# ATOS introduction
# ST/Linaro Collaboration Context

**Presenter: Christian Bertin**

**Development team: Rémi Duraffort, Christophe Guillon, François de Ferrière, Hervé Knochel, Antoine Moynault**

**Consumer Product Division, Compilation team**

2016/3/8

life.augmented

Linaro

# Project Rationale

- Optimizing large open source applications by hand is a dead-end
  - ⇒ Sources and makefiles must be optimized as is

**Let the machine do the hard work**

**(HiPEAC Roadmap)**

Develop a tool, which automatically tunes the compiler for a specific set of use cases, seamlessly applying advanced compilation optimization

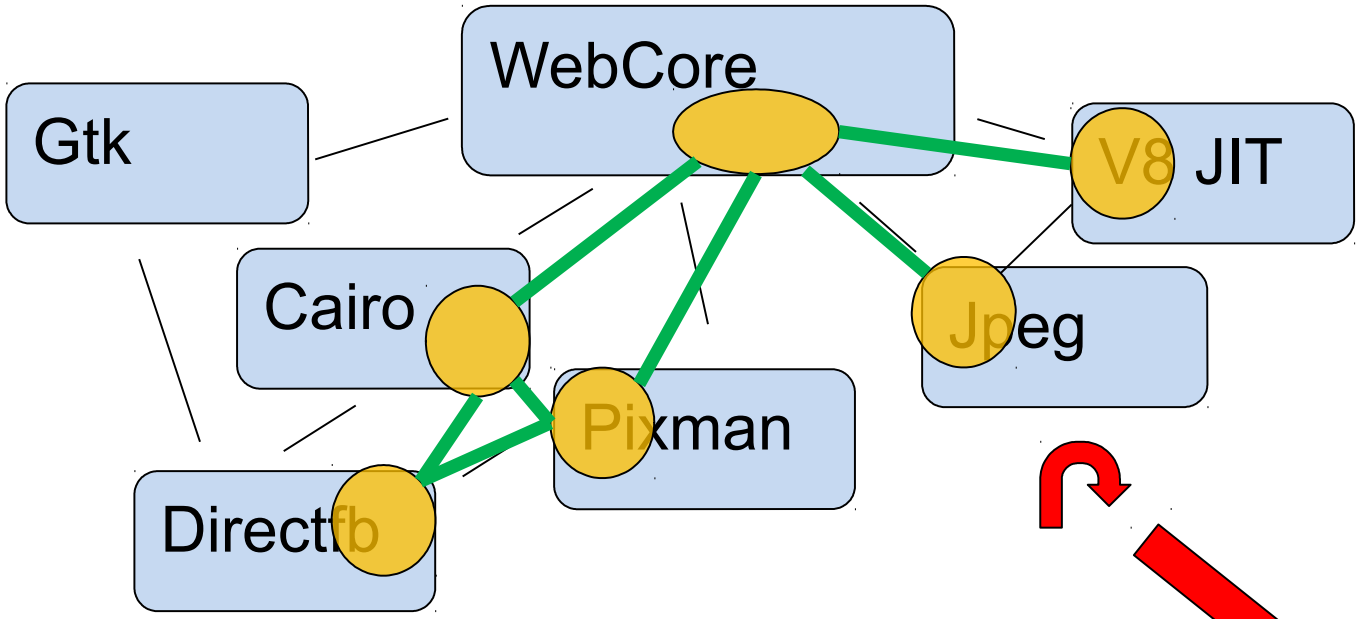
# Auto Tuning Optimization System

ATOS stands for:

"Auto Tuning Optimization System"

- Functional requirements:
  - Automatically find best C/C++ compiler configuration for a given objective.
  - Explore on top of any build system with a given set of executable use cases
  - Preserve original sources, makefiles and build scripts (they are not modified)
- Application:
  - Find best performance / code size tradeoffs search for:
    - a given set of executables/libraries[/kernel modules] and
    - a given set of benchmarking use cases

# Optimization Challenges




Identify Hot Code

**Advanced Profiling Tools**

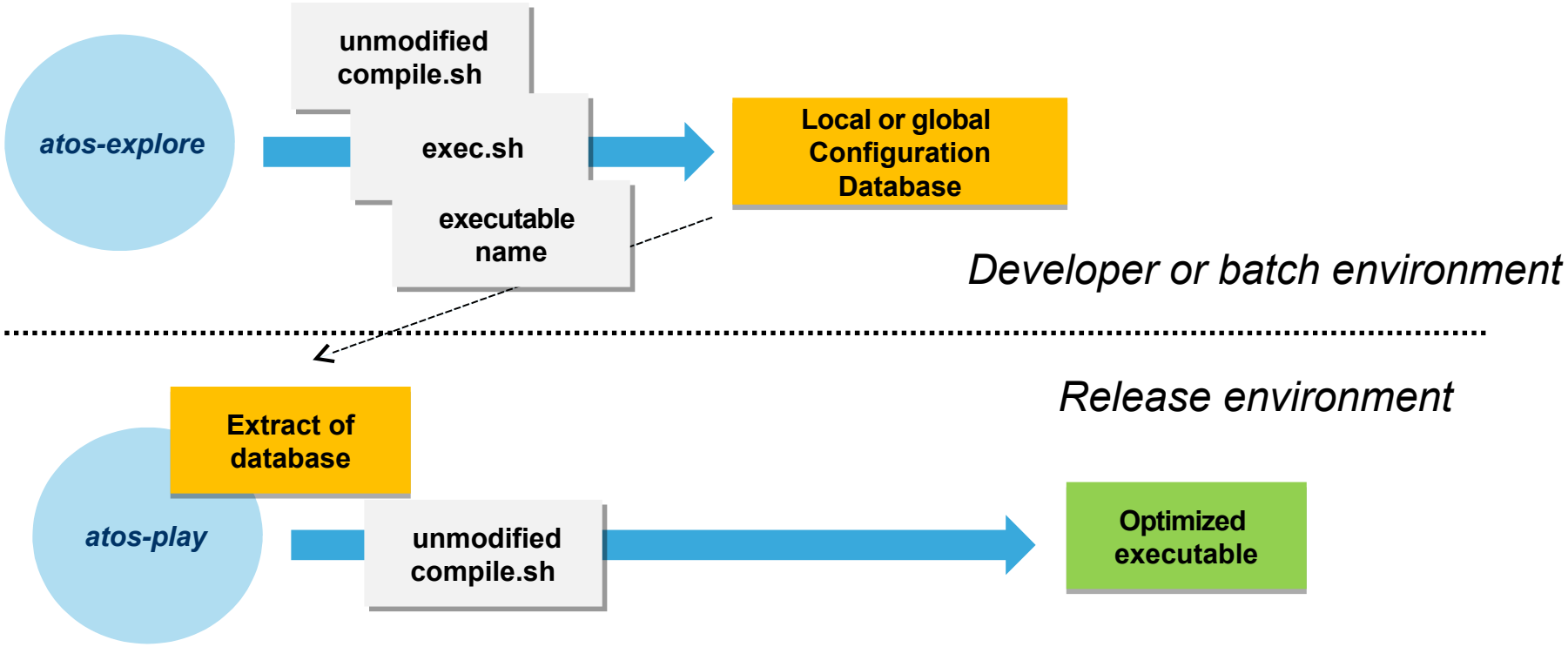
Enlarge Optimization scope

**Inter Procedural and Cross Library**

Find best compiler tuning

**Iterative Optimization**

# ATOS High Level Usage



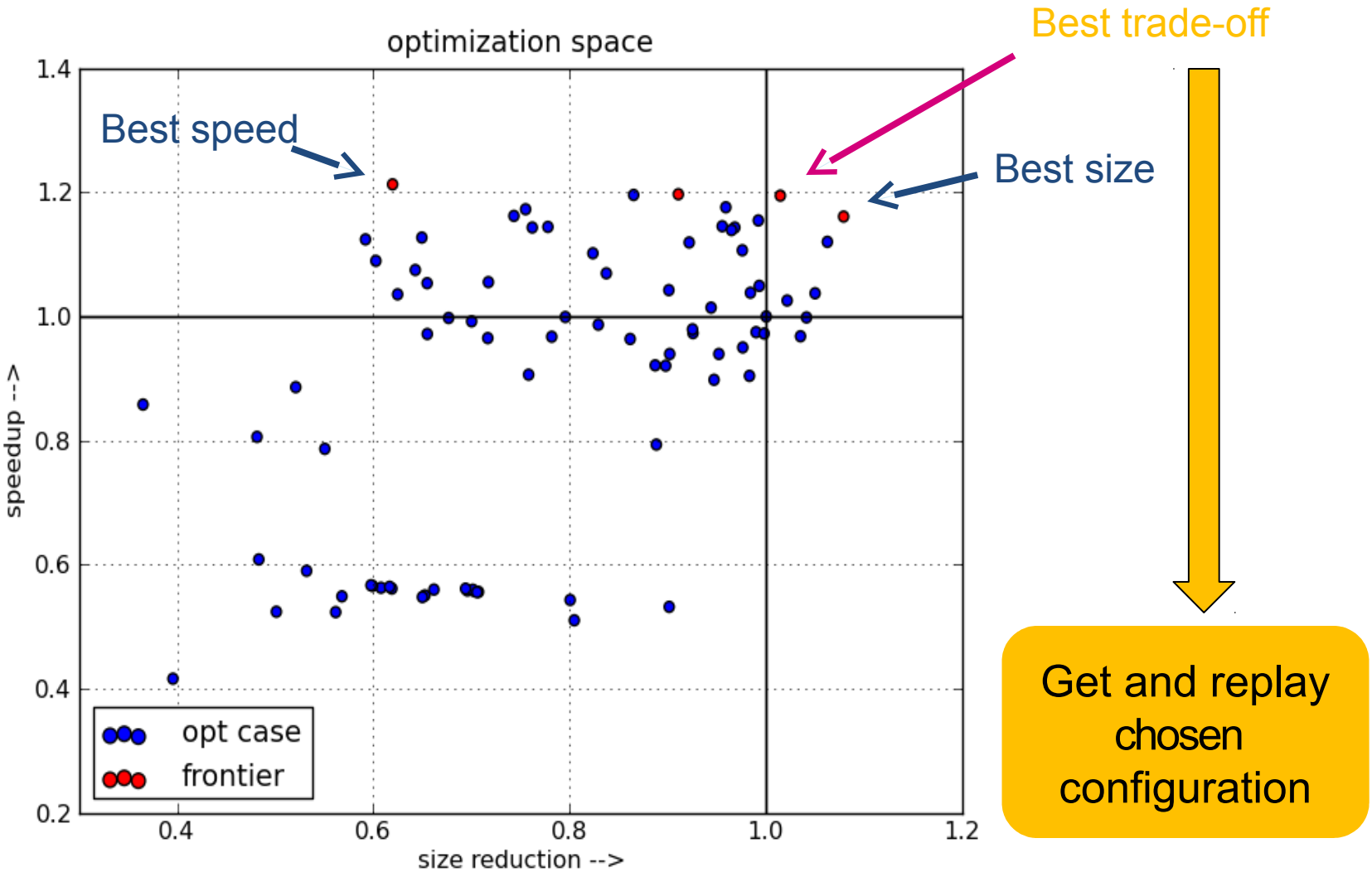

Unmodified compile.sh: actually any build system command:

- make all
- rpmbuild
- build.sh

exec.sh: actually a script running the application

- make run
- direct or remote execution
- run.sh
- emulated or on board

# Example of Preferred Configuration



# ATOS Features

- Whole executable optimization sequences
- File by file exploration of optimization sequences
- Function by function exploration with provided GCC plugin
- Support for 'perf' and 'oprofile‘ tools
- Seamless profiling feedback and link time optimizations support
- Support in GCC 4.5 to 6.0 / LLVM 3.4 and 3.6 /  ARM RVCT
- Simple command line interface
- Native or cross build and run

# ATOS Features (continued)

Latest features/improvements (ATOS v4.0)

- Kernel build support  (*.ko)
- Various exploration schemes: random, staged or genetic
- Parallel build and execution (native or remote) capabilities
- Parameters fine-tuning and flags pruning exploration

# Example of Exploration Usage

***Initialization of session***

```
$ atos-init -b ./build.sh -r ./run.sh -p ./run-oprofile.sh
```

***First basic exploration O(12)***

```
$ atos-explore
```

***Whole program flags exploration O(M.G.3)[def: O(3000)]***

```
$ atos-explore-genetic [-M100] [--generations 10]
```

***File by file exploration O(H.N.36) [def: O(H.3600)]***

```
$ atos-explore-acf -file-by-file [-N100]
```

***Funct by funct exploration O(H.N.36) [def: O(H.3600)]***

```
$ atos-explore-acf [-N100]
```

***Show exploration graph***

```
$ atos-graph
```

Where: O(…): complexity in number of build+run, H: number of determined Hot file (resp. function)

# Example of Replay Usage

***Output best perf tradeoff***

```
$ atos-play -T
speedup | sizered |              target |                          variant_id
+30.56% | -13.44% | sha-shatest-shacmp | bbc60eb968fb803987f4df19304e6a1e
```

***Output best size tradeoff***

```
$ atos-play -T -f size
speedup | sizered |              target |                          variant_id
+13.57% |  +3.58% | sha-shatest-shacmp | 7b9351c456fa303f8e3ef446e2028aff
```

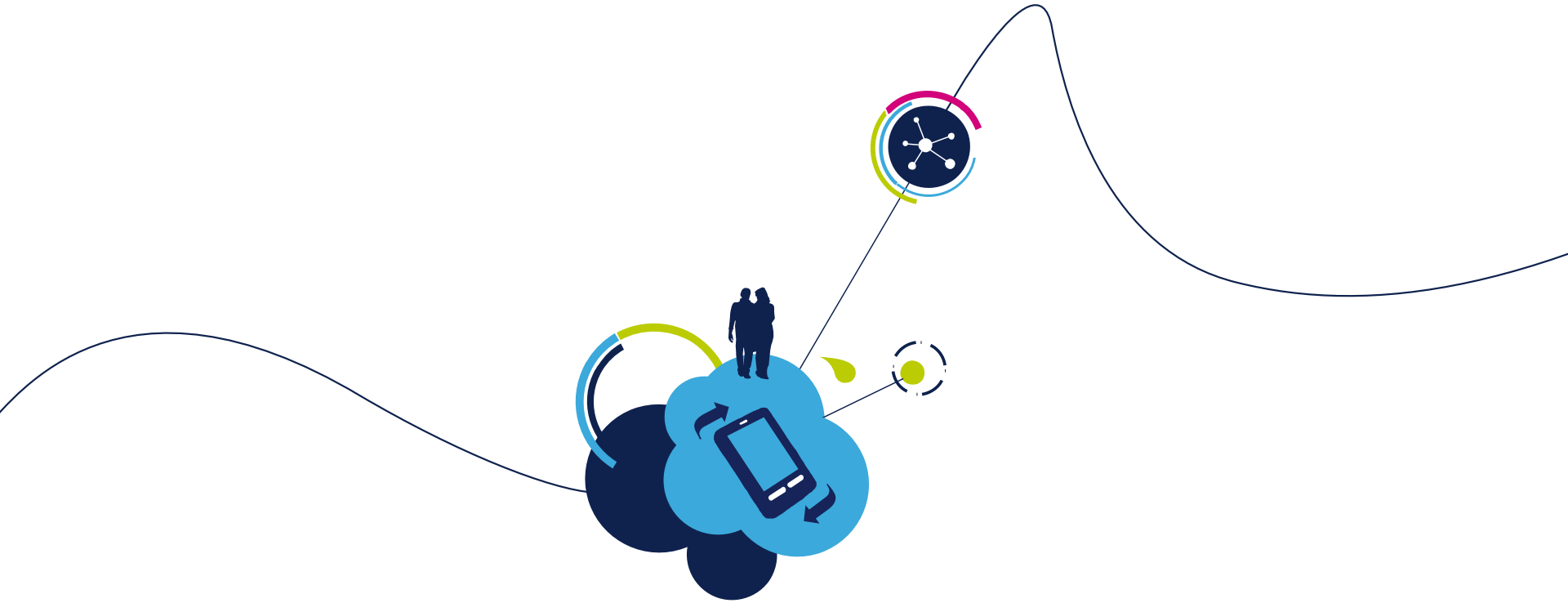
***Compile for best perf tradeoff***

```
$ atos-play
Building Variant [bbc60eb968fb803987f4df19304e6a1e]…
```

***Compile a selected variant***

```
$ atos-play -l 7b9351c456fa303f8e3ef446e2028aff
Building Variant [7b9351c456fa303f8e3ef446e2028aff]…
```

# Video of exploration using parallel build/run features



# ATOS Outcomes

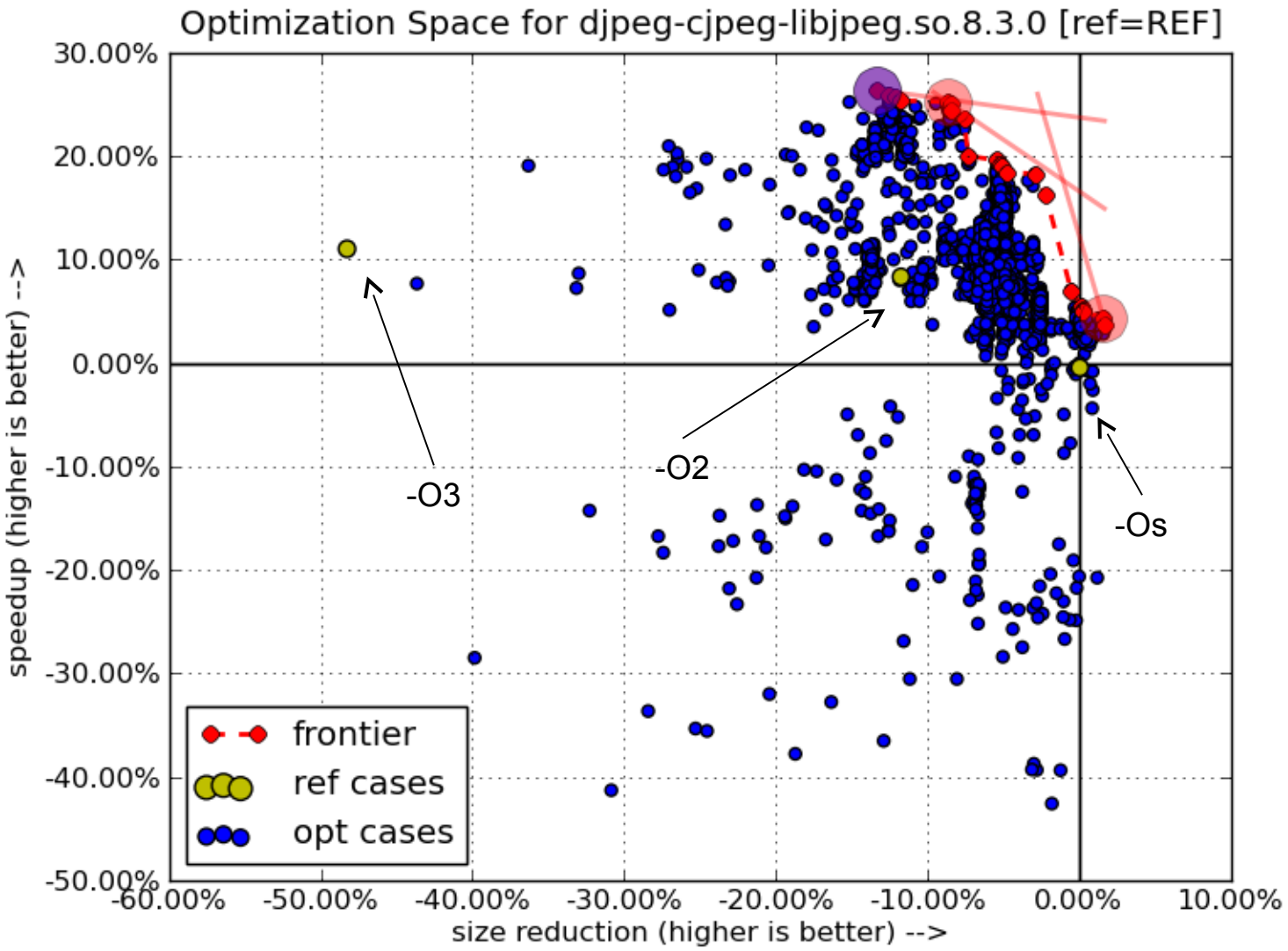
life.augmented

Linaro

# ATOS Optimization Results

- JPEG ST40 / HDK7108 results
  - 26.39% speedup / 13.37% size increase
- ZLIB ST40 / HDK7108 results
  - 12.54% speedup / 1.41% size increase
- Stagecraft ST40 / HDK7108 results
  - 5-28% speedup (30 benchs) / 14% size reduction
- HEVC ARMv7 / Orly results
  - 9.22% speedup / 21.21% size reduction
- SPEC2000 ARMv7 / U9540 results
  - 18.7% speedup SPECINT / 10.2% speedup SPECFP / size increase n/a

# JPEG ST40 / HDK7108

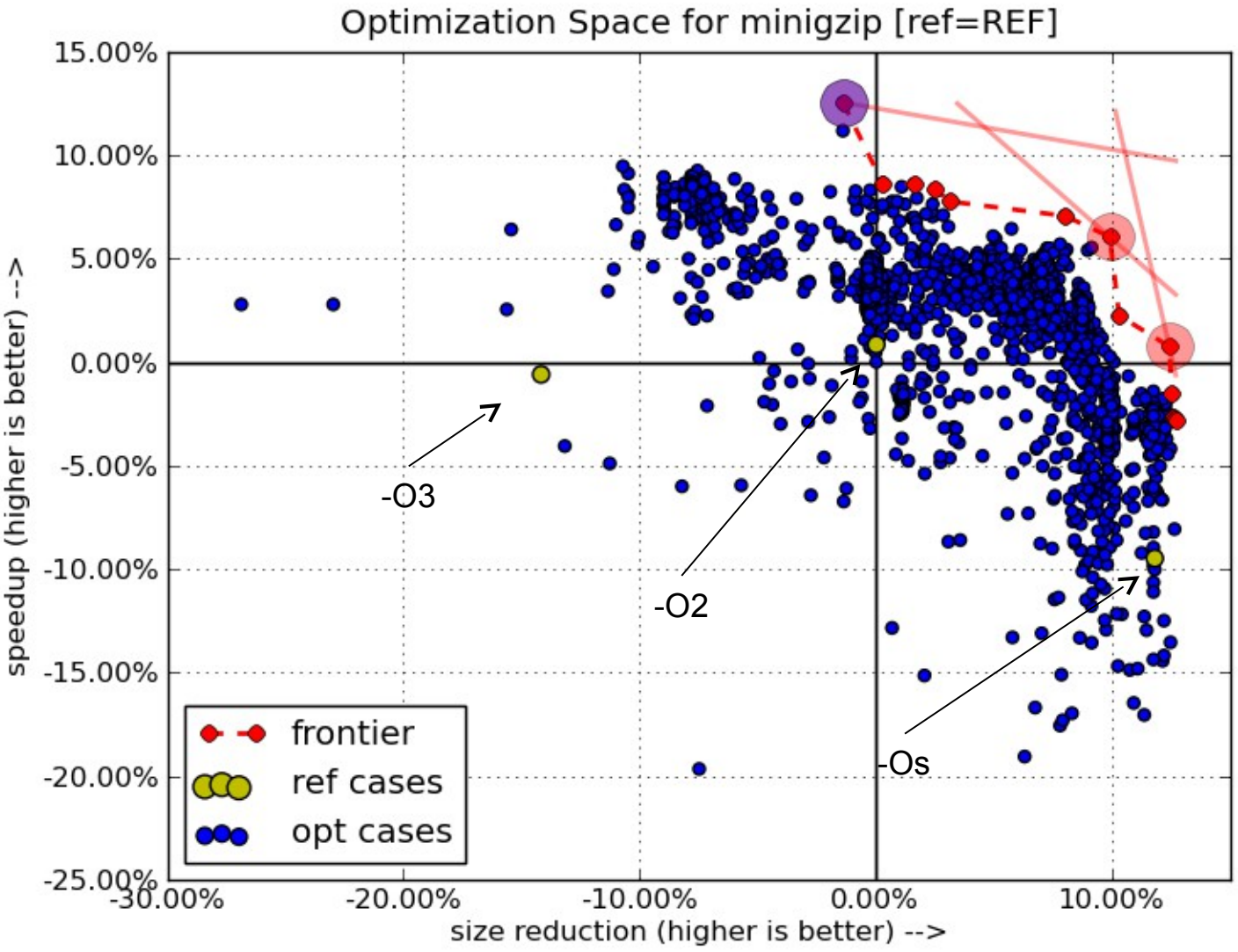



**Best perf v2.0:**
**26.39% speedup**
**13.37% size increase**

life.augmented

Linaro

# ZLIB ST40 / HDK7108

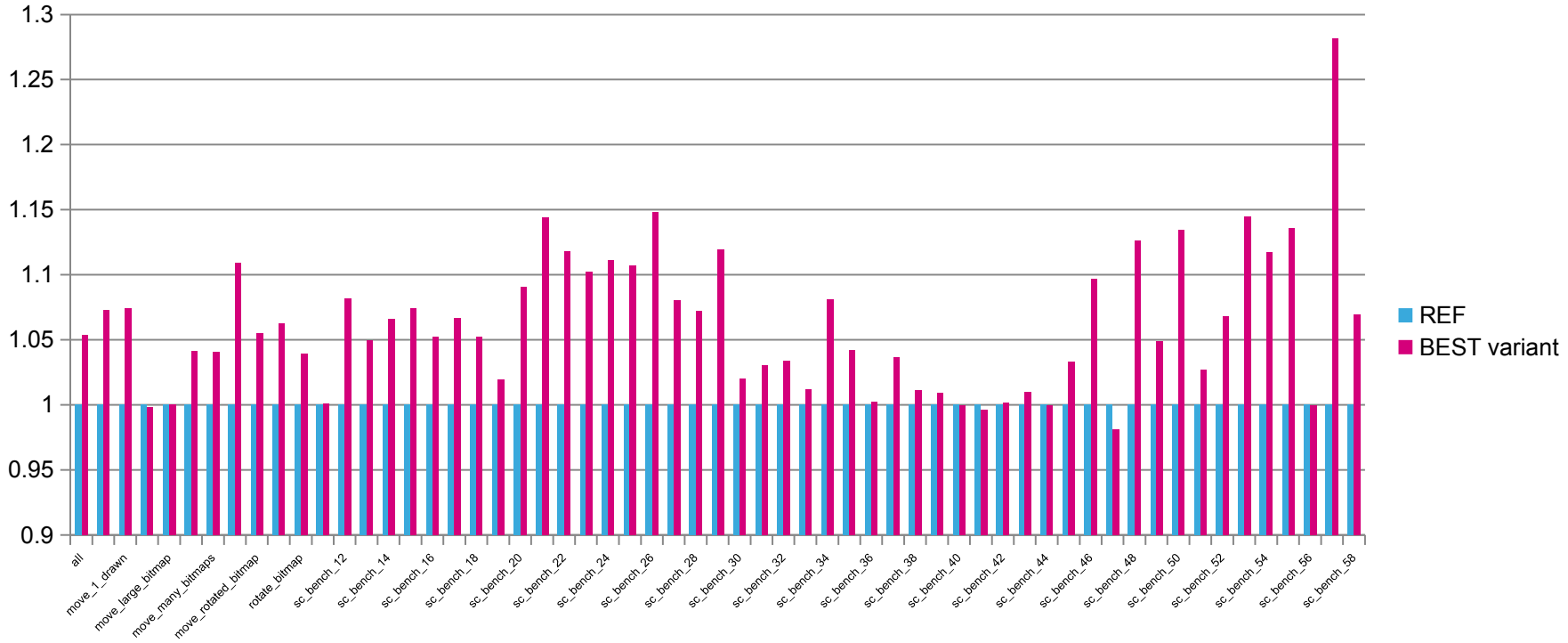



**Best perf v2.0:**
**12.54% speedup**
**1.41% size increase**

life.augmented

Linaro

# StageCraft ST40 / HDK7108




- *Results:*
  - *Performance: 5-15 % speedup range in FPS, one at 28%*
  - *Size: 14% footprint reduction (out of 35Mb initial footprint)*

- Reference: build from CPD –O2/-O3 mix, compiler gcc 4.5.2 (no lto, no plugins)
- Benchmark: Sagem provided benchmarks
- Profile: 58 use cases, 30 of which not fully HW accelerated
- Optimization scope Stagecraft binary and 21 accompanying shared objects
- Outcome: Customer interested but actually the packages are provided by ST/CPD

life.augmented

Linaro

# ATOS* HEVC** results
# Summary and robustness to source updates

Speed-up and code size compared to reference configuration : **-O3** build
Performance measurement unit: CPU time (process user time)
Benchmark: decoding of the 300 first frames of a typical stream***

| Experiment | Performance first | |
|---|---|---|
| | speed-up | size reduction |
| 1. Initial exploration (HEVC version 1) (4618 runs) | 6.18% | 22.41% |
| 2. Replay best after update (HEVC version 2) (0 more run) | 7.55% | 20.16% |
| 3. More advanced fine-tuning on top of 2. (1267 additional runs) | **9.22%** | **21.21%** |

*Source update without additional exploration*

*Some more explorations*

* ATOS version: atos v2.0
** HEVC version: HEVC full SW ARM Cortex, Optimized, 2 distinct versions
*** stream: RA_LC_TotalRecallHotelTransylvania_1280x720_25_QP26.bin

# SPECINT2000* ARMv7 / U9540

| | Time -OS | Speedup -Os/-O3 | Speedup -Os/ATOS | Speedup -O3/ATOS |
|---|---|---|---|---|
| 164.gzip | 40.9 | 7.4 | 24.8 | 16.1 |
| 175.vpr | 37.6 | 11.7 | 16.8 | 4.6 |
| 176.gcc | 4.0 | 6.7 | 25.4 | 17.6 |
| 181.mcf | 45.6 | 2.2 | 5.0 | 2.8 |
| 186.crafty | 31.7 | 8.5 | 21.1 | 11.5 |
| 197.parser | 12.1 | 5.3 | 29.3 | 22.8 |
| 252.eon | 12.4 | 54.7 | 80.1 | 16.4 |
| 253.perlbmk | 68.6 | 4.2 | 22.2 | 17.3 |
| 254.gap | 7.0 | 0.7 | 8.6 | 7.8 |
| 255.vortex | 15.4 | 2.5 | 85.6 | 81.2 |
| 256.bzip2 | 48.2 | 3.1 | 17.7 | 14.1 |
| 300.twolf | 17.8 | 0.3 | 28.3 | 27.8 |
| **SPECint** | | **+8.2%** | **+28.4%** | **+18.7%** |

* SPECINT2000 results and improvements on train dataset
(Speedup -O3/ATOS: +13.97% when re-executed on REF dataset)

life.augmented

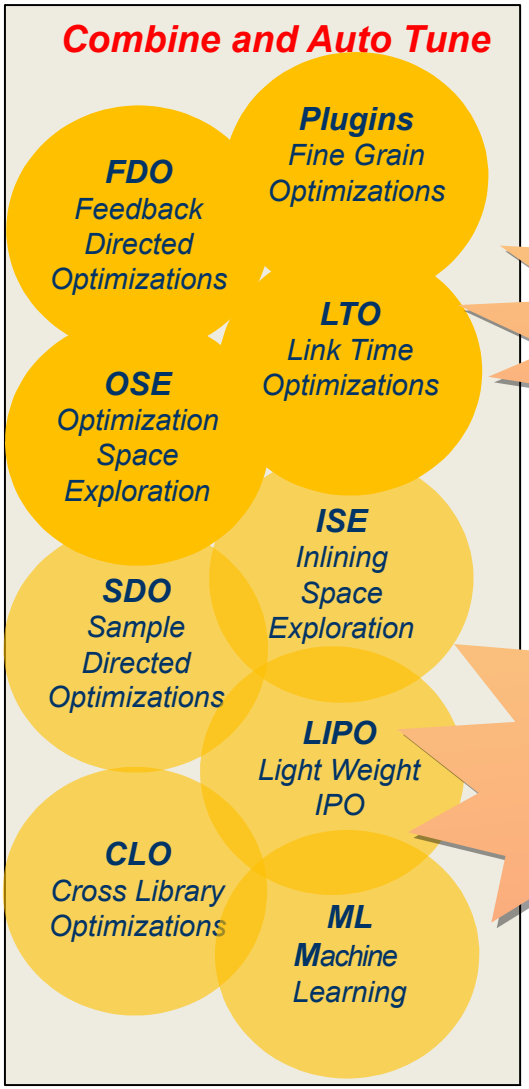
Linaro

# SPECFP2000* ARMv7 / U9540

| | Time -OS | Speedup -Os/-O3 | Speedup -Os/ATOS | Speedup -O3/ATOS |
|---|---|---|---|---|
| 177.mesa | 47.1 | 27.9 | 54.4 | 20.7 |
| 179.art | 22.3 | 0.3 | 9.2 | 8.9 |
| 183.equake | 40.4 | 3.3 | 6.4 | 2.9 |
| 188.ammp | 165.0 | 66.9 | 82.1 | 9.1 |
| **SPECfp** | | **+22.0%** | **+34.4%** | **+10.2%** |

* SPECINT2000 results and improvements on TRAI dataset
(Speedup -O3/ATOS: +6.53% when re-executed on REF dataset)

# Compilation Technologies for ATOS




**Well proven technologies**

Fine grain optimizations from whole to per file to per function

Use application training run to drive optimizations

Perform optimizations across object files

Explore the space of possible compiler optimizations mix

**Compiler techno. under improvement or development**

*Explore the space of possible inlining decisions*

*Continuously monitor user apps to drive optimizations*

*Group related object files for improving optimization search*

*Perform optimizations across libraries*

*Reuse past results for accelerating search*

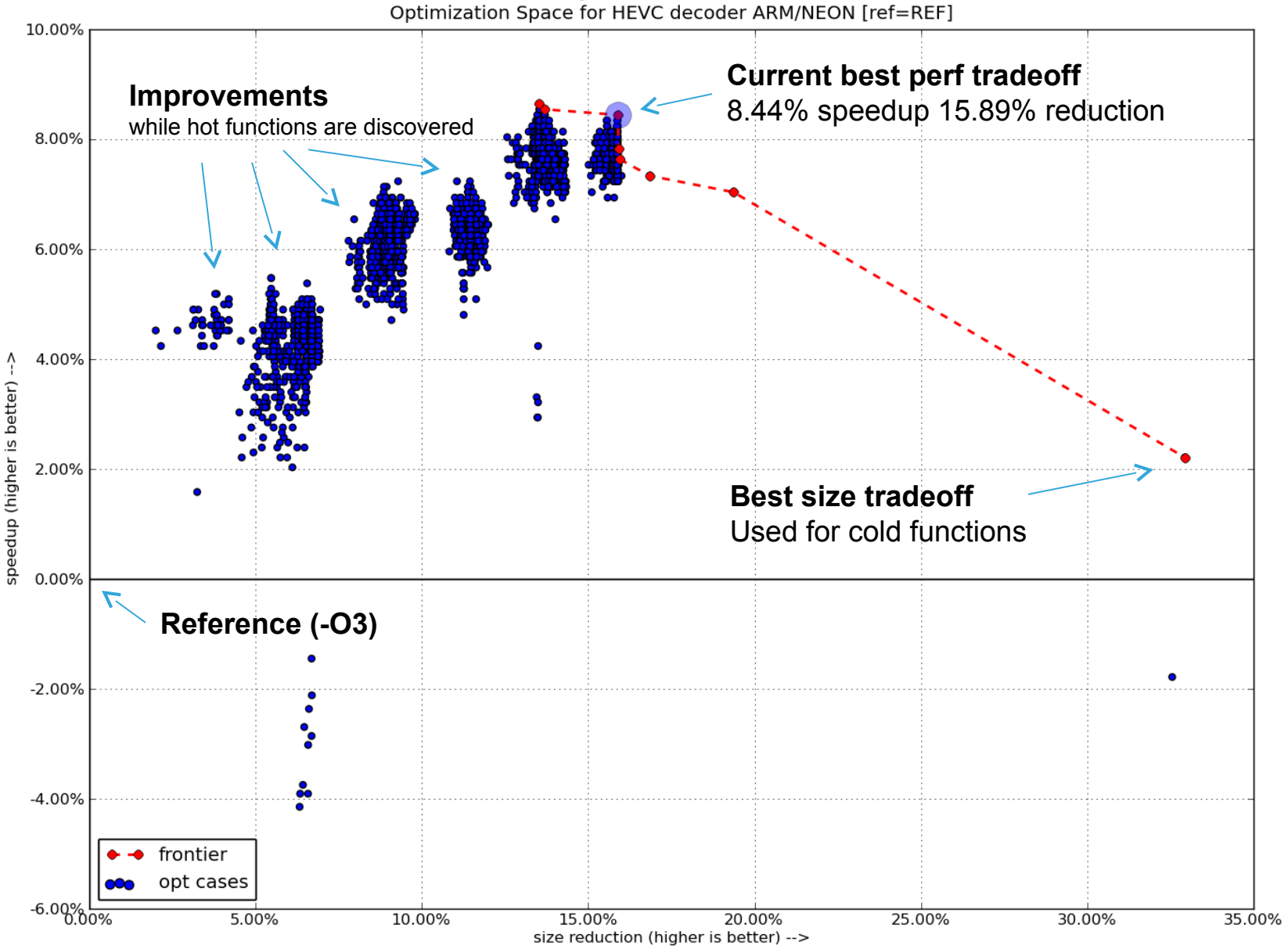
Fully automated build/run optimizer (ATOS)

life.augmented

Linaro

# ATOS function-by-function exploration





# ATOS Status

- ATOS has been in production since 2 years
- ATOS FOSS-OUT is done (approval e/o February)
- ATOS to be made open source by March 2016 on GitHub
  - License GPL v2 and later
- STMicroelectronics initial gatekeeper

# ATOS Delivery Script

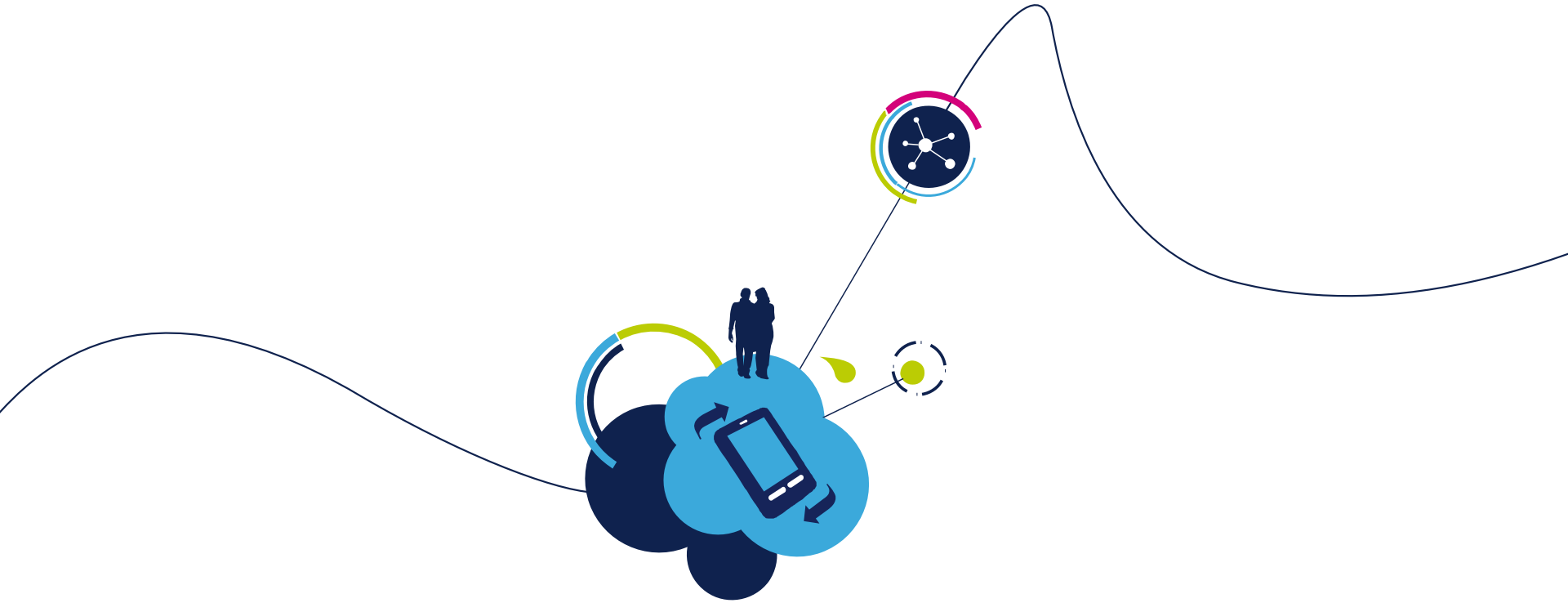
ATOS surdriver script to install in compiler directory:

```
#! /bin/bash

CC_PATH="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"

ATOS_OPTS="-Os --param inline-unit-growth=90 --param large-
function-insns=4000 --param large-stack-frame-growth=3000 --param
max-inline-insns-auto=20 --param max-inline-insns-recursive-
auto=1000 --param max-inline-insns-recursive=300 --param max-
inline-insns-single=1200 --param max-inline-recursive-depth=8
-fearly-inlining -findirect-inlining -fno-inline-functions -fno-
inline-functions-called-once --param max-early-inliner-
iterations=1 --param iv-consider-all-candidates-bound=10 -fno-
aggressive-loop-optimizations -falign-loops -fno-move-loop-
invariants -fno-rerun-cse-after-loop -ftree-dce -fno-tree-loop-
distribution -fno-tree-loop-im -ftree-loop-optimize -fno-tree-
scev-cprop -ftree-slp-vectorize -fno-unroll-loops"

${CC_PATH}/sh4-linux-uclibc-gcc ${1+"$@"} ${ATOS_OPTS}
```



# Thanks for your attention

life.augmented

Linaro