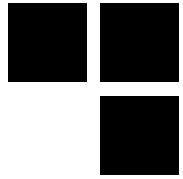
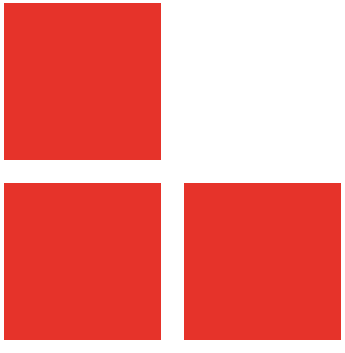
SYNACKTIV
DIGITAL SECURITY

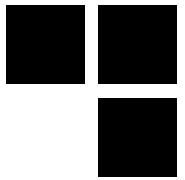


Performance Optimization – general guidelines

simon.marechal@synacktiv.com

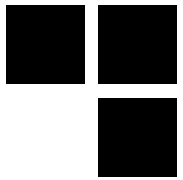
2021/2022





Introduction

Start with the basics



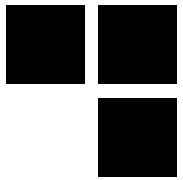
Make it work, make it right, make it fast.

– Kent Beck

Better than the famous Dijkstra quote!

- get a baseline that you are confident will work
- find a way to check there are no regressions during optimization
  - when working on someone else's code, it is easy to miss invariants
  - your future self is someone else :)

Architecture and system design



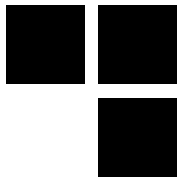
It is important not to optimize too much during design

- makes refactoring, changes, more costly
- wasted work

However, it is important to keep performance in mind!

- architectural choices
- it is good to have an feeling on how fast it will be in the end
- it is OK to work in a high level language, as long as you can get low level

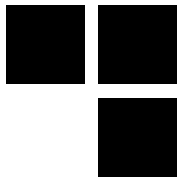
Measure first



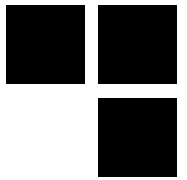
Always measure before optimizing!

- the compiler might have done a better job already;
- the hot path might not be where you expect it;
- the algorithmically sensible decision might turn out to be a disaster in practice (small n, high constant factors)

Make the slow parts faster :)

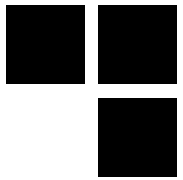


- go for the parts that eats up most of the computation time first
  - however that might be normal
  - that is why you should have an idea on how fast things should be first



Minimal tool set for program optimization

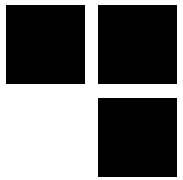
Micro-benchmarks



A test suite that measure the speed of a small part of your program

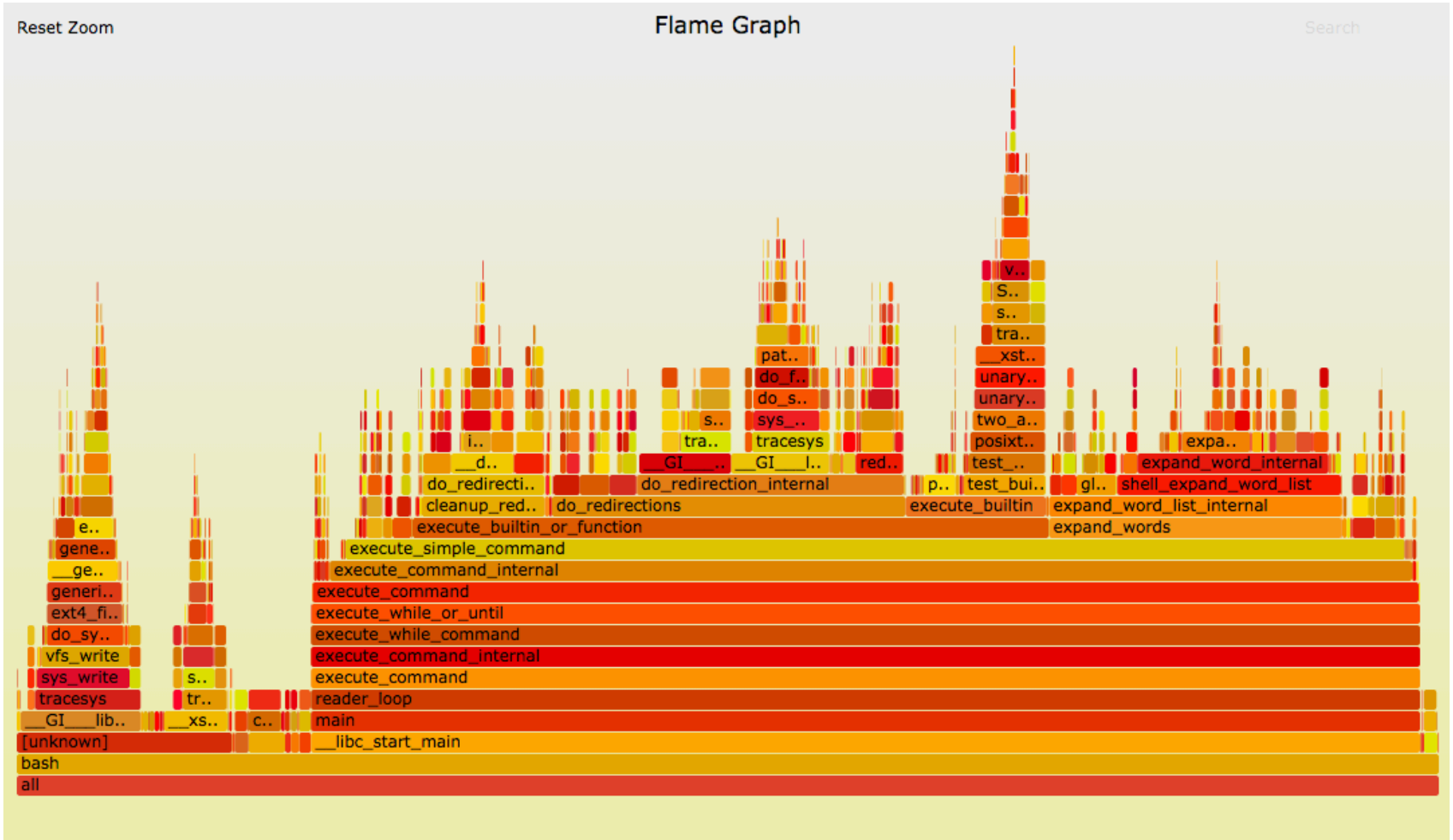
- kind of artificial, high variance between runs is expected
  - 5-15% on a typical computer
- the test framework must:
  - warm-up the CPU
  - make sure the algorithm is not optimized away
  - display distributions
- write benches with several input sizes
- Click here for a good example

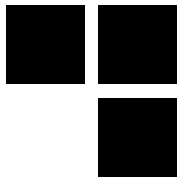
Execution traces



- realistic measurements
- however, many factors can muddle the conclusions
  - dependencies, such as storage, databases, etc.
  - can be hard to interpret

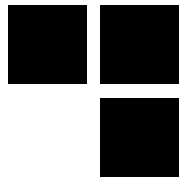
Flame graphs





Other concerns

Talking with the kernel



System calls are very expensive!

- reduce them (ie. techniques for caching the current time)
- zero-copy tools
- batch them (io-uring techniques)

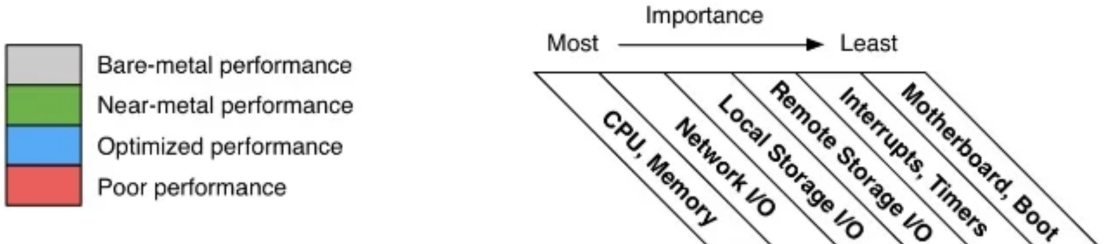
VMs and containers impose a tax on everything, although it tends to be less and less expensive

Note that most large orgs have a dedicated kernel team to just *make it work*

- super common systems like TCP are not tuned anymore

Hypervisor performance by generation

VM Improvements



Importance: Most → Least

	#	Tech	Type	With	CPU, Memory	Network I/O	Local Storage I/O	Remote Storage I/O	Interrupts, Timers	Motherboard, Boot
	1	VM	Fully Emulated		VS	VS	VS	VS	VS	VS
Old	2	VM	Xen PV 3.0	PV drivers	P	P	P	P	VS	VS
	3	VM	Xen HVM 3.0	PV drivers	VH	P	P	P	VS	VS
	4	VM	Xen HVM 4.0.1	PVHVM drivers	VH	P	P	P	P	VS
	5	VM	Xen AWS 2013	PVHVM + SR-IOV(net)	VH	VH	P	P	P	VS
	6	VM	Xen AWS 2017	PVHVM + SR-IOV(net, stor.)	VH	VH	VH	P	P	VS
	7	VM	AWS Nitro 2017		VH	VH	VH	VH	VH	VS
New	8	HW	AWS Bare Metal 2017		H	H	H	H	H	H
			Bare Metal		H	H	H	H	H	H

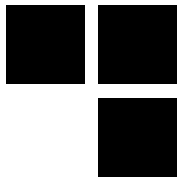
VM: Virtual Machine. HW: Hardware.
VS: Virt. in software. VH: Virt. in hardware. P: Paravirt. Not all combinations shown.
SR-IOV(net): ixgbe/ena driver. SR-IOV(storage): nvme driver.

http://www.brendangregg.com/blog/2017-11-29/aws-ec2-virtualization-2017.html

LISA21 Computing Performance: On the Horizon (Brendan Gregg)

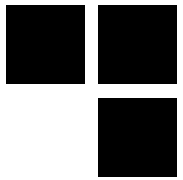
Source: [Gregg 17]

Local system performance



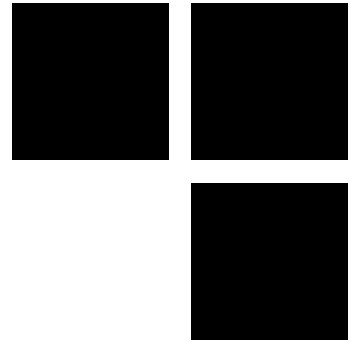
- for storage dependent elements, use the fastest storage you can afford and **never** use RAID5
- scheduling makes all the difference ...
- the performance of many systems depend on a centralized persistent store, like a database
  - learn how to optimize queries, table representation, indexes
  - investigate all the tunable settings (database and kernel)
  - if at all possible, avoid virtualizing it!

Whole system performance



- everything must be tuned
- observability is *really* hard
- look at everything Brendan Gregg does

QUESTIONS?



Thank you for your attention

SYNACKTIV
DIGITAL SECURITY