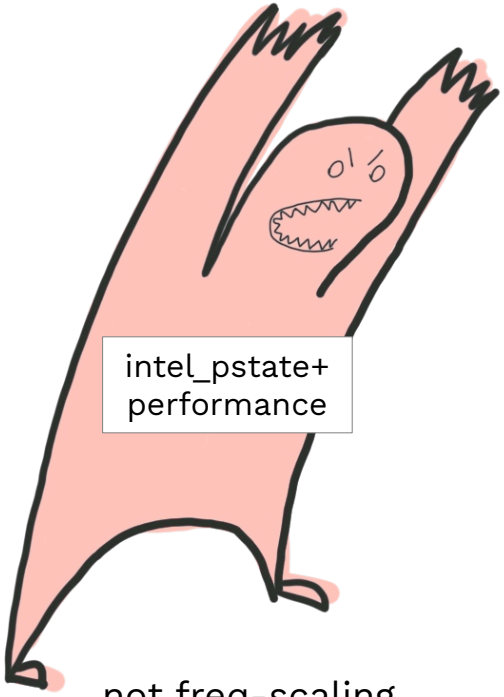
A bit more of this please?

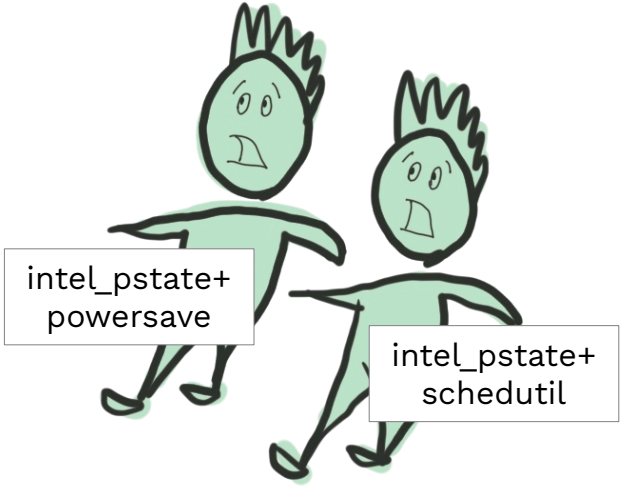
When it comes down to a **judgment call**, favoring performance over energy saving may **keep frequency scaling alive** in the server world

photo by Roman Avdagić, Foto-Škrinja j.d.o.o.
https://www.flickr.com/photos/romanski/44296560680
license: CC-BY 2.0 https://creativecommons.org/licenses/by/2.0
modified to include quote

perf-gov is not freq-scaling



not freq-scaling



freq-scaling

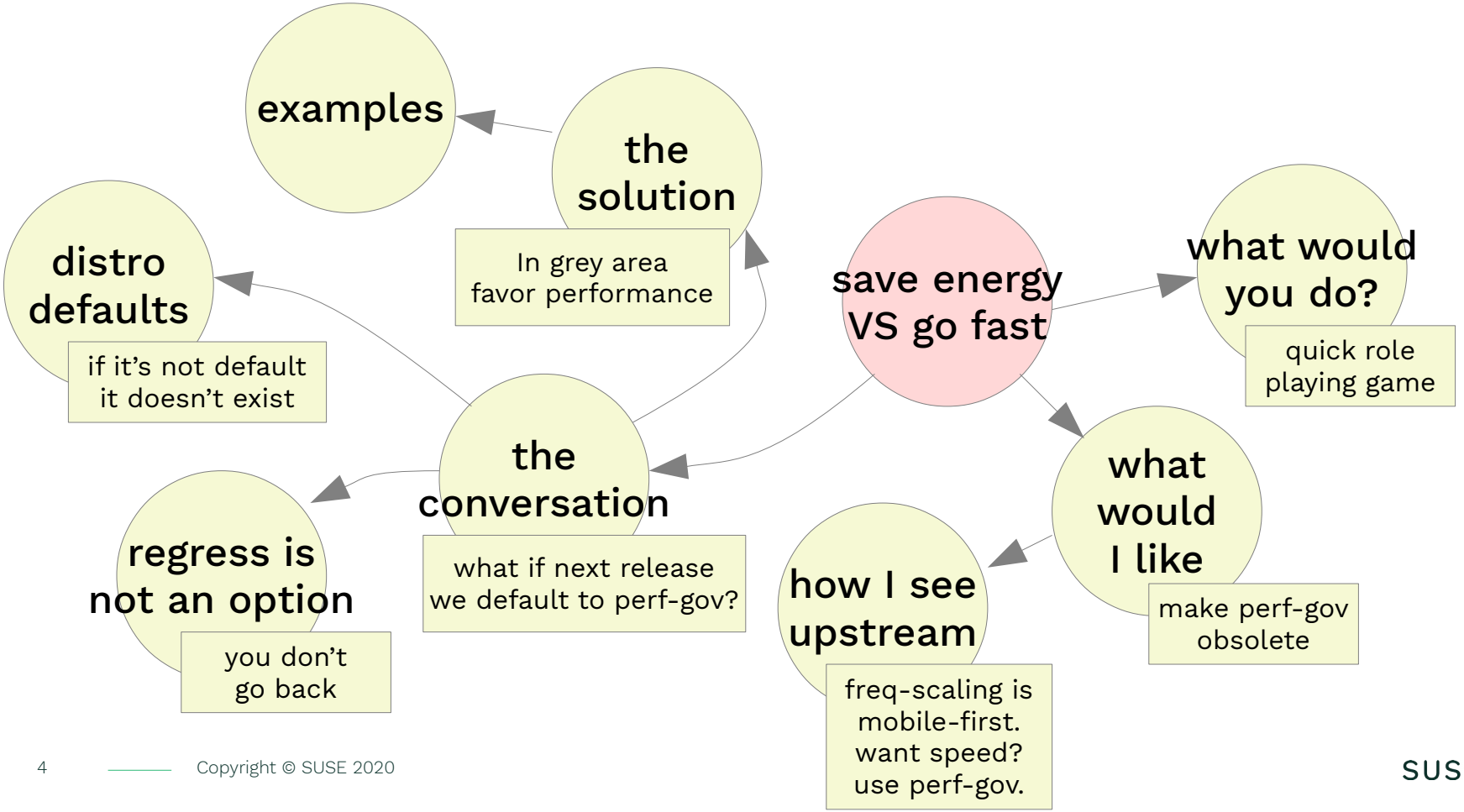
MAY 18, 2020, OSPM

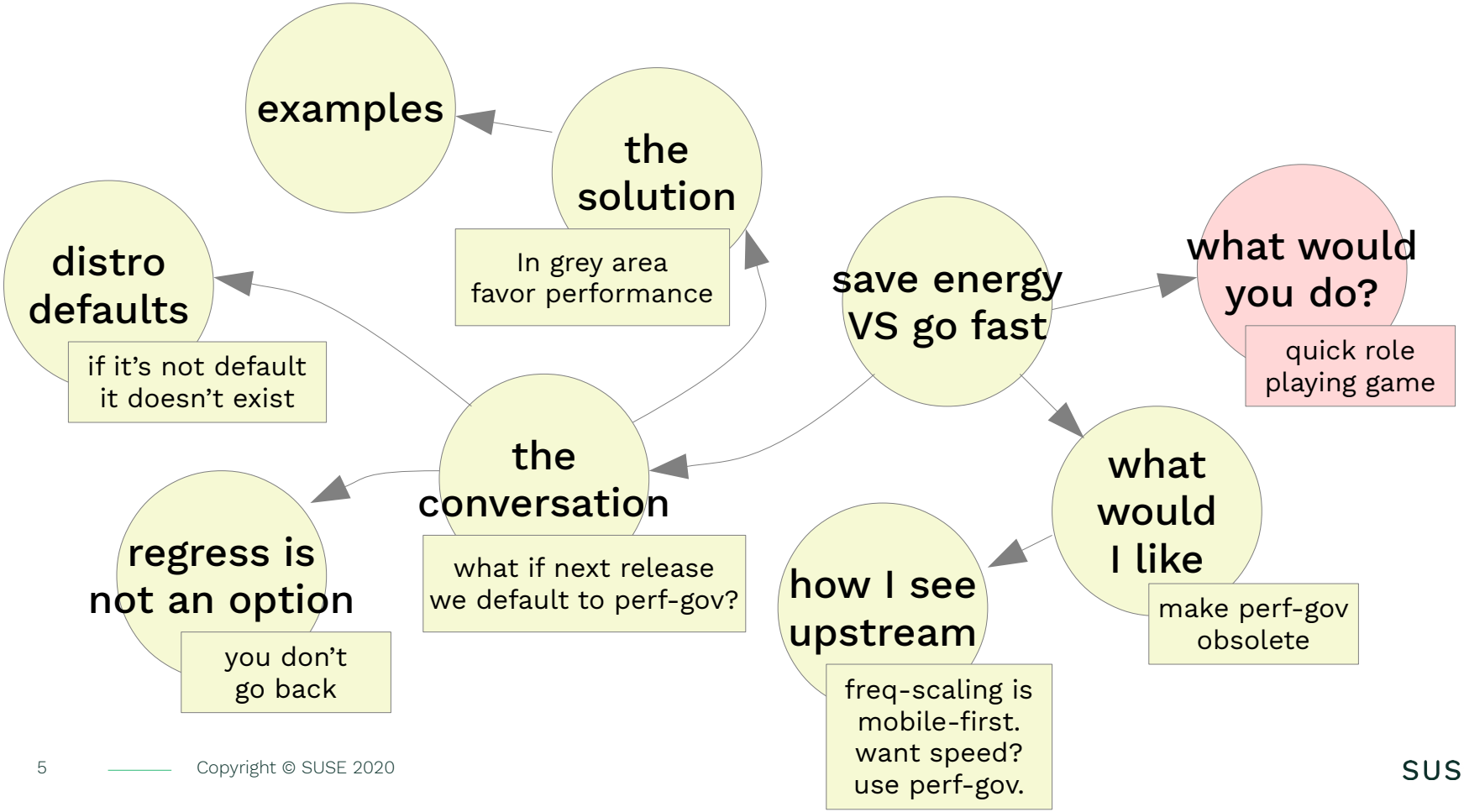
Frequency scaling in the datacenter

The case for a more aggressive intel_pstate/powersave

Giovanni Gherdovich
ggherdovich@suse.cz

SUSE





What would you do? #1

You're head of IT at **Bazinga Bank LLC**

- ~100 branches
- IT dept: ~50 people
- on premises (regulations)

Question: which freq-scaling policy?

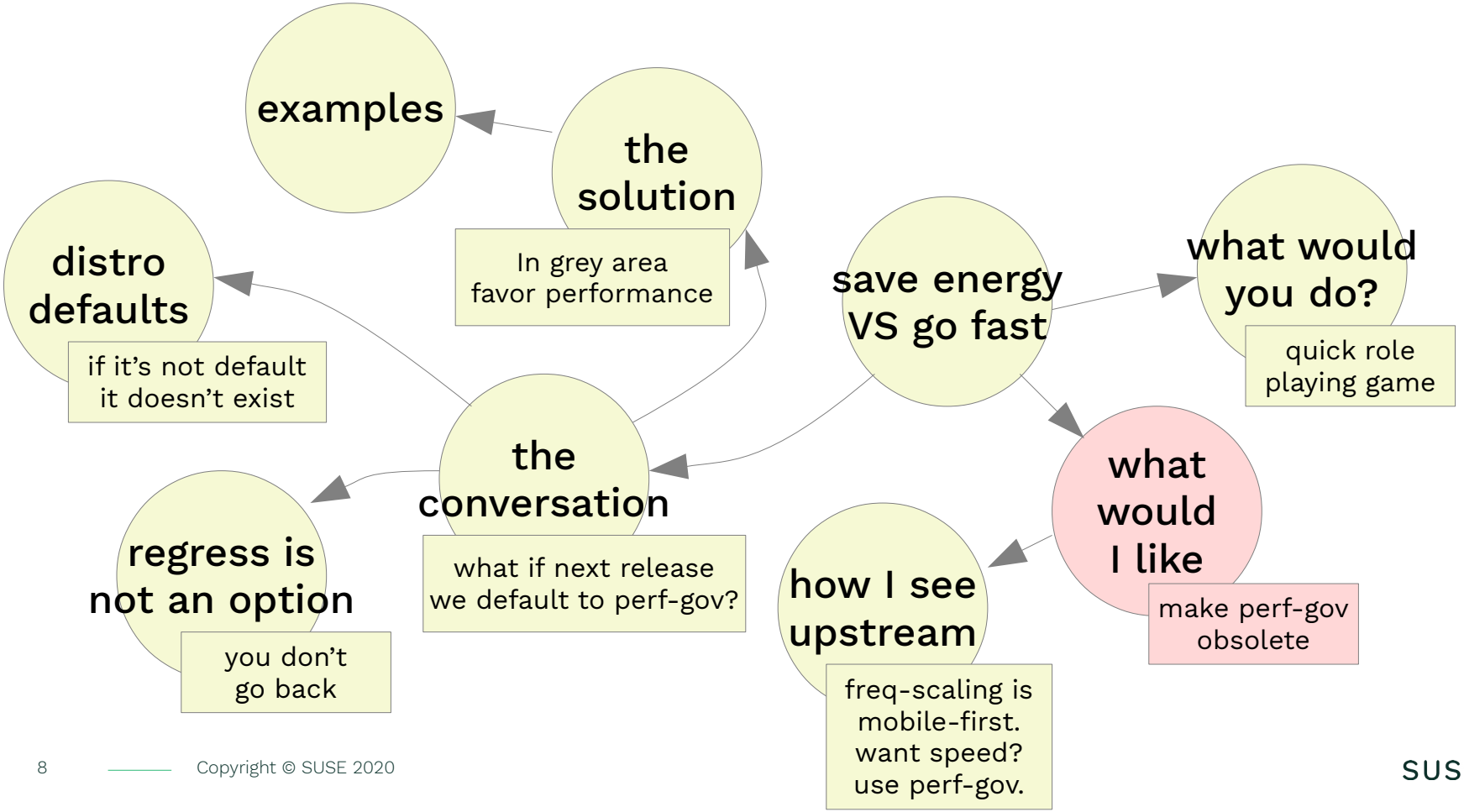
A) intel_pstate/powersave (the smart one)
B) intel_pstate/performance (the always-max one)
C) intel_pstate/schedutil (the next-gen one)
D) whatever the default is

What would you do? #2

You’re an OS vendor (SUSE, Red Hat, Canonical, ...)

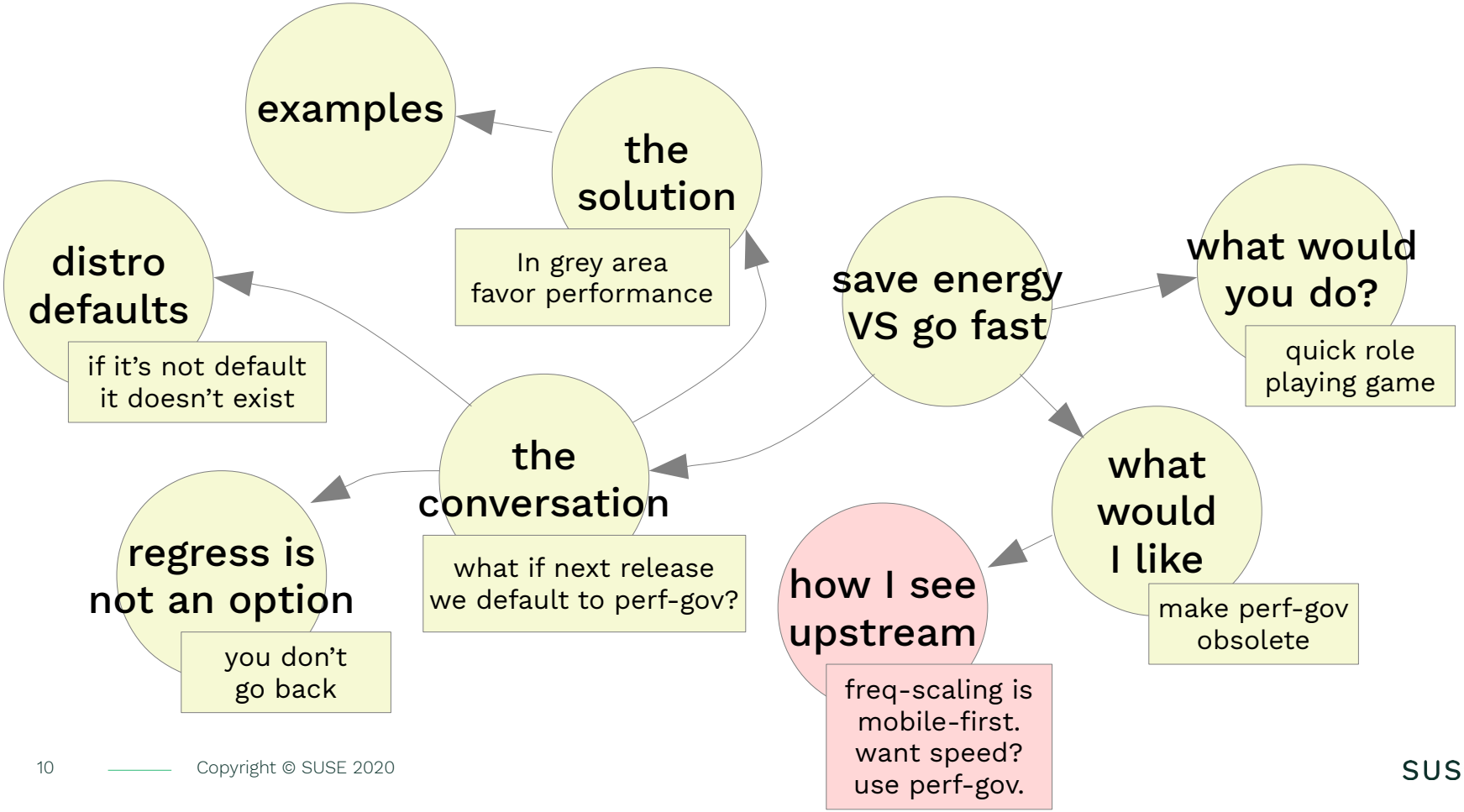
Question: which freq-scaling policy by default?

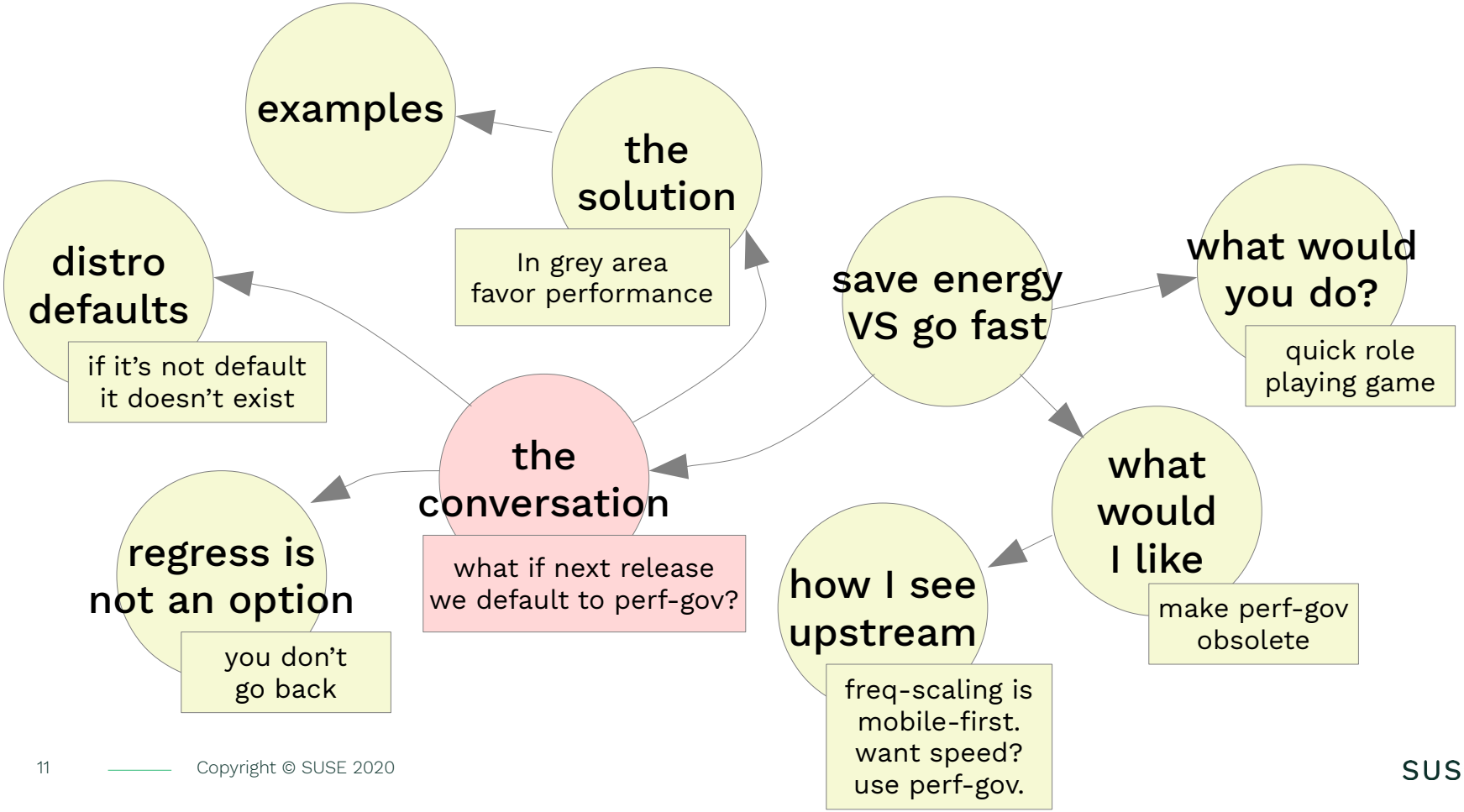
A) intel_pstate/powersave (the smart one)
B) intel_pstate/performance (the always-max one)
C) intel_pstate/schedutil (the next-gen one)



What would I like

- see freq-scaling the obvious choice on server
- see intel_pstate/performance obsolete, only HPC/HFT
- see schedutil win



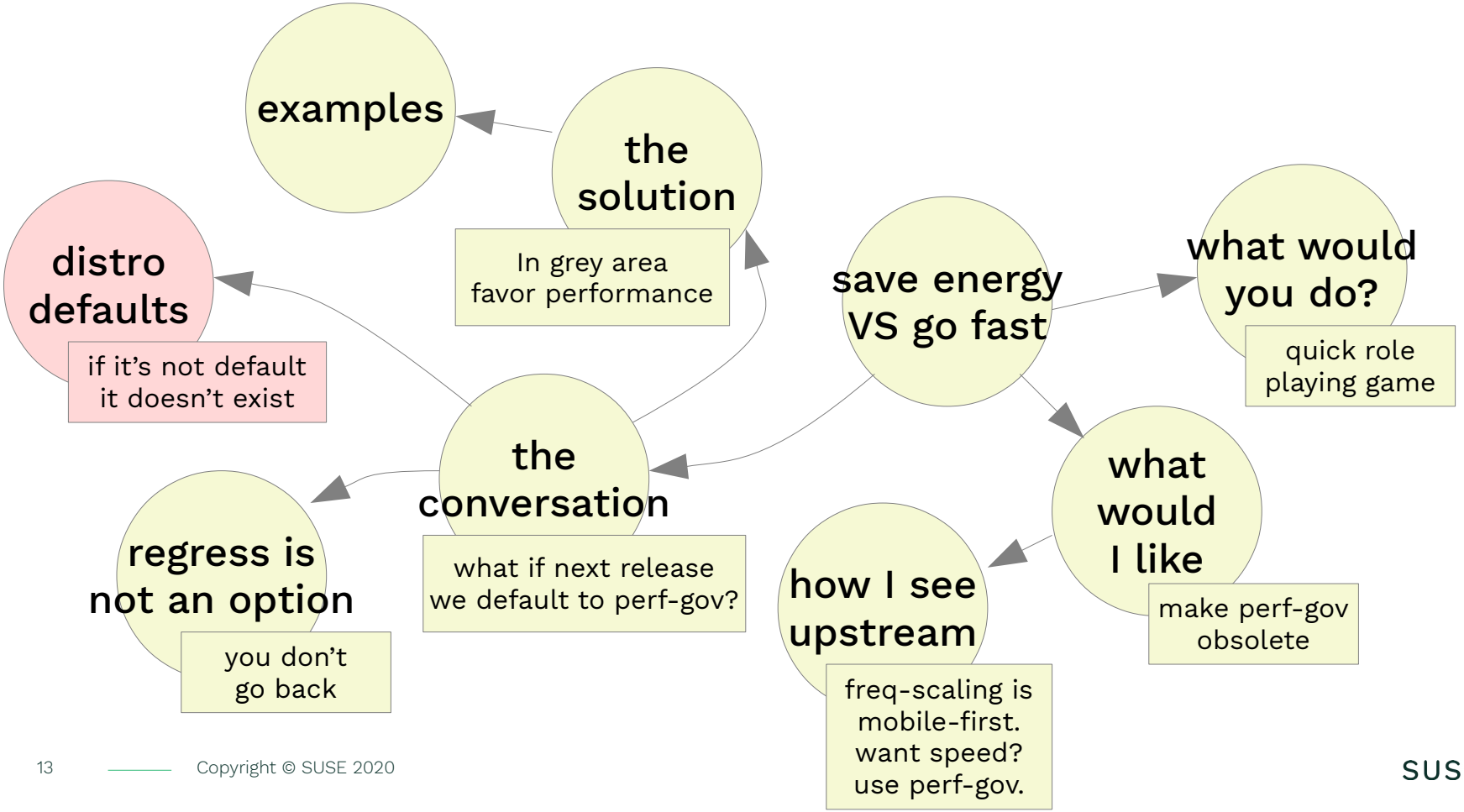


The conversation

what if next release we default to intel_pstate/performance?

PROs
- getting “powersave” in tip-top shape is hard work (out-of-tree)
- remove a whole dimension of complexity

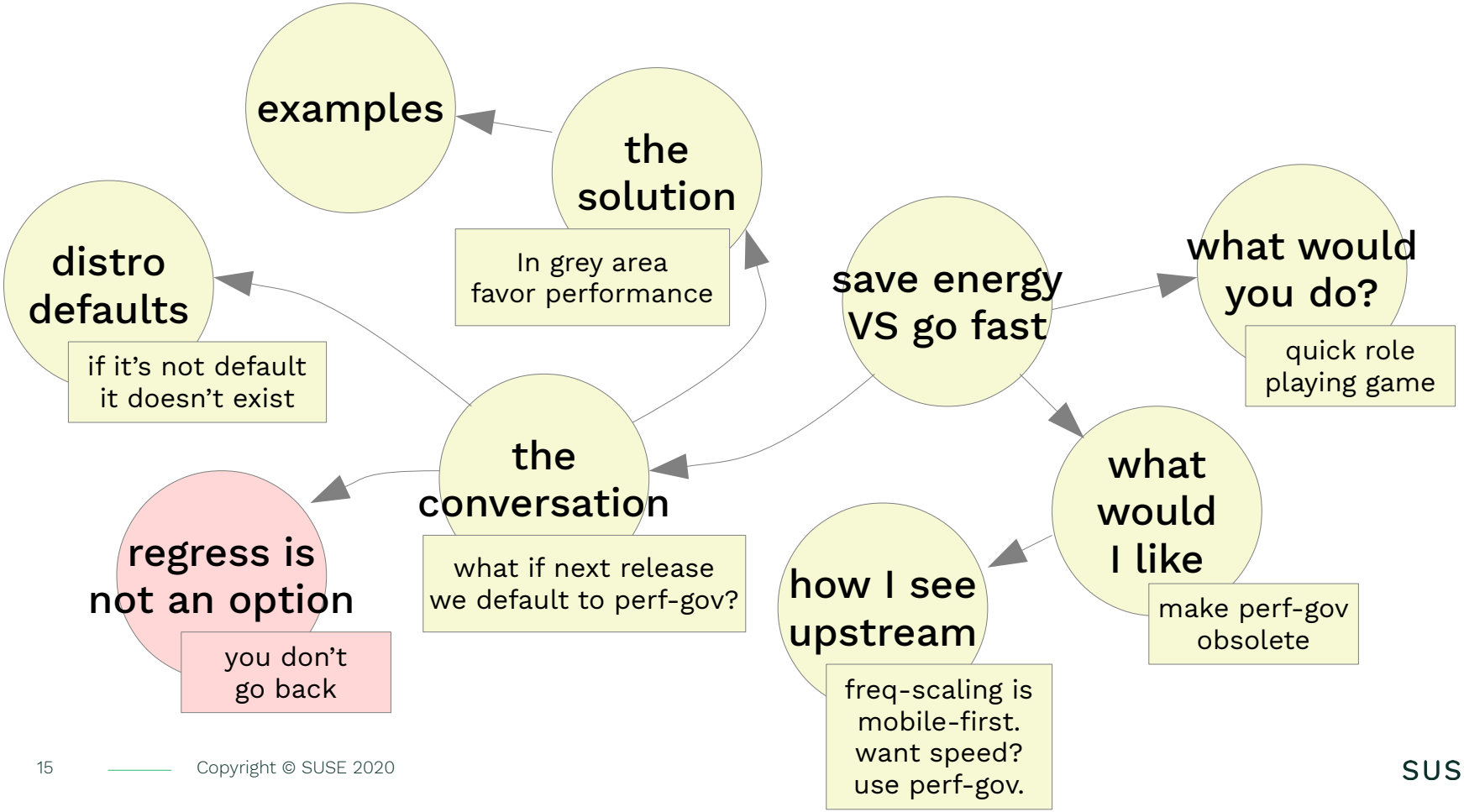
CONs
- lose handle on tech you’ll need one day
- mask bugs that are worth fixing instead
- you can’t go back (dashboard will go red)



Distro defaults

- users, media eval the default config
- if bad, they move on

→ distro prioritizes defaults when testing



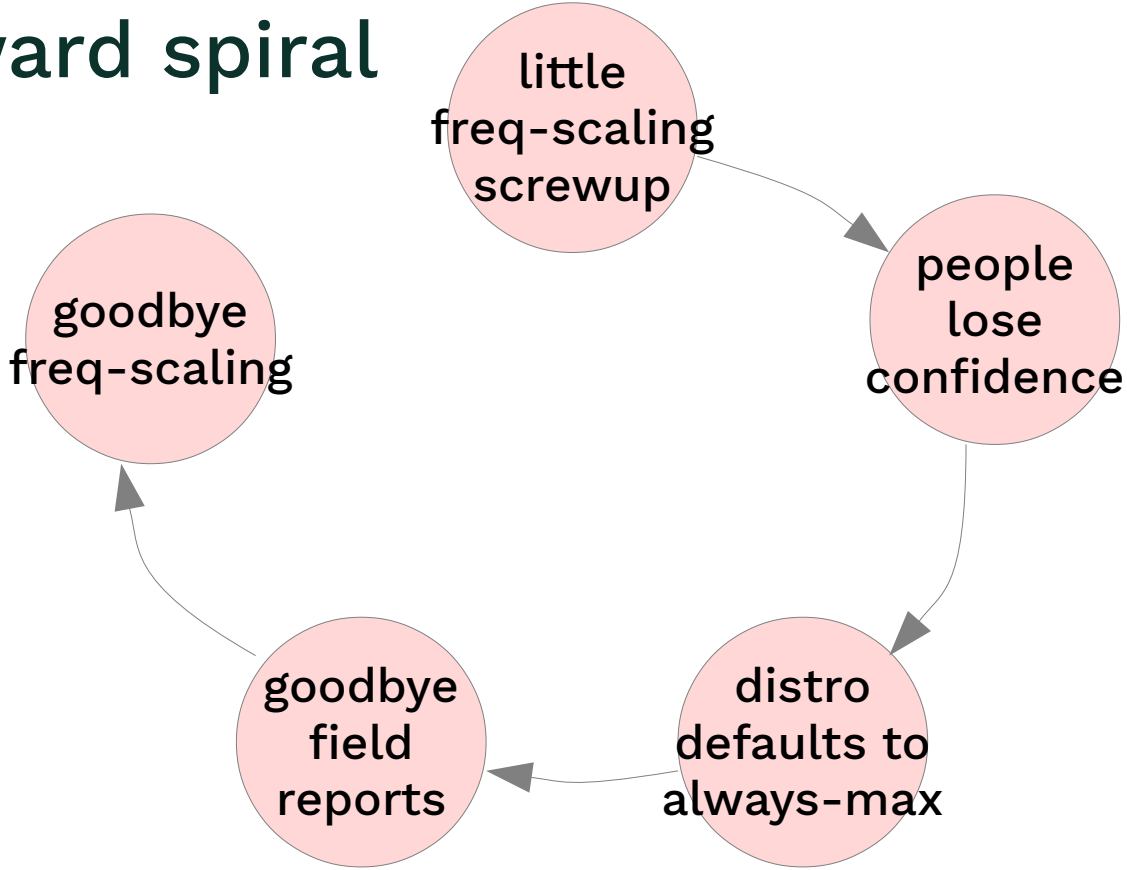
Regress is not an option

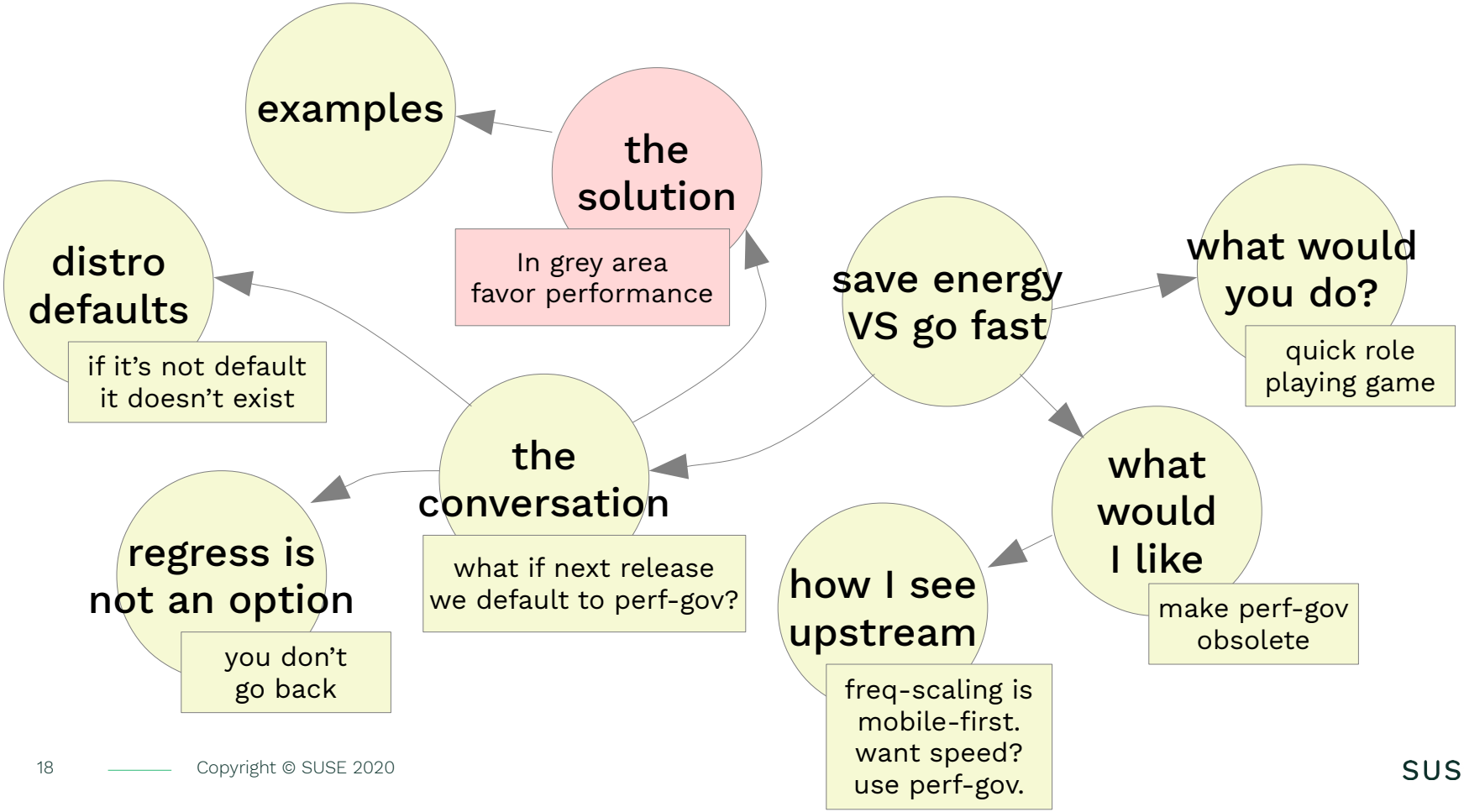
- users, media compare with previous release
- a regression is hard to sell

if switch powersave → perf-gov, it’s gone

(tests are often perf, not perf-per-watt)

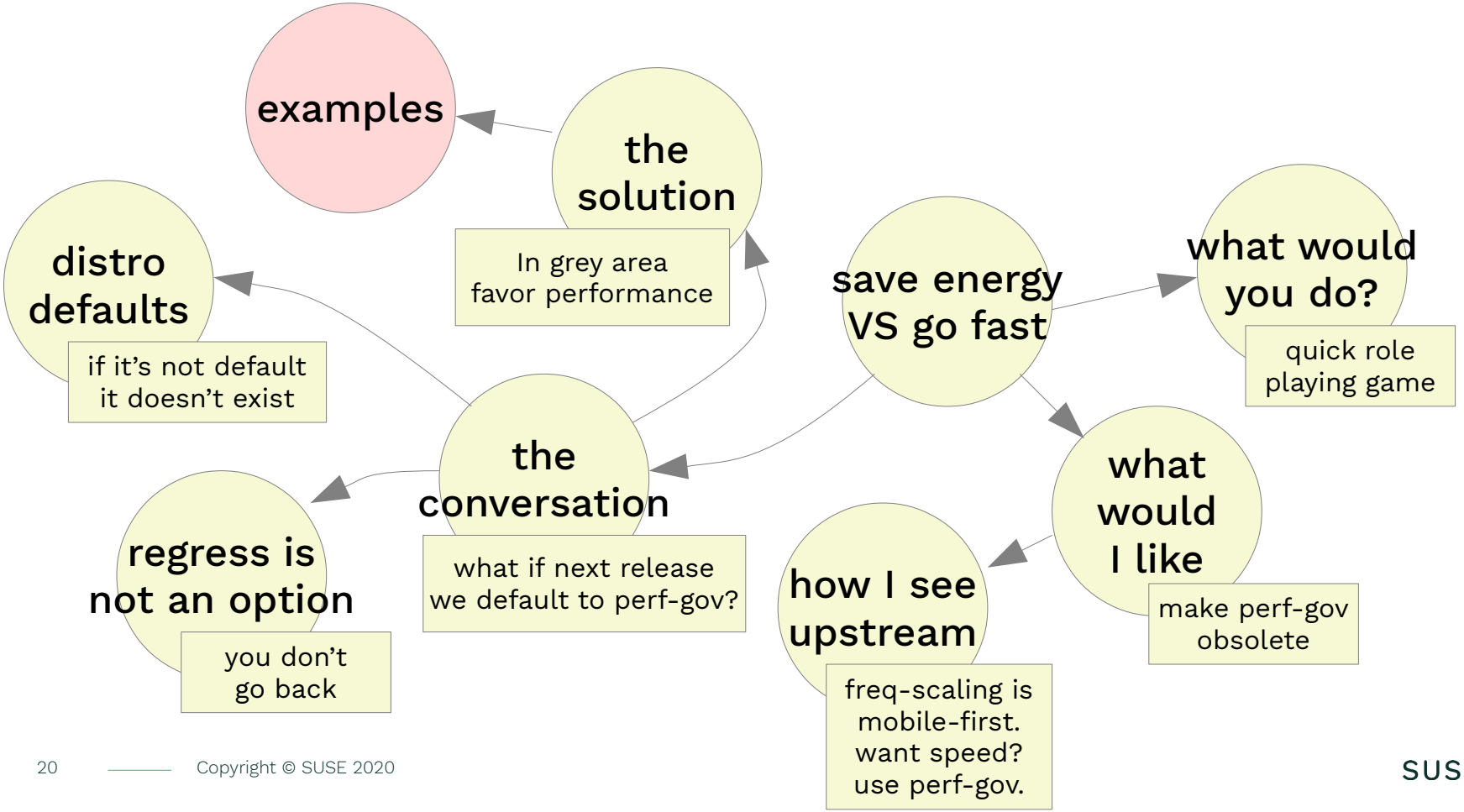
Downward spiral





In grey area favor performance

for the greater good



ioboost less aggressive (v5.1)

```
author      Rafael J. Wysocki <rafael.j.wysocki@intel.com>  2019-02-07
committer   Rafael J. Wysocki <rafael.j.wysocki@intel.com>  2019-02-18
commit      b8bd1581aa6110eb234c0d424eccd3f32d7317e6

cpufreq: intel_pstate: Rework iowait boosting to be less aggressive

The current iowait boosting mechanism in intel_pstate_update_util()
is quite aggressive, as it goes to the maximum P-state right away,
and may cause excessive amounts of energy to be used, which is not
desirable and arguably isn't necessary too.

Follow commit a5a0809bc58e ("cpufreq: schedutil: Make iowait boost
more energy efficient") that reworked the analogous iowait boost
mechanism in the schedutil governor and make the iowait boosting
in intel_pstate_update_util() work along the same lines.

Signed-off-by: Rafael J. Wysocki <rafael.j.wysocki@intel.com>
```

SLES out-of-tree patches

- **“idle boost”:** temporarily boost P-state when exiting from idle (similar to how it's boosted if a task has blocked recently for IO).
- **“ramp up faster”:** Ramp up frequency faster when utilisation reaches threshold.

Overview

- 40 cores (80 threads) Broadwell (2014)
- two sockets, memory 512G, SSD storage
- kernel v5.7-rc5
- XFS filesystem

BENCHMARK					UNIT	BETTER IF
Performance Ratios						
dbench	1	0.91	0.89	0.89	TIME_MSECONDS	lower
kernbench	1	1	0.93	0.93	TIME_SECONDS	lower
Performance-per-Watt Ratios						
dbench	1	0.77	0.76	0.76	OPS_PER_SECOND_PER_WATT	higher
kernbench	1	1	1	0.99	OPS_PER_SECOND_PER_WATT	higher

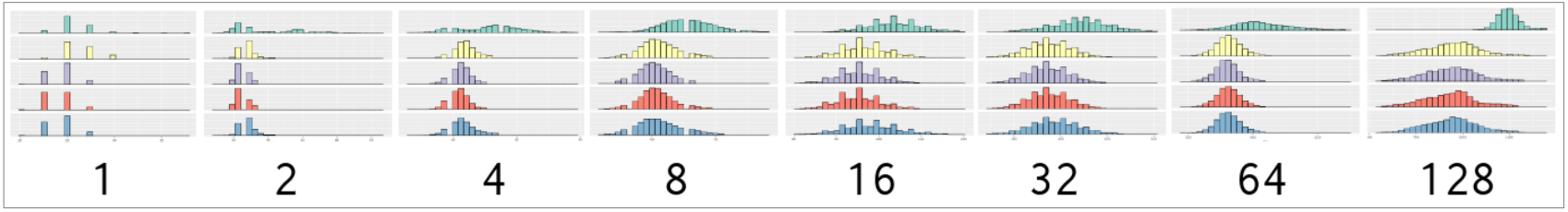
POWERSAVE (BASELINE)
REVERT-WEAK-IOBOOST
SPICY-POWERSAVE
PERFORMANCE

dbench

revert-weak-ioboost: revert b8bd1581aa61 (“cpufreq: intel_pstate: Rework iowait boosting to be less aggressive”)
spicy-powersave : revert-weak-ioboost + idleboost + ramp-up-faster

UNIT : TIME_MSECONDS
PARAM : CLIENTS
LOWER is better

	5.7.0-rc5 powersave		5.7.0-rc5 revert-weak-ioboost			5.7.0-rc5 spicy-powersave			5.7.0-rc5 performance		
1	30.20	±13.9%	29.12	±8.2%	(3.59%)	27.82	±5.0%	(7.88%)	27.65	±4.5%	(8.46%)
2	34.63	±14.9%	32.46	±9.0%	(6.27%)	31.52	±6.9%	(8.98%)	31.45	±8.2%	(9.20%)
4	45.99	±14.8%	42.44	±12.4%	(7.72%)	41.54	±13.0%	(9.67%)	41.17	±11.3%	(10.48%)
8	66.23	±16.2%	61.78	±17.4%	(6.72%)	60.99	±17.8%	(7.91%)	61.19	±17.9%	(7.61%)
16	104.54	±16.6%	97.46	±17.5%	(6.78%)	96.54	±17.0%	(7.65%)	96.15	±17.2%	(8.03%)
32	191.79	±17.7%	178.55	±17.6%	(6.91%)	177.95	±17.5%	(7.22%)	178.25	±17.9%	(7.06%)
64	424.50	±15.4%	375.74	±15.1%	(11.48%)	372.76	±14.5%	(12.19%)	374.76	±15.0%	(11.72%)
128	1278.89	±9.7%	982.24	±19.2%	(23.20%)	946.37	±16.0%	(26.00%)	936.16	±15.5%	(26.80%)



kernbench

```
revert-weak-ioboost: revert b8bd1581aa61 (“cpufreq: intel_pstate:
                     Rework iowait boosting to be less aggressive”)
spicy-powersave    : revert-weak-ioboost + idleboost + ramp-up-faster
```

```
UNIT  : TIME_SECONDS
PARAM : THREADS
LOWER   is better
```

	5.7.0-rc5 powersave		5.7.0-rc5 revert-weak-ioboost			5.7.0-rc5 spicy-powersave			5.7.0-rc5 performance		
2	481.81	±0.50%	480.52	±0.54%	(0.27%)	447.76	±0.63%	(7.07%)	440.83	±0.55%	(8.51%)
4	268.92	±2.86%	263.08	±0.24%	(2.17%)	240.30	±0.56%	(10.64%)	236.57	±0.50%	(12.03%)
8	147.46	±1.00%	146.91	±1.32%	(0.38%)	133.07	±0.71%	(9.76%)	131.29	±0.31%	(10.96%)
16	81.38	±0.44%	82.00	±1.19%	(-0.77%)	75.08	±0.32%	(7.74%)	74.44	±0.67%	(8.52%)
32	47.83	±1.22%	47.91	±1.13%	(-0.17%)	45.72	±1.18%	(4.41%)	45.02	±2.00%	(5.89%)
64	34.49	±1.17%	34.29	±1.70%	(0.58%)	33.03	±1.55%	(4.22%)	33.41	±1.28%	(3.14%)
128	33.55	±0.56%	33.43	±0.51%	(0.38%)	32.18	±0.41%	(4.09%)	32.12	±0.16%	(4.26%)
160	33.61	±0.39%	33.52	±0.72%	(0.27%)	32.26	±0.18%	(4.03%)	32.21	±0.34%	(4.16%)

A bit more of this please?

When it comes down to a **judgment call**, favoring performance over energy saving may **keep frequency scaling alive** in the server world

photo by Roman Avdagić, Foto-Škrinja j.d.o.o.
https://www.flickr.com/photos/romanski/44296560680
license: CC-BY 2.0 https://creativecommons.org/licenses/by/2.0
modified to include quote