Indice

Capitolo 1

Risultati Sperimentali

1.1 Il Problema del gas

Spieghiamo il problema del gas nell'esecuzione degli smart contract in ehtereum.

Perchè ci serve la stima??

Se io so quanto è il gas associato ad ogni istruzione, mi basterà sapere come verrà tradotto il mio programma in bytecode. Da lì potrei stabilire il costo fisso. Ma questo non basta. Il gas viene consumato in circostanze differenti.

Il costo totale richiesto per eseguire un programma è determinato in base a 3 fattori:

1. il costo intrinseco di ciascuna istruzione di basso livello; questo valore è fissato.

2. i costi determinati dalla creazione di un contratto o dalla chiamata di un altro programma. Questi sono determinati dalle istruzioni CREATE , CALL and CALLCODE.

3. eventuali costi aggiunti, che vengono addebitati nel caso in cui la memoria richiesta dal programma superi una certa soglia

Mentre alcuni di questi valori possono essere facilmente previsti, altri possono essere determinati soltanto durante l'esecuzione del contratto. Essendo difficili da prevedere, potrebbero far sì che la quantità di gas richiesta in fase di esecuzione ecceda quella messa a disposizione dal committente prima. Dunque conoscere questi costi prima del lancio del programma in rete permetterebbe agli utenti di investire somme di denaro adeguate.

Soltanto delle tecniche di analisi precise ci permettono di stimare questi

consumi, poichè ci permettono di calcolare in anticipo quali "sorprese" riserverà il codice durante la sua esecuzione.

Lo scopo di questo capitolo è quello di riassumere gli esperimenti condotti. Spiego che cosa ho fatto.

Una volta individuati i programmi capaci di produrre bound espliciti ai consumi di gas si è testato il loro comportamento su input diversi. Per dedurre la completezza del software i programmi di input sono stati scelti in modo da testare diversi costrutti base messi a disposizione dal linguaggio Solidity.

1.2 Caratteristiche del Tool GASTAP

1.3 Test Condotti

1.3.1 Caso di Studio: CryptoPhoenix.sol

1.3.2 Operazioni di Assegnamento

1.3.3 Costrutto for

1.3.4 Cicli for Annidati

1.3.5 Costrutto while

while.sol

sqrt.sol

1.3.6 Ricorsione

Ricorsione Diretta

Ricorsione Indiretta

Ricorsione Multipla

1.4 Risultati