bBench: A Performance Benchmark for Blockhain Applications

Nuno Laranjeiro

cnl@dei.uc.pt

The current blockchain team at **Coimbra**

- **Faculty Members**
  - Naghmeh Ivaki - naghmeh@dei.uc.pt
  - Nuno Laranjeiro - cnl@dei.uc.pt
- **PhD Students**
  - Fernando Vidal
  - Sadaf Azimi
  - Ali Golzari
- **MSc Students**
  - Michelangelo Formato, Univ. of Napoli, Federico II
  - Alessandro Cavaliere, Univ. of Salerno
- **Former Students**
  - Bruno Dias (MSc)
  - Maria Viegas (MSc)

1 2 9 0

UNIVERSIDADE D COIMBRA

Main topics

1) Study and systematization of smart contract vulnerabilities
2) Assessment of smart contract vulnerability detection tools
3) Development of a vulnerability detection tool
4) Automated execution of blockchain transaction revocation models
5) **Benchmarking blockchain applications**

Blockchain applications

- Highly decentralized and complex nature of the entire system
- Immutability of data generated by smart contracts
- Distributed nature of the ledger where this data is stored
- Costs associated with running a blockchain application (e.g., gas fees or the effective cost of executing a transaction

- **Very challenging to fully assess the performance/behavior of a blockchain application**

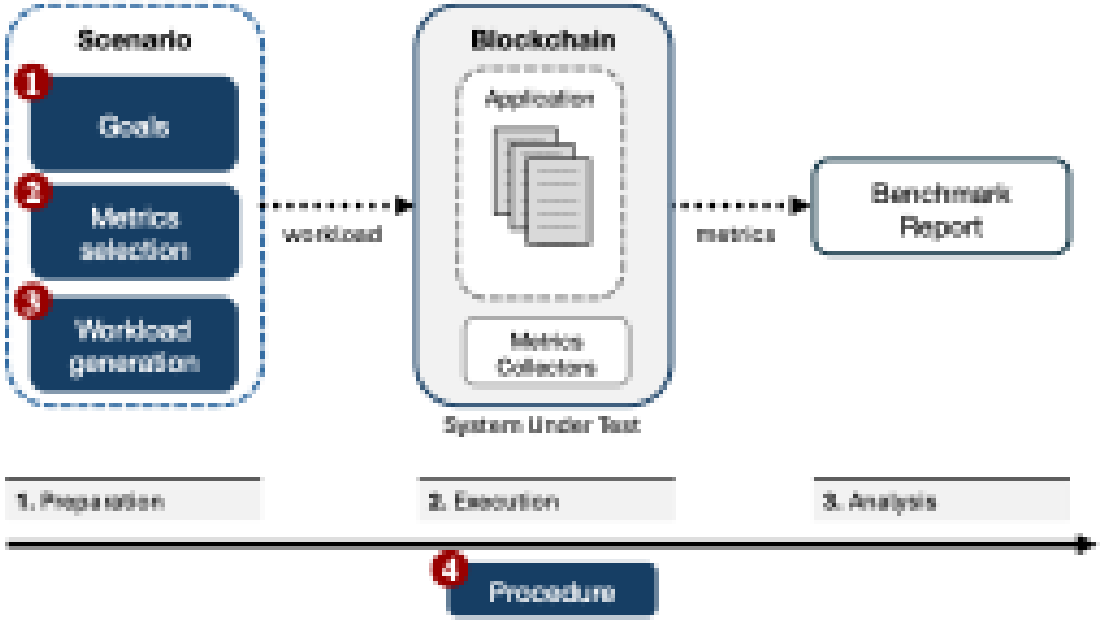
State of the art

- Limited sets of metrics / reporting
  - Many works report usual metrics (e.g., throughput, latency) and not blockchain-specific metrics
- Limited configurability
  - General network emulation
  - No tuning of blockchain specific behaviors, e.g., transaction cost dynamics (transaction data size, contract size/complexity)
- Little support for workload generation

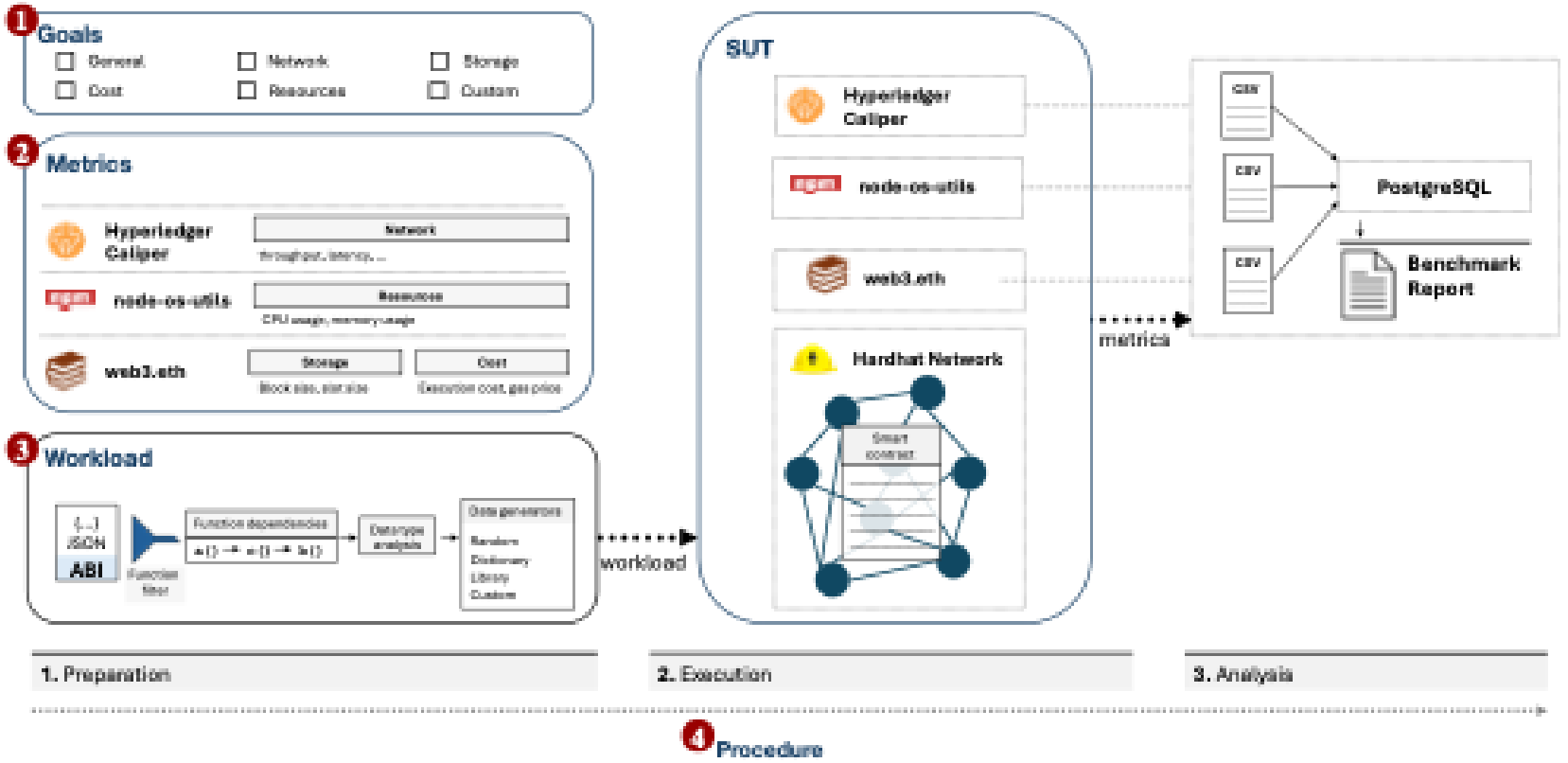
Our proposal: bBench

- Build on established general concepts from performance benchmarking, e.g., workload, metrics
- Consider the blockchain specificities, e.g., gas, ledger space
- Use state of the art tooling (modify as needed)
- Report across relevant groups of metrics
  - Network behavior
  - Computational resource consumption
  - Storage usage
  - Operational cost

Conceptual design of the benchmark



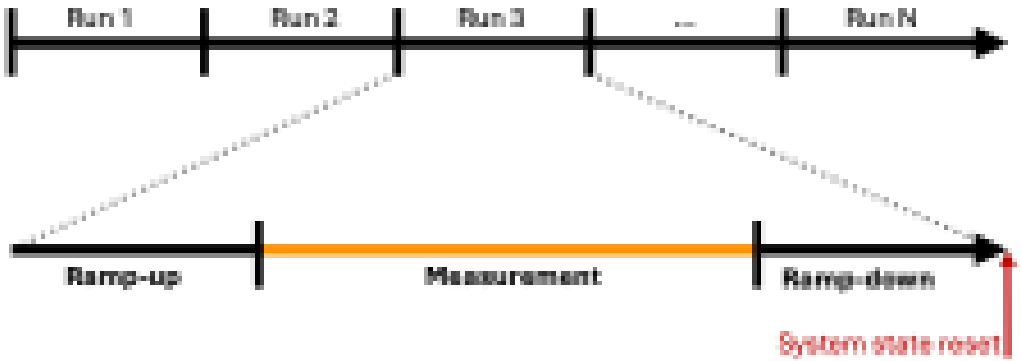
Benchmark implementation



Group	Metric	Unit	Formula	Source	Reference
Network	**Throughput**	Transactions commited per second	committed transactions / period in seconds	Caliper	throughput: Duan et al. (2020); Dinh et al. (2023). peak transaction throughput: Gramoli et al. (2023); Nasrulin et al. (2022)
	Latency	Miliseconds	sum(latency-individual) / number of committed transactions	Caliper	latency distribution over time: Gramoli et al. (2023). average latency: Gramoli et al. (2023). latency: Duan et al. (2020); Dinh et al. (2023); Yue et al. (2023) . serverLatency:Touloupou et al. (2022)
	Committed Transactions	Percentual	committed transactions / total transactions	Caliper	emit rate: Rasolroveicy et al. (2024). commit timeouts: Klenik et al (2022)
	Committed Consensus	Percentual	committed transactions / verified transactions (validated by consensus mechanisms)	Web3.Eth	proportion of commited: Gramoli et al. (2023). endorsement timeouts: Klenik et al (2022) .
Resource	**CPU Usage**	Percentual	avg(cpu usage) per individual transaction	node-os-utils	resource utilization: Rasolroveicy et al. (2024).
	Memory Usage	MegaByte	Sum(mem_end - mem_start) per individual transaction	node-os-utils	consumption: Saingre et al (2020)
Storage	**Block Size**	Bytes	avg (block sizes generated in the experiment) per individual transaction	Web3.Eth	—
	State Size	Bytes	avg(memory used by all declared variables in the contract) per individual transaction	Web3.Eth	storage usage: Yue et al. (2023);
Cost	**Gas Price**	Wei	avg(gas price) per individual transaction	Web3.Eth	gas consumption: Rasolroveicy et al. (2024).
	Execution Cost	Ether	sum (balance_end - balance_start) per individual transaction	Web3.Eth	—

Case study

- Three smart contracts, each with three versions



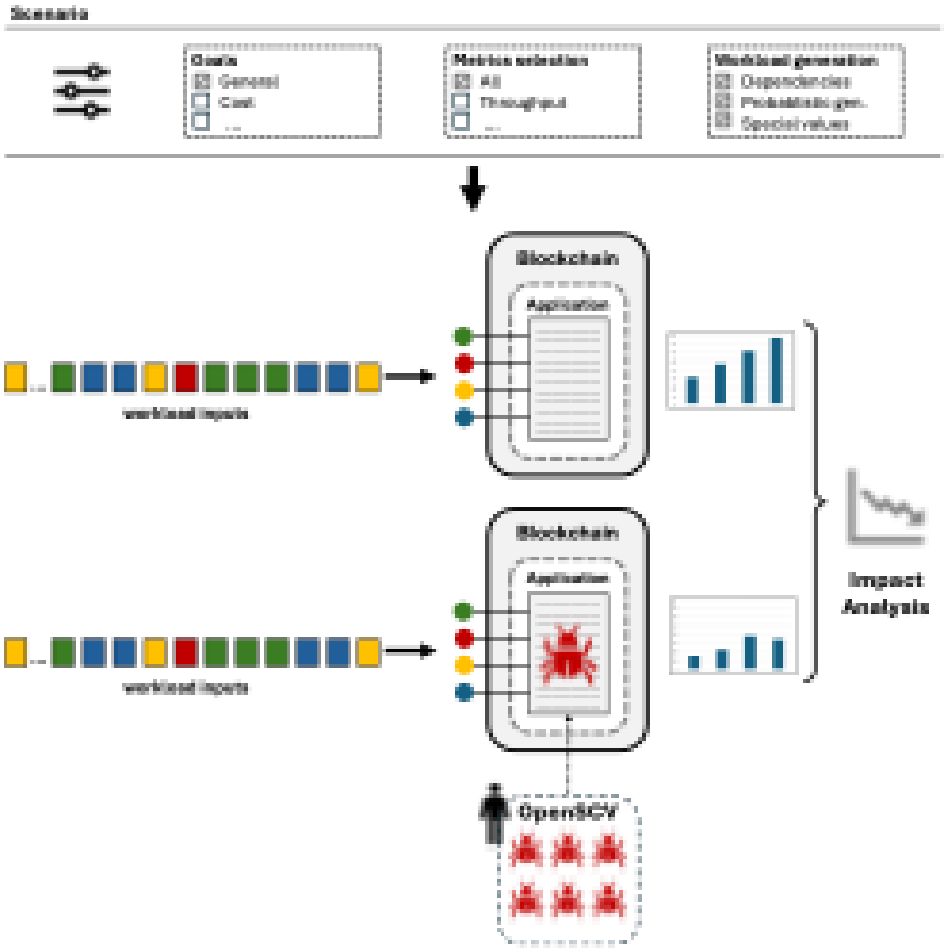
dApp	ID	Operation	Return type	Modifier	Payable
CLASS-V0 Class Attendance Management System	1	setEnroll (uint _roll, uint _year)	void	Owner	
	2	createStudent (uint _studId, uint _age,string memory _fName, string memory _lName, address _aStud)	void	-	
	3	createTeacher (uint _teachId,string memory _fName, string memory _lName, string memory _discipline, address _aTeach)	void	-	
	4	incrementAttendance (address _aTeach, address _aStud)	void	Teacher	
	5	getStudents ()	object list	Teacher	
	6	getParticularStudent ()	object	Student	
	7	getTeacherList ()	object list	Teacher	
	8	addHistory (address _aStud, address _aTeach, string memory _comment)	void	Teacher	
EHR-V0 Electronic Health Record Blockchain	1	setInfo (string firstName, string lastName, string IID, string bdate, string email, string phone, string zip, string city, string encryption_key)	void	Owner	
	2	start_visit (address _unique_id,uint _time)	string	Owner	
	3	addDoctors (address _doctor_address)	string	Owner	
	4	addAudits (address _audit_address)	string	Owner	
	5	doctor_print_record (address _unique_id)	array	Doctor	
	6	doctor_query_record (address _unique_id)	array	Doctor	
	7	doctor_update_record (address _unique_id)	array	Doctor	
	8	doctor_delete_record (address _unique_id)	array	Doctor	
	9	get_record_details (address _unique_id)	string	Patient	
ROOM-V0 Room Renting	1	setReserveRoom ()	void	-	Y
	2	setAddDaysToPay (uint256 _amount, uint8 _qtdDay)	void	-	
	3	getCurrentBill ()	int	-	
	4	getCurrentDay ()	int	-	
	5	getDiscount ()	int	-	
	6	setReleaseRoom ()	void	-	Y

Vulnerable versions

- Fernando Vidal, Naghmeh Ivaki, Nuno Laranjeiro. OpenSCV: an open hierarchical taxonomy for smart contract vulnerabilities. Empirical Software Engineering 29, 101 (2024)
- **https://openscv.dei.uc.pt**

Name	Target Operation	Injected Vulnerability	Expected Impact
CLASS-V1	addHistory	5.16 Wrong Logic	Storage
CLASS-V2	createStudent	5.4.2 Wrong Selection of Guard Function	Cost
CLASS-V3	getTeacherList	8.2.1 Expose Private Data	Network
EHR-V1	printRecord	5.7.2 No effect code execution	Cost
EHR-V2	printMyRecord	8.1.2 Owner Manipulation	Network
EHR-V3	createPatientID	5.13.3 Read from Arbitrary Storage Location	Unknown
ROOM-V1	releaseRoom	5.4.2 Wrong Selection of Guard Function	Cost
ROOM-V2	toString	5.7.2 No effect code execution	Cost
ROOM-V3	addDaysToPay	7.1.2 Integer Overflow	Cost

Main idea


Scenario
Goals
Metrics selection
Workload generation
Blockchain
Application
workload inputs
Blockchain
Application
workload inputs
Impact Analysis
OpenSCV

Some highlights — CLASS

- **CLASS-V3 (8.2.1 Exposed Private Data vulnerability)**
- Highest throughput, lowest latency, and also highest committed transaction rate. Light functions were made public
- Highest execution cost (additional data being manipulated, leading to higher Ether consumption)

dApp	Network								Resources				Storage				Cost			
	Throughput (Tx/s)	RD	Latency (ms)	RD	Committed Transactions (%)	RD	Committed Consensus (%)	RD	CPU(%)	RD	Mem(MB)	RD	State Size (Bytes)	RD	Block Size (Byte)	RD	Exe. Cost (ETH)	RD	Gas Price (ETH)	RD
CLASS-V0	344,15	–	29,29	–	70%	–	100%	–	82,54	–	1172,81	–	16800	–	349957	–	1,91	–	1E-09	–
CLASS-V1	342,90	–	28,18	–	61%	-0,14	100%	–	82,56	–	947,59	-0,19	13200	-0,21	489663	0,40	2,18	0,14	1E-09	–
CLASS-V2	351,83	–	25,71	-0,12	64%	–	100%	–	82,42	–	1154,83	–	16800	–	443010	0,27	2,36	0,24	1E-09	–
CLASS-V3	355,86	–	20,71	-0,29	80%	0,14	100%	–	81,71	–	1243,34	–	16800	–	469579	0,36	3,10	0,62	1E-09	–

Some highlights — EHR

- **EHR-V3 (5.13.3 Read from Arbitrary Storage vulnerability)**
- Unrestricted access to array indices resulted in less in-memory storage being used (smaller state size).
- The activation of the vulnerability led to numerous invalid references (e.g., non-existent patient IDs)

dApp	Network								Resources				Storage				Cost			
	Throughput (Tx/s)	RD	Latency (ms)	RD	Committed Transactions (%)	RD	Committed Consensus (%)	RD	CPU(%)	RD	Mem(MB)	RD	State Size (Bytes)	RD	Block Size (Byte)	RD	Exc. Cost (ETH)	RD	Gas Price (ETH)	RD
EHR-V0	370,79	–	30,00	–	77%	–	100%	–	78,05	–	5980,95	–	24200	–	1855292	–	9,46	–	1E-09	–
EHR-V1	367,94	–	30,00	–	80%	–	100%	–	75,27	–	5536,19	–	24200	–	2074288	0,12	11,85	0,17	1E-09	–
EHR-V2	363,02	–	30,00	–	79%	–	100%	–	76,62	–	5652,04	–	24200	–	1697767	–	9,02	–	1E-09	–
EHR-V3	369,03	–	30,00	–	80%	–	100%	–	76,78	–	5449,38	–	19800	-0,18	1904775	–	8,32	-0,12	1E-09	–

Some highlights — ROOM

- **ROOM-V3 (7.1.2 Integer Overflow vulnerability)**
- Many more committed transactions (no limit check)
- More gas consumption and more processing time
  - transactions are finalized instead of being early reverted

dApp	Network								Resources				Storage				Cost			
	Throughput (Tx/s)	RD	Latency (ms)	RD	Committed Transactions (%)	RD	Committed Consensus (%)	RD	CPU(%)	RD	Mem(MB)	RD	State Size (Bytes)	RD	Block Size (Byte)	RD	Exc. Cost (ETH)	RD	Gas Price (ETH)	RD
ROOM-V0	334,56	–	22,73	–	10,3%	–	100%	–	77,56	–	4589,80	–	2200	–	2151103	–	20,08	–	1E-09	–
ROOM-V1	330,23	–	27,00	0,19	10,1%	–	100%	–	75,60	–	4361,85	–	2000	–	2204843	–	25,54	0,27	1E-09	–
ROOM-V2	355,70	–	24,00	–	8,1%	-0,22	100%	–	77,15	–	4595,30	–	2000	–	2748311	0,28	24,42	0,22	1E-09	–
ROOM-V3	369,90	0,11	24,00	–	95,8%	8,26	100%	–	75,71	–	4577,60	–	2000	–	2709086	0,26	28,17	0,40	1E-09	–

Conclusion and future work

- Further experimentation with diferent types of contracts
- Usability of the tool is undergoing
- bBench –- soon available at https://blockchain.dei.uc.pt

Questions?

Nuno Laranjeiro
cnl@dei.uc.pt