Technology Innovation Institute

FSE|24
15th - 19th July 2024
Porto De Galinhas, Brazil

# TACE

## Rapit Taint Assisted Concolic Execution

**Ridhi Jain**
(ridhi.jain@tii.ae)
(Presenter)

**Norbert Tihanyi**
(norbert.tihanyi@tii.ae)

**Mthandazo Ndhlovu**
(mthandazo.ndhlovu@tii.ae)

**Mohamed Amine Ferrag**
(mohamed.ferrag@tii.ae)

**Lucas Carvalho Cordeiro**
(lucas.cordeiro@manchester.uk)

# Why Software Testing?

# Why Software Testing?

Dec' 21


Sept' 22


Jan' 23

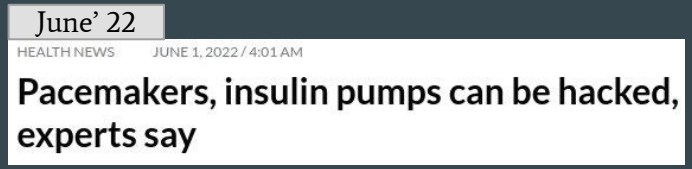

Feb' 23


May' 15


July' 13


Software crashes may not only cause heavy **monetary losses** but can also be **life threatening**.

June' 22




June' 22

# Motivation

[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.
[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.
[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

# Motivation

Fuzzing can be helpful.


[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

# Motivation

Fuzzing can be helpful.

Depends on Quality of input.


[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

# Motivation

Fuzzing can be helpful.

Depends on Quality of input.

Symbolic execution can improve improve input quality.


[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

# Motivation

Fuzzing can be helpful.

Depends on Quality of input.

Symbolic execution can improve improve input quality.

Interpretation

No. of constraints


[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

# Motivation

Fuzzing can be helpful.

Depends on Quality of input.

Symbolic execution can improve improve input quality.

Interpretation

No. of constraints

SymCC [1], SymQEMU[2]

Don’t Interpret, Compile!


[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

# Motivation

Fuzzing can be helpful.

Depends on Quality of input.

Symbolic execution can improve improve input quality.

Interpretation

No. of constraints

SymCC [1], SymQEMU[2]

Don’t Interpret, Compile!

No. of constraints


[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

# Motivation

Fuzzing can be helpful.

Depends on Quality of input.

Symbolic execution can improve improve input quality.

Interpretation

No. of constraints

SymCC [1], SymQEMU[2]

Don’t Interpret, Compile!

No. of constraints

LSym[3]

Constraint Debloating


[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

# Motivation

Fuzzing can be helpful.

Depends on Quality of input.

Symbolic execution can improve improve input quality.

Interpretation

No. of constraints

SymCC [1], SymQEMU[2]

Don’t Interpret, Compile!

No. of constraints

LSym[3]

Constraint Debloating

Manual Summary Creation

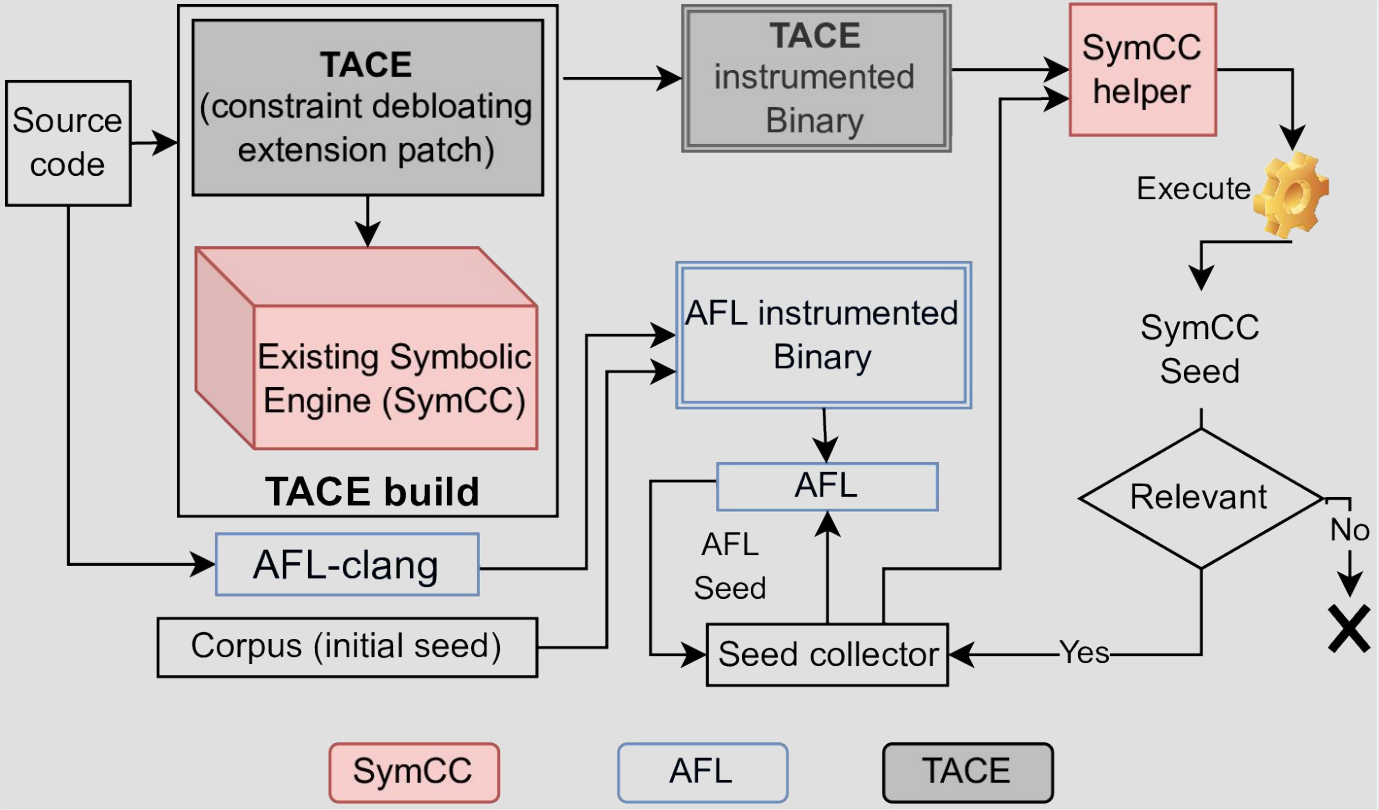
Unavailable


[1] Poeplau, Sebastian, and Aurélien Francillon. "Symbolic execution with SymCC: Don't interpret, compile!." *Proceedings of the 29th USENIX Conference on Security Symposium*. 2020.

[2] Poeplau, Sebastian, and Aurélien Francillon. "SymQEMU: Compilation-based symbolic execution for binaries." *NDSS*. 2021.

[3] Mi, Xianya, et al. "LeanSym: Efficient hybrid fuzzing through conservative constraint debloating." *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*. 2021.

# TACE - Hybrid Fuzzing Architecture



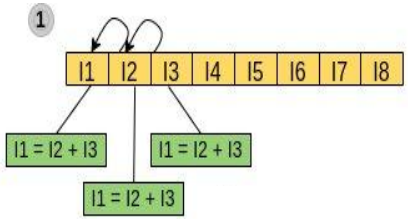

https://github.com/tacetool/TACE

# Taint Assisted Concolic Grouping

(Instruction 1) I1 = I2 + I3
(Instruction 2) I3 = I8
(Instruction 3) I4 = 2 * I6
(Instruction 4) I7 = I4 + I6

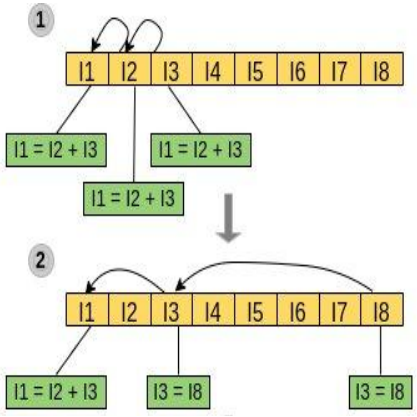
# Taint Assisted Concolic Grouping

(Instruction 1) I1 = I2 + I3
(Instruction 2) I3 = I8
(Instruction 3) I4 = 2 * I6
(Instruction 4) I7 = I4 + I6



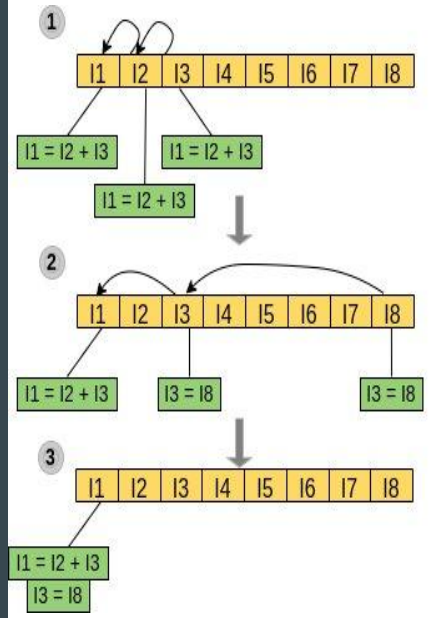
# Taint Assisted Concolic Grouping

(Instruction 1) I1 = I2 + I3
(Instruction 2) I3 = I8
(Instruction 3) I4 = 2 * I6
(Instruction 4) I7 = I4 + I6



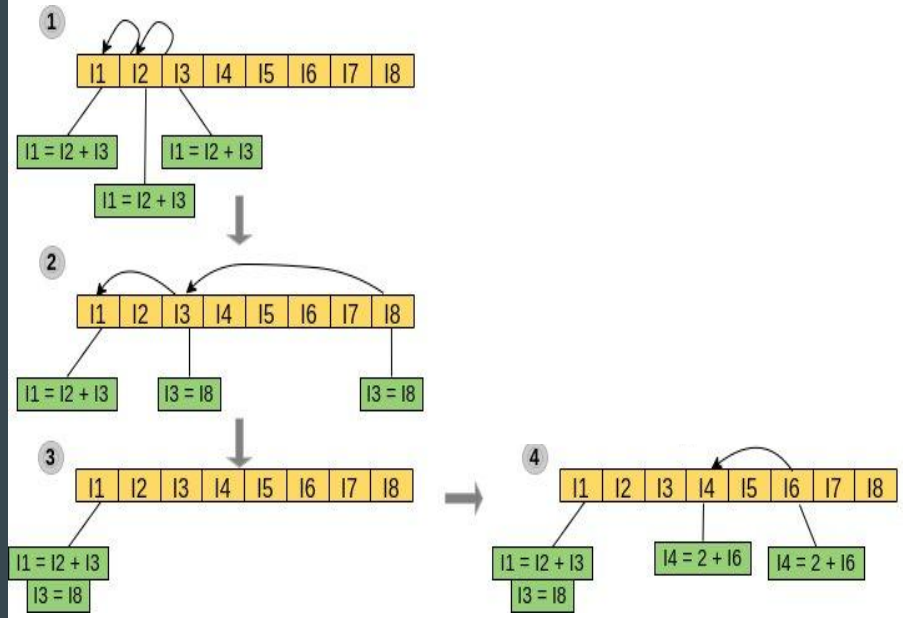
# Taint Assisted Concolic Grouping

(Instruction 1) I1 = I2 + I3
(Instruction 2) I3 = I8
(Instruction 3) I4 = 2 * I6
(Instruction 4) I7 = I4 + I6



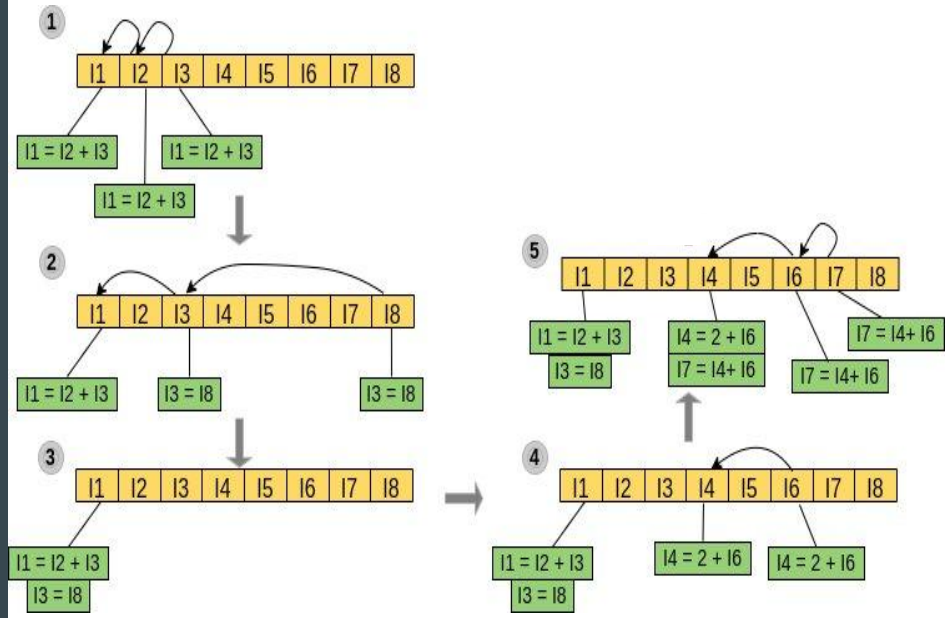
# Taint Assisted Concolic Grouping

(Instruction 1) I1 = I2 + I3
(Instruction 2) I3 = I8
(Instruction 3) I4 = 2 * I6
(Instruction 4) I7 = I4 + I6



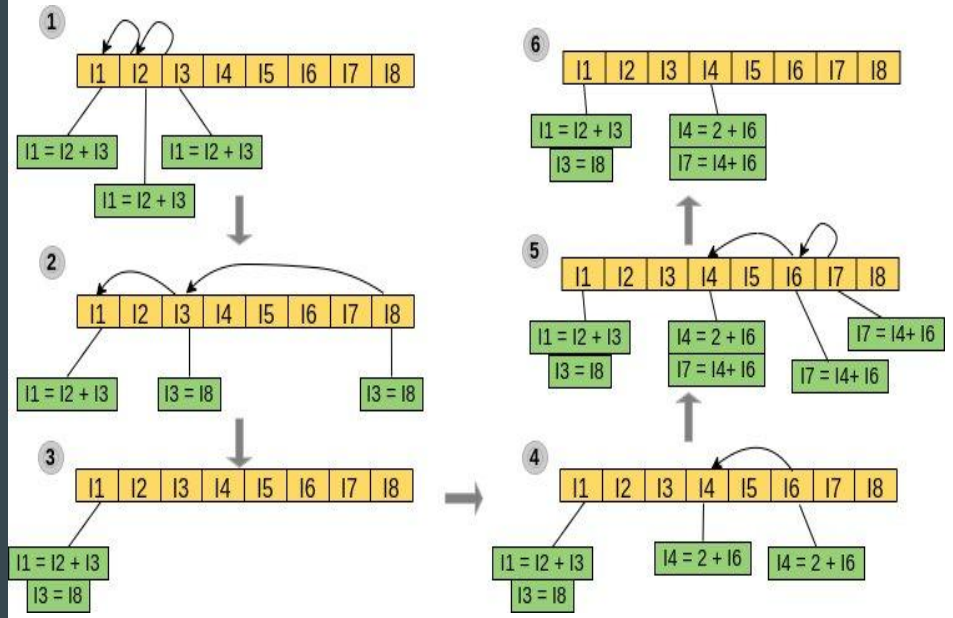
# Taint Assisted Concolic Grouping

**(Instruction 1) I1 = I2 + I3**
**(Instruction 2) I3 = I8**
**(Instruction 3) I4 = 2 * I6**
**(Instruction 4) I7 = I4 + I6**



# Taint Assisted Concolic Grouping

**(Instruction 1) I1 = I2 + I3**
**(Instruction 2) I3 = I8**
**(Instruction 3) I4 = 2 * I6**
**(Instruction 4) I7 = I4 + I6**



# Research Questions

**RQ1** What performance gains are offered by TACE compared to state-of-the-art tools?

**RQ2** Does TACE report correct and reproducible bugs?

# Results: Constraints Solving Time

| Depth | SymQEMU (sec) | TACE (sec) | Symbolic Dep | Concrete Dep | Improvement (x Times) |
|---|---|---|---|---|---|
| 1 | 4.28 | 4.30 | 3 | 2 | 0.99 |
| 2 | 7.44 | 5.69 | 4 | 3 | **1.30** |
| 3 | 9.76 | 6.26 | 4 | 4 | **1.55** |
| 4 | 15.02 | 7.84 | 4 | 6 | **1.91** |
| 5 | 26.11 | 9.35 | 4 | 10 | **2.79** |
| 6 | 122.65 | 7.81 | 4 | 18 | **15.7** |
| 7 | 145.50 | 4.38 | 4 | 34 | **0** |
| 8 | 280.48 | 5.59 | 4 | 66 | **33.2** |
| | | | | | **1** |
| | | | | | **50.1** |
| | | | | | **7** |

# Results: Constraints Solving Time

| Depth | SymQEMU (sec) | TACE (sec) | Symbolic Dep | Concrete Dep | Improvement (x Times) |
|---|---|---|---|---|---|
| 1 | 4.28 | 4.30 | 3 | 2 | 0.99 |
| 2 | 7.44 | 5.69 | 4 | 3 | **1.30** |
| 3 | 9.76 | 6.26 | 4 | 4 | **1.55** |
| 4 | 15.02 | 7.84 | 4 | 6 | **1.91** |
| 5 | 26.11 | 9.35 | 4 | 10 | **2.79** |
| 6 | 122.65 | 7.81 | 4 | 18 | **15.7** |
| 7 | 145.50 | 4.38 | 4 | 34 | **0** |
| 8 | 280.48 | 5.59 | 4 | 66 | **33.2** |

**RQ1** What performance gains are offered by TACE compared to state-of-the-art tools?

# Hybrid Fuzzing Statistics from SymCC vs TACE

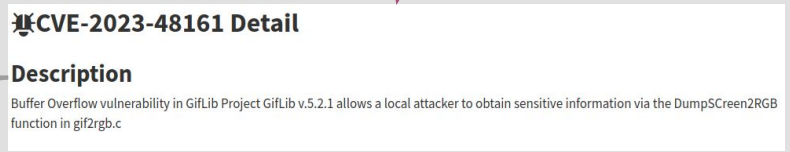
| Test Details | | SymCC | | | | | TACE | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Target Project** | **Time (hh:mm:ss)** | **#Unique Hangs** | **#Unique tmouts** | **#Unique Crashes** | **#New Edges** | **#Cycles** | **#Unique Hangs** | **#Unique tmouts** | **#Unique Crashes** | **#New Edges** | **#Cycles** |
| minizip-ng | 24:00:00 | 0 | 4 | 0 | 5 | 391 | 0 | 203 | 0 | 204 | 3.6k |
| TCPDump | 24:00:00 | 14 | 336 | 0 | 4148 | 223 | 36 | 1201 | 0 | 7 | 286 |
| GifLib | 24:00:00 | 2 | 71 | 0 | 78 | 108k | 13 | 62 | 7 | 72 | 181k |
| OpenJpeg | 24:00:00 | 0 | 1 | 0 | 3 | 89 | 10 | 894 | 0 | 71644 | 104k |
| bzip2 | 24:00:00 | 0 | 2 | 0 | 4 | 76.2 | 0 | 2 | 0 | 4 | 229k |

# Hybrid Fuzzing Statistics from SymCC vs TACE

**RQ2** Does TACE report correct and reproducible bugs?

| Test Details | | SymCC | | | | | TACE | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Target Project | Time (hh:mm:ss) | #Unique Hangs | #Unique tmouts | #Unique Crashes | #New Edges | #Cycles | #Unique Hangs | #Unique tmouts | #Unique Crashes | #New Edges | #Cycles |
| minizip-ng | 24:00:00 | 0 | 4 | 0 | 5 | 391 | 0 | 203 | 0 | 204 | 3.6k |
| TCPDump | 24:00:00 | 14 | 336 | 0 | 4148 | 223 | 36 | 1201 | 0 | 7 | 286 |
| GifLib | 24:00:00 | 2 | 71 | 0 | 78 | 108k | 13 | 62 | 7 | 72 | 181k |
| OpenJpeg | 24:00:00 | 0 | 1 | 0 | 3 | 89 | 10 | 894 | 0 | 71644 | 104k |
| bzip2 | 24:00:00 | 0 | 2 | 0 | 4 | 76.2 | 0 | 2 | 0 | 4 | 229k |





# Thank You

•••

Questions?