



Università degli Studi di Cagliari
Facoltà di Scienze
Corso di Laurea Magistrale in Informatica
A.A. 2021/22

Simulatore dApp per sistemi di tracciamento di processi produttivi in campo agroindustriale (progetto ABATA)

PROGETTO E SVILUPPO DI APPLICAZIONI BLOCKCHAIN

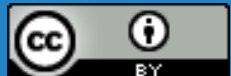
Studente

PISEDDU Enrico

matr. 60/73/65222

Docente del Corso

Prof. MARCHESI Michele



Quest'opera è distribuita con Licenza Creative Commons Attribuzione 3.0 Italia. Maggiori informazioni al link: <https://creativecommons.org/licenses/by/3.0/it/>

Il progetto ABATA (1/2)

➤ Applicazioni della Blockchain per l'Autenticità e la Tracciabilità Alimentare

Obiettivo

- Tracciabilità della catena produttiva di prodotti agrifood
- Certificazione dei prodotti stessi

Strumenti

- Sistemi classici (DBMS, DMS, App)
- Blockchain e Smart Contract

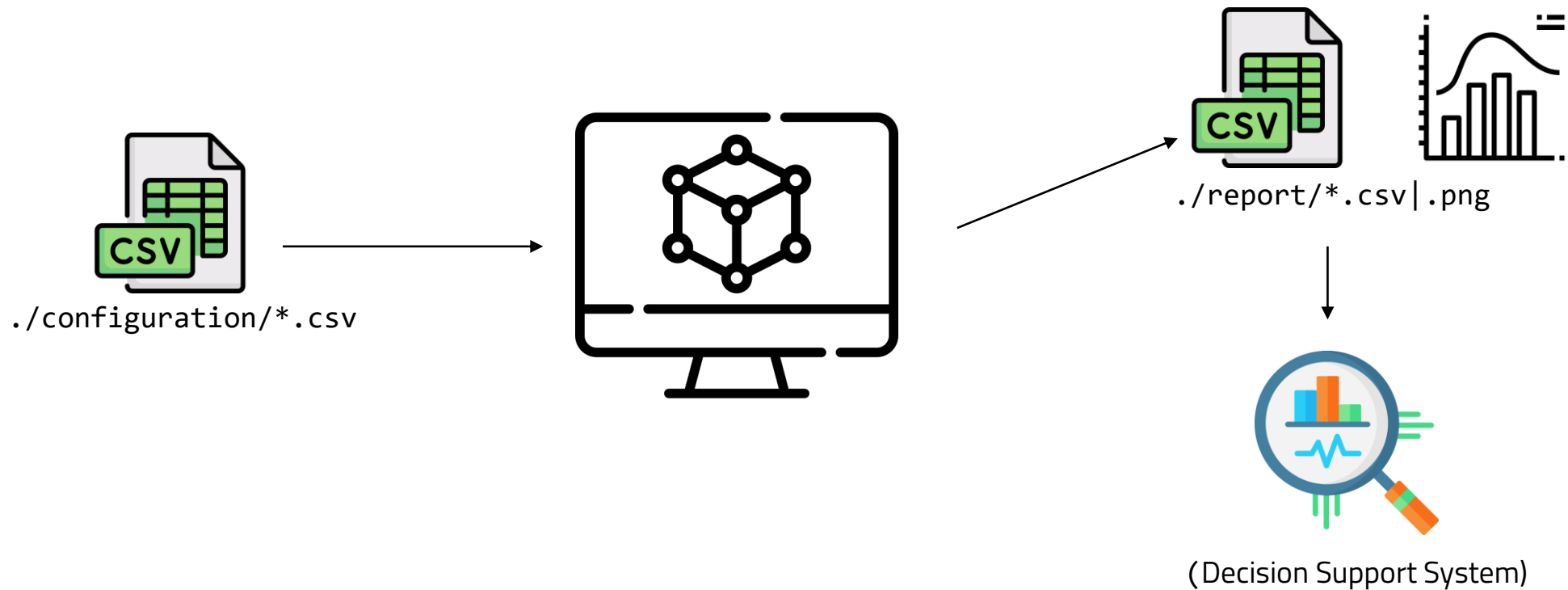


Il progetto ABATA (2/2)

- **Due macro casi d'uso che interessano tipi di attori differenti:**
 - **OPERATORI** che generano transazioni al fine di memorizzare i dati dei prodotti agrifood
 - **CONSUMATORI** che, in un secondo momento, intendono verificare l'autenticità e l'integrità dei prodotti che acquistano o che vorrebbero acquistare

L'idea del simulatore ABATA

- Sviluppo di un programma che simuli il sistema informatico di ABATA, al fine di valutarne le **prestazioni** in funzione del numero di attori coinvolti, indagare su eventuali colli di bottiglia etc.



Oggetti del simulatore

- Sviluppo del simulatore con un approccio ad **oggetti**:
 - Componenti
 - Attori (Operatori e Consumatori)
 - Transazioni degli attori
 - Eventi (legati ad una transazione)

I componenti

➤ I principali **componenti** del sistema sono:

➤ **App** (Operator and Navigator App)

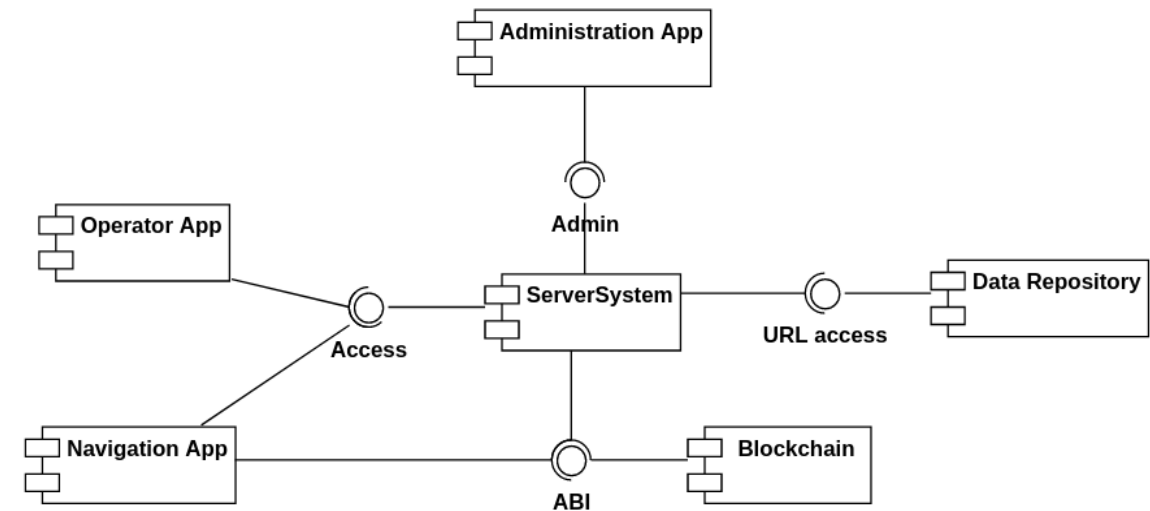
➤ **DBMS**

➤ **DMS (Document Management System)**

➤ **DLT**

➤ Blockchain quando viene utilizzata in modalità «**write**»

➤ Blockchain quando viene utilizzata in modalità «**read only**»



Le caratteristiche dei componenti

- Ogni componente ha delle **proprietà**, descritte in un file `.csv`

	A	B	C	D	E	F	G
1	name, avg. delay (ms), st.dev. delay (ms), max txs, wait when full (ms)						
2	app	3000	1000	10000000	500		
3	appSys	200	80	300	500		
4	dlt	2500	1000	250	2500		
5	dltview	700	200	800	500		
6	dbms	300	150	300	200		
7	dms	500	400	1000	1000		
8							

- **N.B. :** Tali valori sono altamente configurabili e dipendono dallo specifico componente!

Gli attori

- Gli attori si dividono in due categorie:
 - **Operatori**
 - **Consumatori**
- Essi interagiscono con i componenti inviando delle transazioni
- Gli operatori generano principalmente transazioni di scrittura dati, i consumatori, invece, generano transazioni di lettura (es. lettura da Blockchain, DMS...)

Le caratteristiche degli attori

- In un apposito file `.csv`, è possibile definire alcuni parametri che interessano gli attori:

	A	B	C	D	E	F	G
1	Actors data: type, nr. of actors, nr. TXs/day, std.dev. TXs/day, schedule						
2	operators	500	50	30	OP1		
3	customers	50000	3	10	CU1		
4							

- Inoltre si definisce, per ogni tipologia di attore, lo **schedule**:

Attore \ Ora	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Operatore	10%	13%	13%	13%	7%	4%	12%	12%	12%	4%							
Consumatore		3%	6%	8%	8%	12%	6%	6%	8%	8%	8%	5%	10%	7%	5%		

- esso rappresenta la probabilità, per ogni ora della giornata, che una transazione inizi a quell'ora

Le transazioni

- Due tipologie di transazioni, descritte in un file .csv
 - **transazioni base:** esse coinvolgono un singolo componente
 - **transazioni complesse:** sono composte da almeno due transazioni base

TX data: name, nr. txs, % operators, % customers, avg blocks, s.d. blocks

For each txs: type, component, avg time (ms), std. dev. (ms)

Simple write,5,50,0,1,0

DLT_read,dlt,400,200

DB_read,dbms,300,100

Input,app,5000,3000

DB_write,dbms,600,300

DLT_write,dlt,800,300

Document write,6,50,0,1,0

DLT_read,dltview,400,200

DB_read,dbms,300,100

Input,app,30000,20000

Doc_upload,dms,1500,1000

DB_write,dbms,600,300

DLT_write,dlt,800,300

Simple read,2,0,50,4,2

DLT_read,dltview,300,100

Read,app,5000,4000

Document read,4,0,50,3,1

DLT_read,dltview,300,100

Read,app,3000,2000

Doc_download,dms,1000,800

Read,app,30000,20000

Esempio:

1 transazione complessa
composta da 6 transazioni base

La simulazione (1/3)

- Lettura dei dati di configurazione descritti nei file `.csv`
- Generazione dei **componenti**
- Generazione degli **attori** (n operatori ed m consumatori, dove $m \gg n$)
- Generazione, per ogni attore, delle **transazioni** (e rispettivo *timestamp* di inizio) utilizzando un generatore di numeri casuali (distribuiti secondo una log-normale)
- Inserimento delle transazioni in una «**coda**» ordinata secondo il *timestamp*

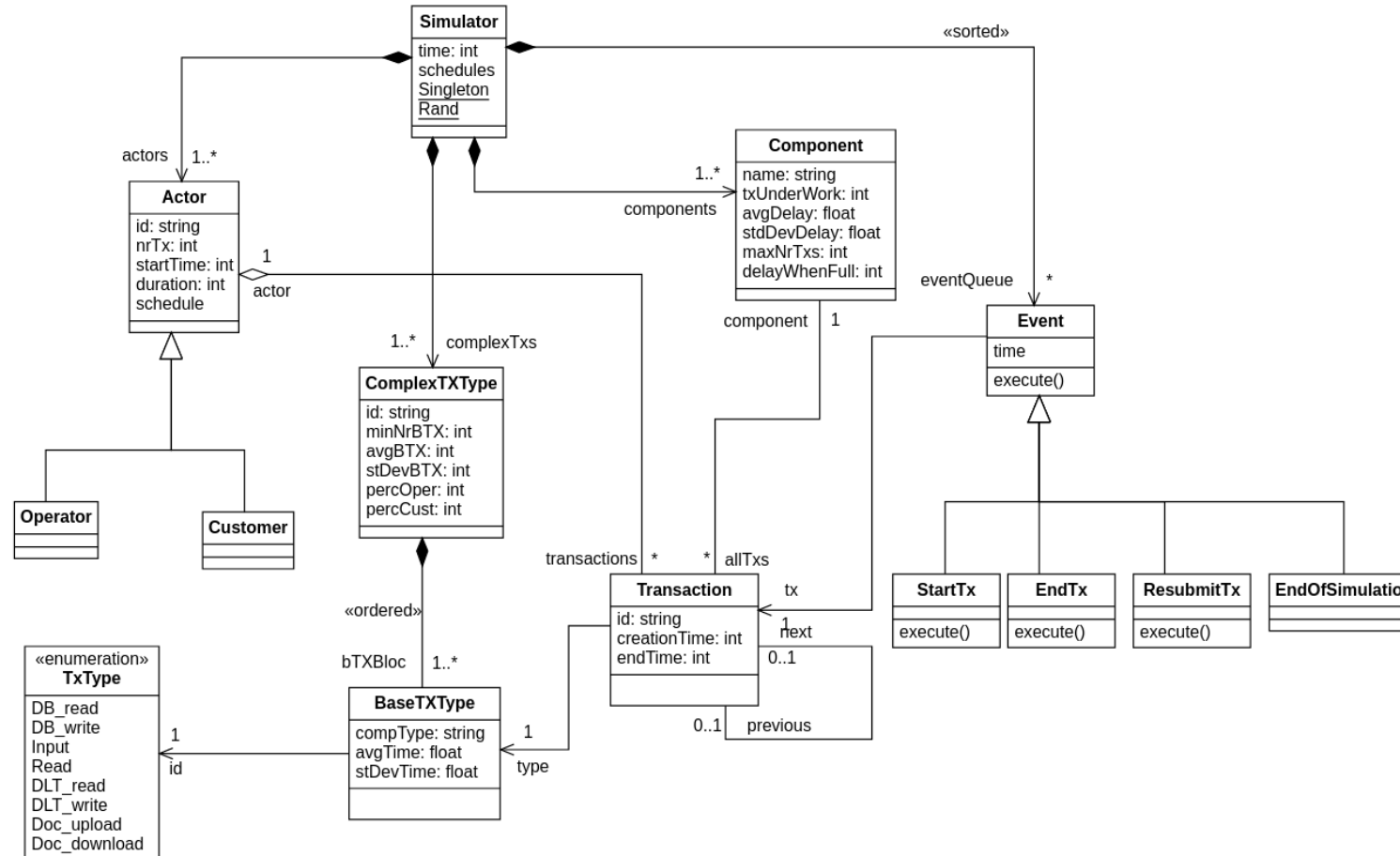
La simulazione (2/3)

- La simulazione procede con un approccio a coda di eventi (**Start** | **End** | **Resubmit** Event)
- Si estrae un evento dalla coda:
 - se l'evento è **StartTX**:
 - Se il componente accetta txs viene generato un evento EndTX, altrimenti un evento ResubmitTX con opportuno ritardo.
 - se l'evento è **EndTX**:
 - L'esecuzione di una tx termina. Se la TX è seguita da un'altra, si genera allo stesso tempo un evento StartTX.
 - se l'evento è **ResubmitTX**:
 - Si riprende l'esecuzione di una tx (causa componente pieno). Se il componente è ancora pieno si genera un'altra ResubmitTX, altrimenti si genera un evento di EndTX

La simulazione (3/3)

- La simulazione procede fintantoché la coda degli eventi è vuota
- Al termine di essa, gli oggetti generati (componenti, transazioni, eventi) risiedono ancora in memoria ed è possibile manipolarli al fine di estrarre informazioni utili.

Il modello ad oggetti del simulatore



Strumenti utilizzati

- Linguaggio **Python** e IDE **PyCharm**
- Libreria `rand` per la gestione dei numeri casuali
- Libreria `csv` per la lettura e scrittura dei rispettivi file
- Libreria `matplotlib` per la generazione e visualizzazione degli output

Output del simulatore

- Gli output vengono generati a partire dall'elaborazione degli attributi degli oggetti al termine della simulazione
- Tipologia di output: file `.csv`, grafici `.png` e informazioni sulla *console* dell'IDE
- Essi quantificano le *performance* dell'intero sistema al fine di trovare eventuali colli di bottiglia, rallentamenti dovuti a qualche componente etc.
- Tipicamente vengono eseguite più simulazioni facendo variare alcuni dati (es. numero di attori) e in seguito si confrontano gli output generati.

Output di una simulazione (1/2)

- **Per ogni componente:**

- Dati un tempo iniziale, finale ed un intervallo di tempo (configurabile):
 - NUMERO DI TXS INIZIATE E TERMINATE IN OGNI INTERVALLO
 - INTERVALLI NEI QUALI IL COMPONENTE HA RAGGIUNTO, EVENTUALMENTE, IL LIMITE MASSIMO DI TXS
- NUMERO DI TXS BASE/sec

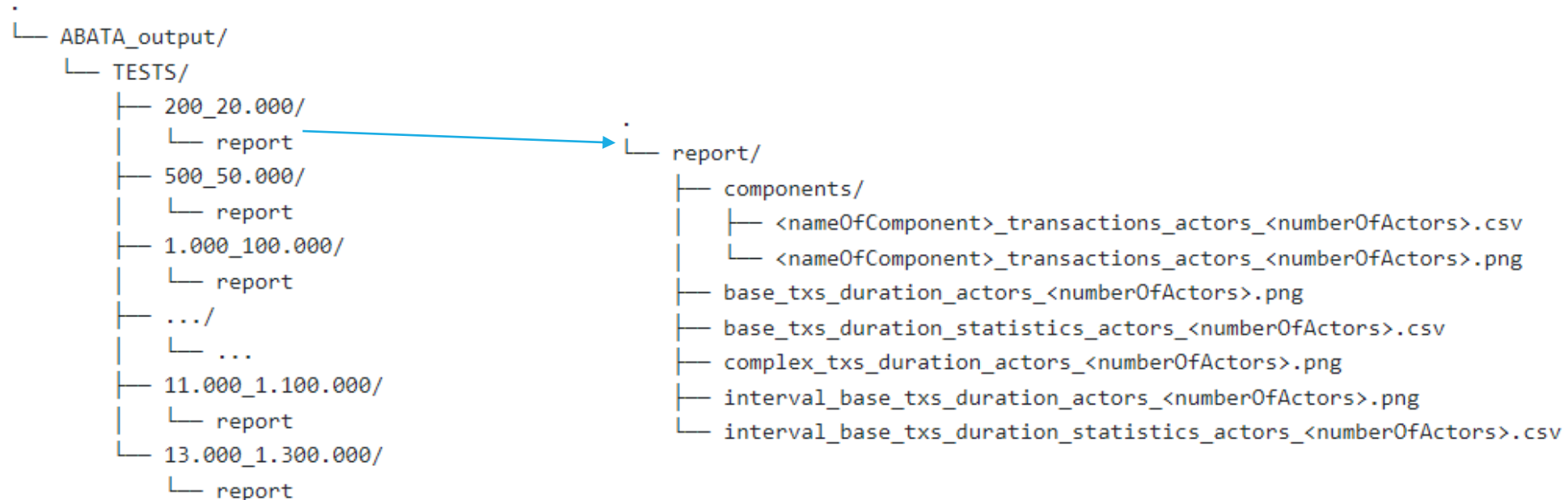
- **Per ogni tipo di transazione base:**

- STATISTICHE SULLA DURATA MEDIA, DEV. ST, CASO MIGLIORE E PEGGIORE
- STATISTICHE SPECIFICHE PER UN DATO INTERVALLO DI TEMPO (configurabile)

- **Media della durata di ogni transazione complessa**

Output di una simulazione (2/2)

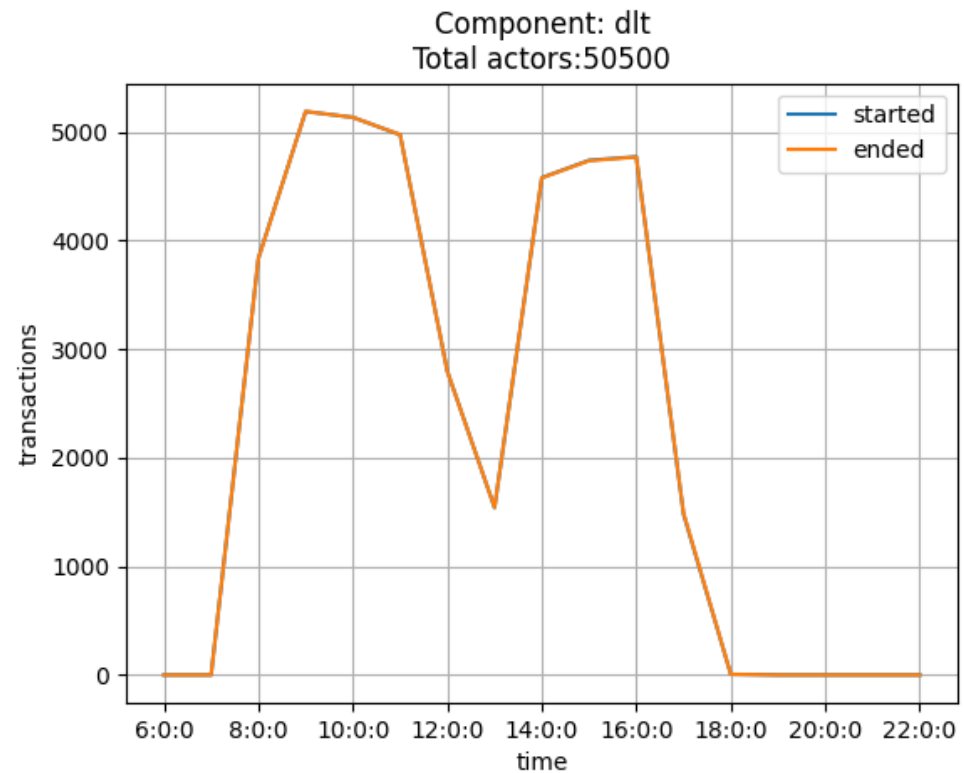
- Gli output precedenti, si riferiscono sempre ad una specifica simulazione e sono salvati in una cartella `/report` e aggregati:



Esempi di output (1/3)

- Per la componente *blockchain* (aka DLT), numero di transazioni iniziate e terminate in ogni intervallo dalle ore 6 alle 22

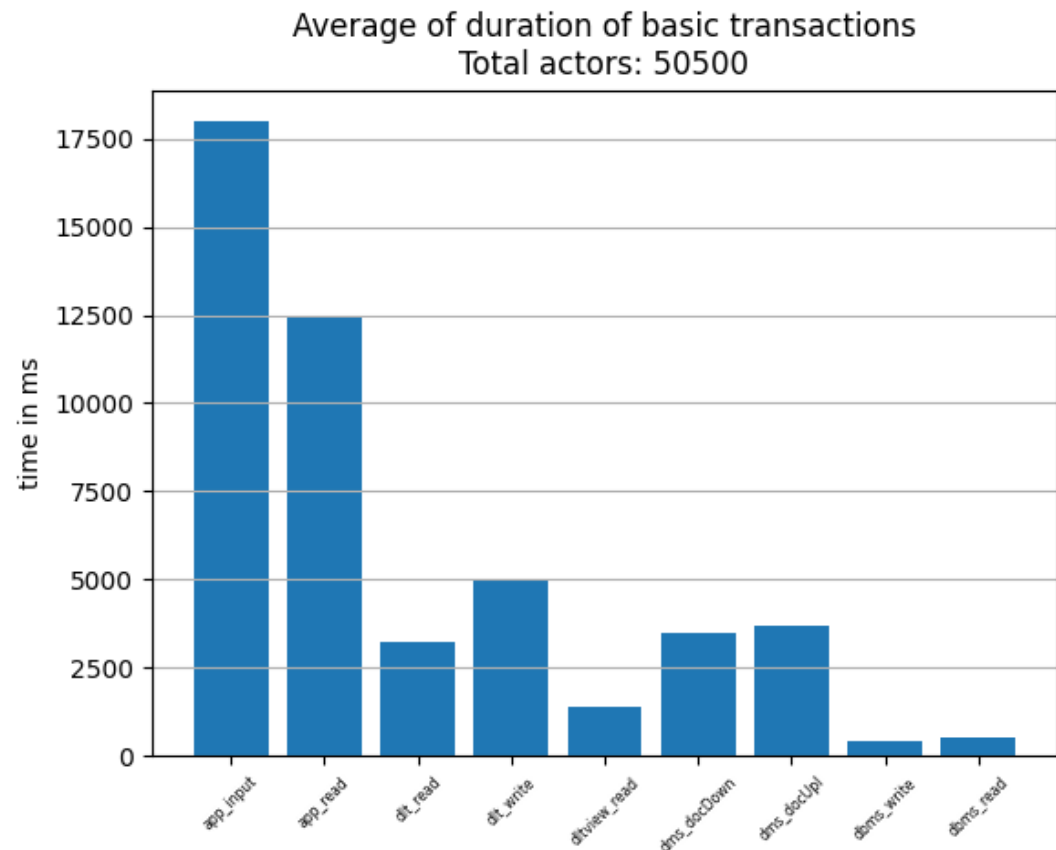
	C1	C2	C3
1	time	started	ended
2	6:0:0	0	0
3	7:0:0	0	0
4	8:0:0	3837	3832
5	9:0:0	5188	5189
6	10:0:0	5135	5134
7	11:0:0	4974	4976
8	12:0:0	2796	2795
9	13:0:0	1541	1542
10	14:0:0	4579	4577
11	15:0:0	4739	4735
12	16:0:0	4772	4769
13	17:0:0	1481	1492
14	18:0:0	6	7
15	19:0:0	0	0
16	20:0:0	0	0
17	21:0:0	0	0
18	22:0:0	0	0



Esempi di output (2/3)

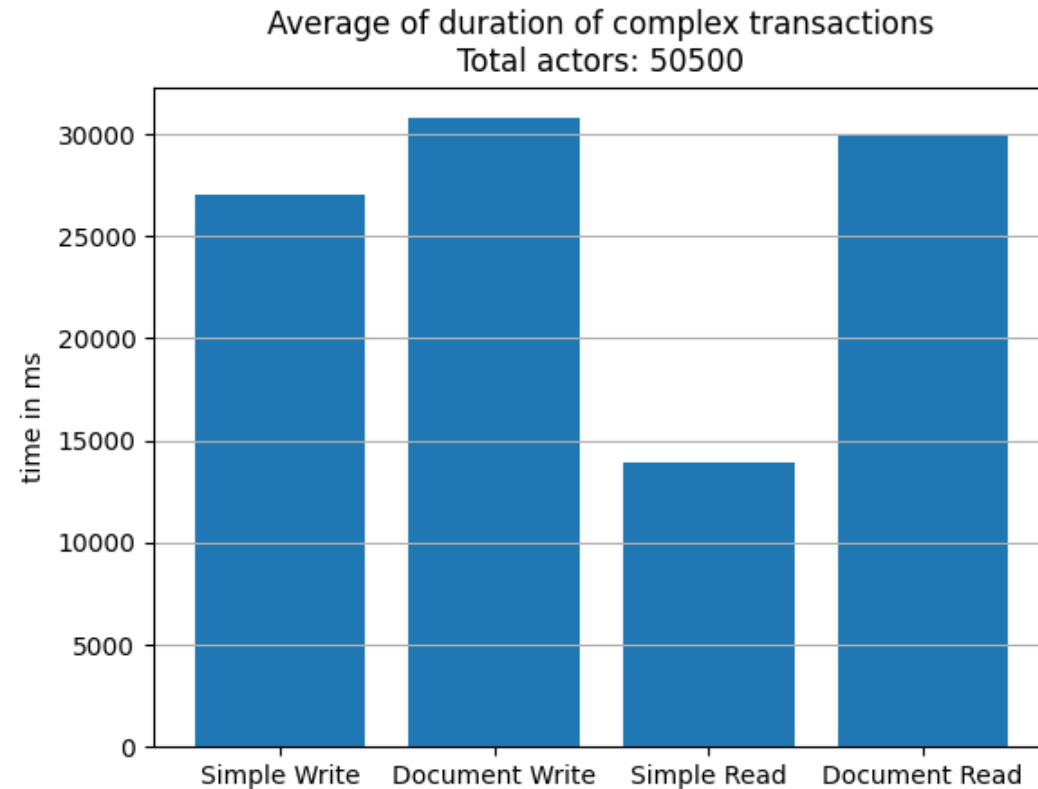
- Per ogni transazione base, durata media, mediana, deviazione st., caso peggiore e migliore **(valori espressi in ms)**

	A	B	C	D	E	F	G
1	base_tx_name,avg,median,stdDev,max,min						
2	app_input	18023.673536377613	17293,5101.273488024014	50047,6423			
3	app_read	12503.30236544847	9015.0,8352.908660878025	79775,1324			
4	dlt_read	3200.777060041408	3023,1018.9755680922194	9352,1130			
5	dlt_write	5002.772967133419	4826,1395.226898259578	14902,1609			
6	dltview_read	1399.3394417771092	1373.0,282.0247414032345	3324,524			
7	dms_docDown	3498.9854712627543	3426.0,675.8119870178226	9079,1541			
8	dms_docUpl	3705.744829886591	3614.0,807.3049104967539	8620,1643			
9	dbms_write	400.6784809074667	370,153.12463305681777	1963,122			
10	dbms_read	500.3616570407612	471,156.65767095137664	1721,166			



Esempi di output (3/3)

- Durata media di ogni transazione complessa (composta da più transazioni base)



Simulazioni eseguite

- Sono state eseguite 10 simulazioni, considerando un numero di attori sempre crescente (i.e. transazioni crescenti) a parità di configurazione di componenti:

- Rapporto 1:100

Nr° Simulazione	Attori	
	#Operatori	#Consumatori
1	200	20.000
2	500	50.000
3	1.000	100.000
4	2.000	200.000
5	5.000	500.000
6	7.500	750.000
7	10.000	1.000.000
8	10.500	1.050.000
9	11.000	1.100.000
10	13.000	1.300.000

Risultati ottenuti (1/3)

Figura 9. Statistiche per ogni transazione base (20.200 attori)

	A	B	C	D	E	F
1	base_tx_name	avg	median	stdDev	max	min
2	app_input	18000.604408465813	17290	5048.934530844104	45145	6359
3	app_read	12489.064248270493	8991	8364.762933824608	74984	1474
4	dlt_read	3201.8313702394976	3022.5	998.864380920667	8017	1230
5	dlt_write	5002.2343704281675	4838	1397.2956924107507	13966	1756
6	dltview_read	1398.9152316296868	1371.0	281.6542481425018	3607	584
7	dms_docDown	3498.396526677232	3426.0	675.2681789253147	9005	1661
8	dms_docUpl	3704.8511339406864	3623	791.3441237740084	7871	1723
9	dbms_write	402.0588120550083	371	155.93674858962322	1629	124
10	dbms_read	498.77840632010145	470	156.5837911967398	1703	185
11						

Figura 10. Statistiche per ogni transazione base (1.060.500 attori)

	A	B	C	D	E	F
1	base_tx_name	avg	median	stdDev	max	min
2	app_input	18009.285587154336	17266.0	5086.105146214073	59331	5381
3	app_read	12497.151095534184	8997	8355.70654141969	105919	1184
4	dlt_read	3197.237254939685	3026	1014.1309397277355	12277	942
5	dlt_write	4997.124007026898	4810.0	1405.5188811741023	16175	1481
6	dltview_read	1399.7050849649445	1372.0	282.27586885676783	3634	523
7	dms_docDown	3498.6849382843666	3427.0	675.2348145697226	13051	1517
8	dms_docUpl	3699.989555007849	3612	802.585585464368	11106	1558
9	dbms_write	399.3262075554562	369.0	151.61899940912102	2151	86
10	dbms_read	499.66896530469154	471.0	157.10858701328723	2122	151

Nr° Simulazione	Attori	
	#Operatori	#Consumatori
1	200	20.000
2	500	50.000
3	1.000	100.000
4	2.000	200.000
5	5.000	500.000
6	7.500	750.000
7	10.000	1.000.000
8	10.500	1.050.000
9	11.000	1.100.000
10	13.000	1.300.000

- Nessun comportamento «anomalo» del simulatore
- Nessun componente ha raggiunto il limite max di txs gestibili

Risultati ottenuti (2/3)

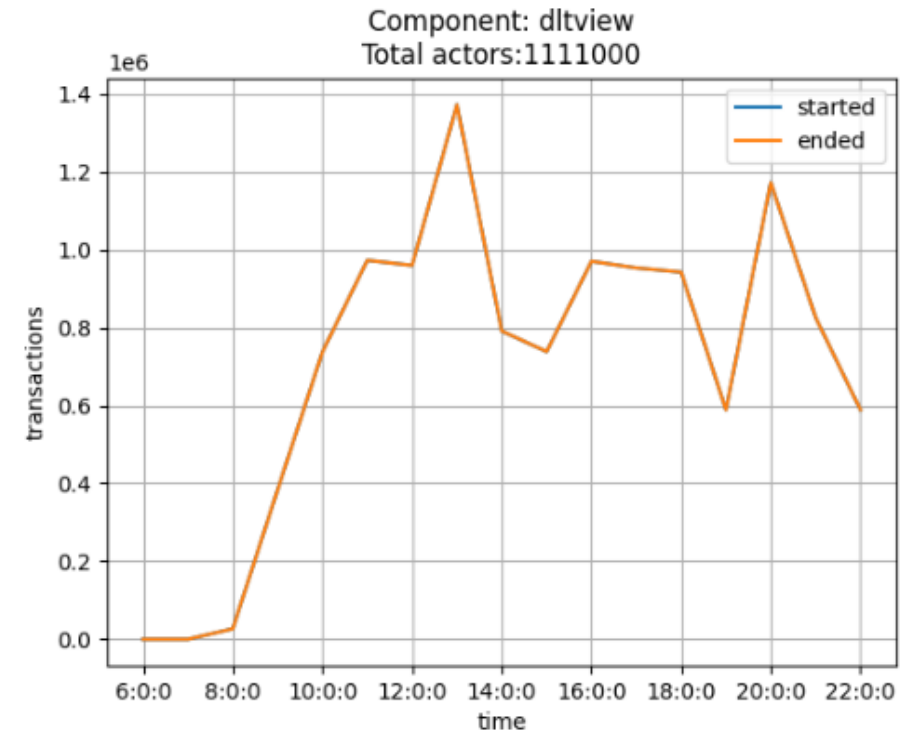
1.111.000 attori

1.313.000 attori

```
main x
Output generation in progress...
Components full
app full at:
  not full
appSys full at:
  not full
dlt full at:
  not full
dltview full at:
  13:3:31
  14:3:31
dbms full at:
  not full
dms full at:
  not full
```

```
Output generation in progress...
Components full
app full at:
  not full
appSys full at:
  not full
dlt full at:
  not full
dltview full at:
  13:0:15
  14:0:15
  20:8:54
dbms full at:
  not full
dms full at:
  not full
```

Nr° Simulazione	Attori	
	#Operatori	#Consumatori
1	200	20.000
2	500	50.000
3	1.000	100.000
4	2.000	200.000
5	5.000	500.000
6	7.500	750.000
7	10.000	1.000.000
8	10.500	1.050.000
9	11.000	1.100.000
10	13.000	1.300.000



- Il componente che raggiunge, per primo, il numero max di transazioni gestibili contemporaneamente è la blockchain

Risultati ottenuti (3/3)

Figura 15. Statistiche simulazione con 1.111.000 attori

	A	B	C	D	E	F
1	base_tx_name	avg	median	stdDev	max	min
2	app_input	18006.62292446792	17261.5	5081.443320683642	62456	5836
3	app_read	12499.104367172666	9001.0	8355.150672748032	102025	1122
4	dlt_read	3198.2055764288584	3026	1015.4669610260436	12044	951
5	dlt_write	4999.564458283368	4814.0	1409.1217220753638	17197	1465
6	dltview_read	1475.310087114268	1413	372.6424594835801	5420.527136541903	507
7	dms_docDown	3498.972855823519	3427.0	675.6626716006174	13407	1443
8	dms_docUpl	3699.105389665435	3611	801.8626249888683	9967	1536
9	dbms_write	399.7447601321278	370.0	152.17443489656827	2130	101
10	dbms_read	499.32716106754253	471.0	157.0172590847383	2054	157
11						

Figura 16. Statistiche simulazione con 1.313.000 attori

	A	B	C	D	E	F
1	base_tx_name	avg	median	stdDev	max	min
2	app_input	18003.82282364509	17264.0	5080.04663873389	65343	5212
3	app_read	12499.13327379569	8999.0	8355.526695854573	121815	1206.4123438447714
4	dlt_read	3203.312176947335	3034.0	1016.5165465921314	12794	865
5	dlt_write	5001.141205590785	4817.0	1408.7588957914584	19710	1171
6	dltview_read	1507.785116123353	1433	401.1085332607355	6084.915221139789	485
7	dms_docDown	3498.9396698150636	3427	675.1016912243016	13056	1401
8	dms_docUpl	3699.7388133113345	3612.0	800.9904943632943	10968	1535
9	dbms_write	399.39767444754904	369.0	151.81825387070774	2179	91
10	dbms_read	499.43828158149876	471.0	156.8479384053209	2138	152
11						

Nr° Simulazione	Attori	
	#Operatori	#Consumatori
1	200	20.000
2	500	50.000
3	1.000	100.000
4	2.000	200.000
5	5.000	500.000
6	7.500	750.000
7	10.000	1.000.000
8	10.500	1.050.000
9	11.000	1.100.000
10	13.000	1.300.000

Commenti finali

- Il simulatore presentato è uno strumento **altamente configurabile**
- Strumento per indagare su eventuali colli di bottiglia e *performance* dei componenti del sistema informatico ABATA
- **Utilità** degli output presentati
- (Decision Support System)

Grazie per l'attenzione

