Compile Phase

- `lto_codegen_add_must_preserve_symbol()`
- `lto_codegen_compile_to_file()`



In the Linker...

Input bc files

Scan Phase

Compile Phase

Input object files



In the Linker...

Input bc files

Input object files

Scan Phase

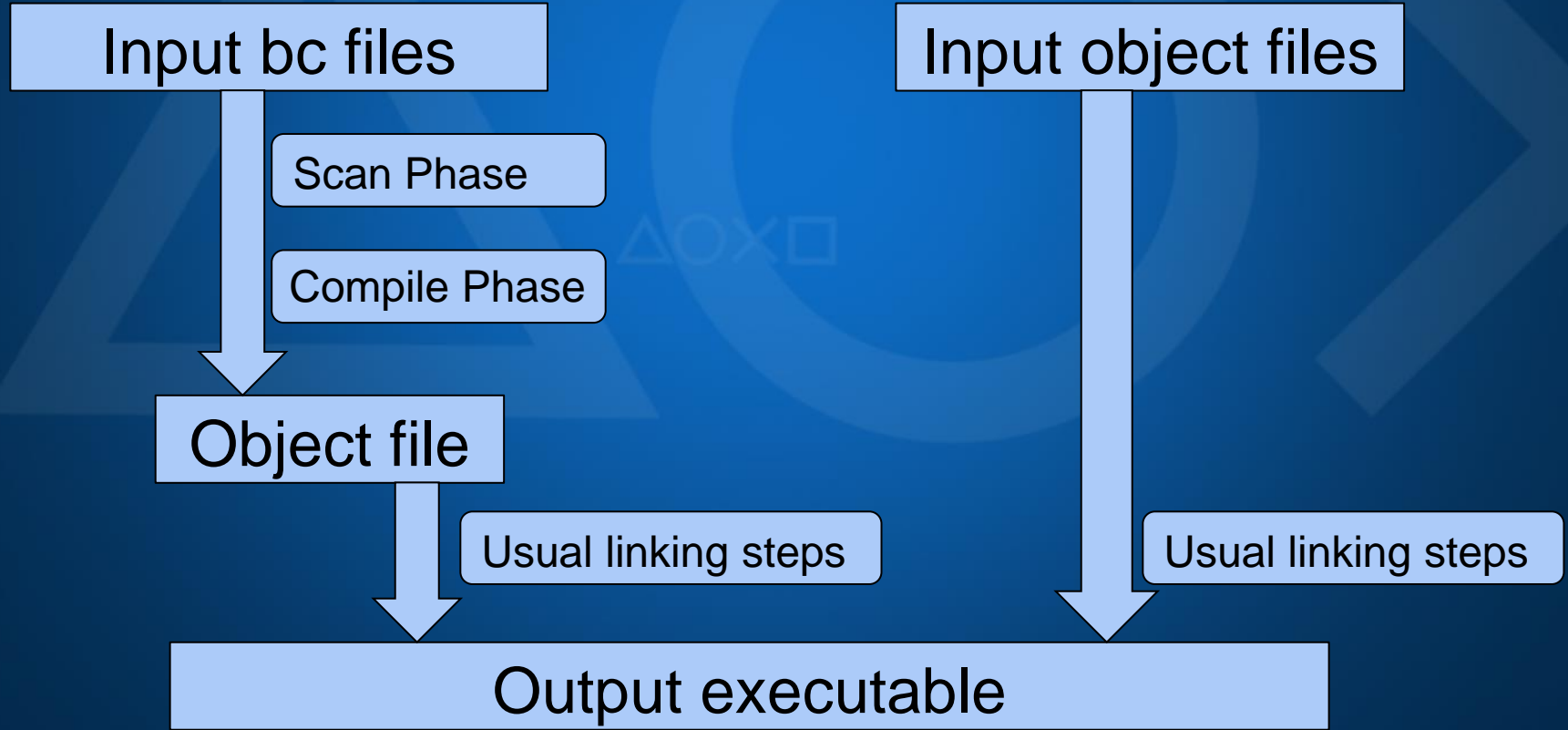
Compile Phase

Object file





In the Linker...





Whose Bug?

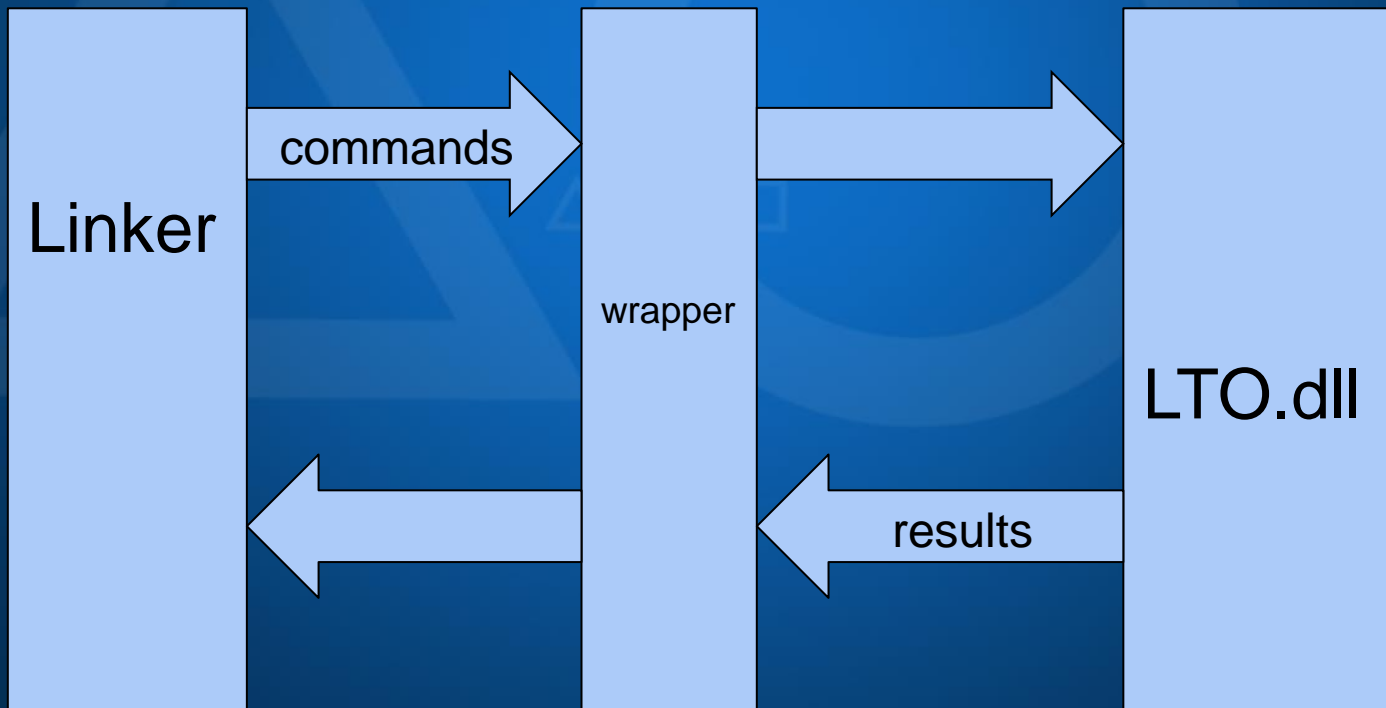
Clearly, this is a bug
in the LTO library.



No, **CLEARLY**, it is
a linker problem.

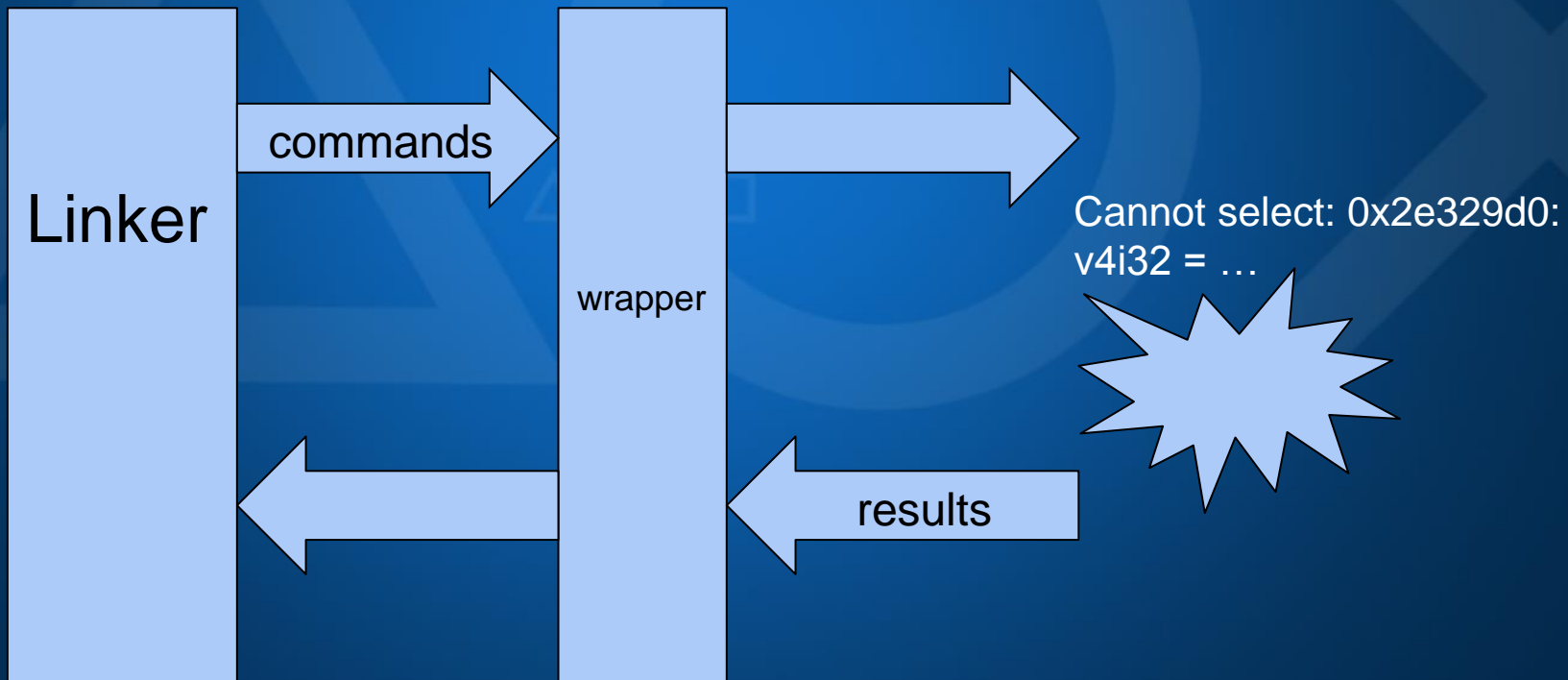


Whose Bug?



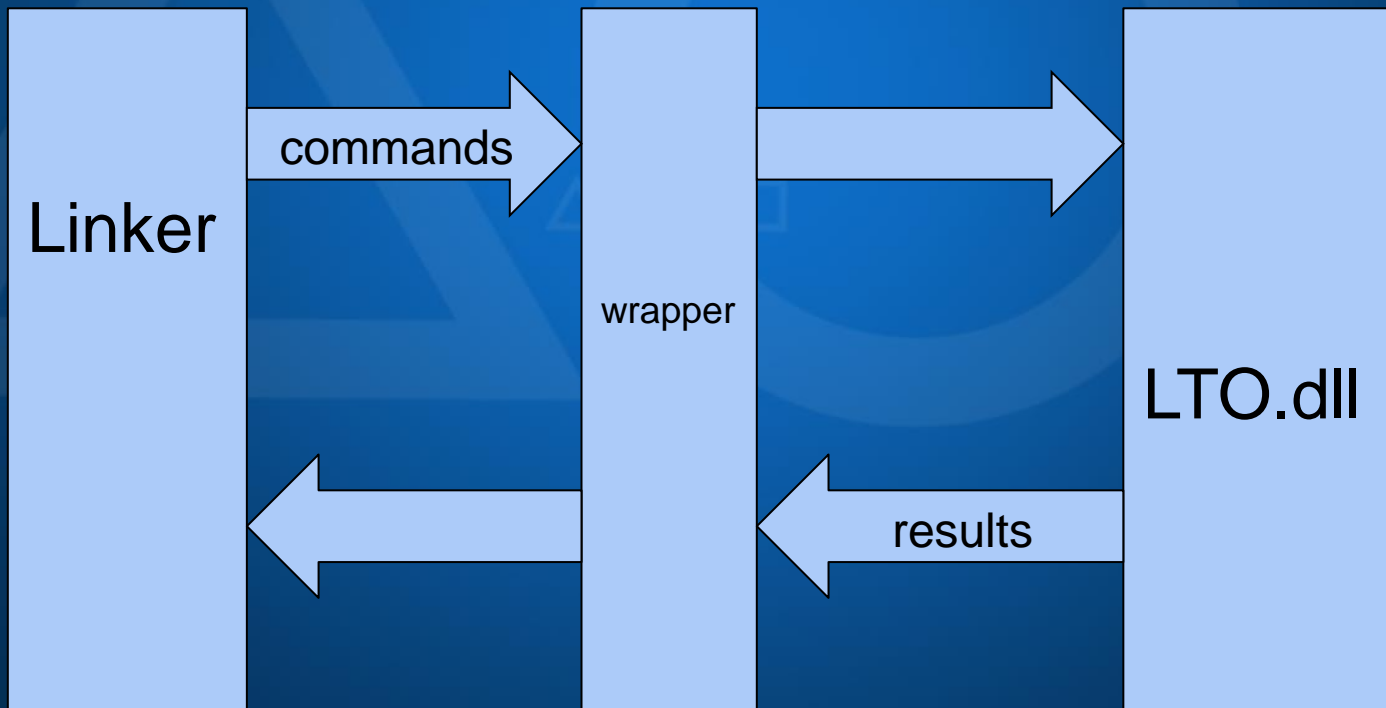


Whose Bug?



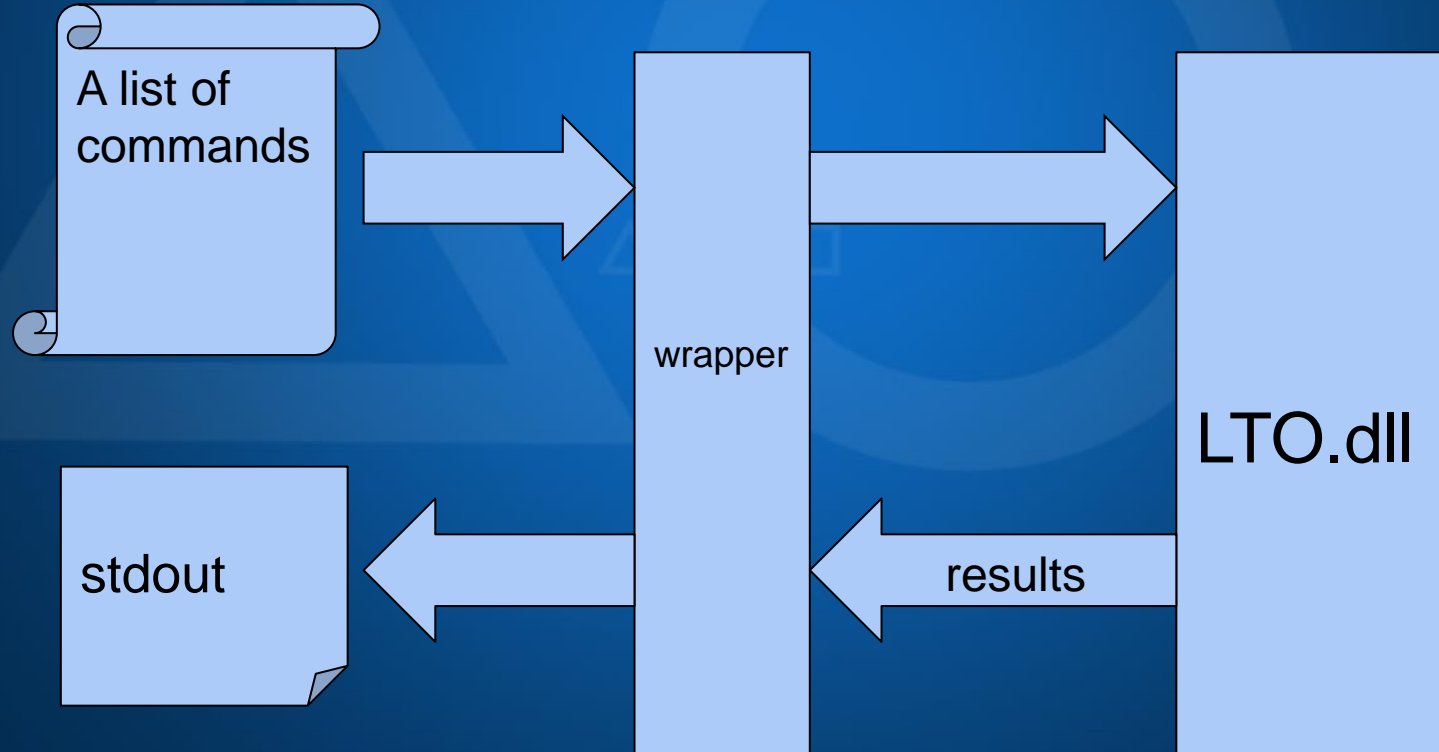


Whose Bug?





Whose Bug?





Evaluation

Results on some PS4™ game titles:

	Game#1	Game#2	Game#3	Game#4	Game#5
Run-Time Performance	Improved 3.39%	Improved 1.43%	Improved 9%	Improved 6.11%	Improved 6%



Evaluation

Results on some PS4™ game titles:

	Game#1	Game#2	Game#3	Game#4	Game#5
Run-Time Performance	Improved 3.39%	Improved 1.43%	Improved 9%	Improved 6.11%	Improved 6%

- Inliner
- Internalize Global Symbols
- Inter-Procedural Sparse Conditional Constant Propagation



Evaluation

Results on some PS4™ game titles:

	Game#1	Game#2	Game#3	Game#4	Game#5
Executable Size	Increased 0.242%	Increased 0.539%	Increased 2%	Decreased 11.8%	Increased 2.06%



Evaluation

Results on some PS4™ game titles:
(without debug info)

	Game#1	Game#2	Game#3	Game#4	Game#5
Build Time (LTO vs non-LTO)	4x	3x	5x	2.5x	10x



Related Work



Related Work

- LTO with debug info: memory usage and build time



Related Work

- LTO with debug info: memory usage and build time
- Versioning of bc files



Related Work

- LTO with debug info: memory usage and build time
- Versioning of bc files
- Further performance improvement in LTO:
 - Move some optimization passes from compiler to LTO? (e.g. loop-unroll, vectorizer)