　　Ethereum是一個分散式且真實存在世界各地的平台，它是個P2P網絡，也就表示它不需要管理員，也不會在節點之間產生錯誤，此應用程式從不停機，而且每個人都可以去創造一個屬於自己，且不用得到其他人同意的創新。從它被編碼之後，此應用程式就是真實且不變的，而且他們之間都是相互影響的。從這個觀點來看，對於智能合約從發想時就是它們最終的型態，此說法是合理的。

　　創造出這理念的核心是World Computer，就是將世界上的全部節點看成一個本體的世界電腦，這項技術被稱為Ethereum Virtual Machine(EVM)，這包含了資料的操作以及運算，而在這其中的每一筆交易都是一筆session(類似資料傳輸)，這是session互動的一個基礎，就像是一段句子是由文法或是單字組成。

**甚麼是Gas呢？**

Gas指的是在全世界這個World Computer的是用量計量單位，打個比方，電力是用千瓦來計量，在Ethereum中更多的運算和儲存空間使用則是會增加Gas值，而用來支撐這些Gas消耗量的就是全世界的礦工(Miner)，他們的算量支撐了這個World Computer的Gas需求量，對此礦工們會收到fee，也就是俗稱的小費。

每個在EVM中的變更操作都會消耗Gas，舉個例子來說，乘法會消耗5Gas，加法則消耗3Gas，[Gas消耗量表](https://docs.google.com/spreadsheets/d/1m89CVujrQe5LAFJ8-YAUCcNK950dUzMQPMJBxRtGCqs/edit#gid=0)。

**就把Gas當作是能源的一種吧！**

在每一筆交易中，都必須耗費必須的Gas量去支撐EVM所需的處理數據及儲存空間，有點像是RAM和ROM，而每一筆交易剩餘的Gas都會退回給予原本提供此筆交易提供的人，如果不夠的話這筆交易就會回復成初始狀態，依然存在此區塊當中，而若使區塊產生變化，其相關的費用也會給予礦工。

在看了以能源的方面看待Gas的觀點後，我們用小費的方式來看看吧！當每一次在EVM中運算所消耗的Gas量(例如乘法消耗5Gas)，每個交易者可以指定一個特定的Gas量，現在每個Gas的價格是2\*10^ -8 ETH，則對於礦工來講小費的算法就是，(初始Gas – 剩餘Gas) \* Gas單價。

以下是Fuel(能源)和Fee(小費)的觀點不同的表格對照：

|  |  |  |
| --- | --- | --- |
|  | 能源 | 小費 |
| 概述 | 每個在EVM中的處理操作都會消耗預定的Gas量(不會受使用者所更改，Gas消耗量表)，而每筆交易使用者都會有他預期提供的Gas量 | 每筆交易使用者都會有他預期提供的Gas量，作為這次履約保證給予miner小費使用 |
| 開始交易 | 每筆交易的發起人需要提供足夠的初始Gas  初始狀態remainingGas=startGas | 交易發起人筆須提供履約的交易金額(startGas\*gasPrice) |
| 每筆運算 | 在交易過程中，remainingGas會持續因為運算而減少。 | 在此筆交易完成或是失敗錢都會前，都會擱置在合約當中 |
| 交易失敗 | remainingGas == 0，且運算尚未完成，會拋出一個Out Of Gas的Exception | 所有的履約保證的會付給礦工 |
| 交易完成 | 剩下的remainingGas會退還給交易發起人或收益人 | (startGas - remainingGas) \* gasPrice = fee to the miner |

　　假如一筆交易的能源過少，它甚至不會到達任何礦工的手上，假如一筆交易有充足的能源，但是給予的fee(小費)太低，儘管到達礦工那邊，也不會有任何的運算產生，費用也會決定交易的優先順序，高能源可能會導致交易的延遲性，這會在這篇文章的高啟動能源的潛在延遲性(Potential delays with high startGas)提及。

**啟動能源(startGas)**

　　這個章節主要會說到交易開始所需要的能源，在交易的計算和儲存之前，World Computer網絡必須去計算它所需的Gas量，具體來說，在交易之前礦工必須要知道他會得到的小費是多少，這所需的啟動能源必須要能支撐他所會消耗的能源，而startGas也是每一筆交易所必要且重要的。

* startGas是[ETH白皮書](https://github.com/ethereum/wiki/wiki/White-Paper)內的術語
* gasLimit是[ETH黃皮書](https://github.com/ethereum/yellowpaper)內的術語
* gas則是在[Geth](https://github.com/ethereum/go-ethereum/releases)或是[web3.js](https://github.com/ethereum/web3.js/releases)內部的用法

　　startGat讓交易變的比較方便，不然當需要Gas時較要有一筆request回到交易所有人再要求Gas的消耗請求，也能讓礦工知道這筆交易所需的算力，而礦工的機器是否足夠能在有限的時間內跑完這次的交易計算，對於使用ETH應用的使用者來講，如何去提供startGas看起來可能沒那麼容易，但還是有工具可以去估算startGas的。

**Gas不足的例外拋出(Out of Gas exception)**

　　一筆交易內會包含了發起人能夠提供的最大能源消耗，而Gas會在EVM的計算中持續消耗，假如在交易完成前用盡Gas，則會有Out of Gas的例外拋出，原本給予礦工的那些費用和給予的紀錄都會寫在區塊鍊當中，但是其他的變動都會被回復(像是contract的建立、數值的儲存、log)。

　　下面是一次交易的例子，用startGas為90為例：

|  |  |  |
| --- | --- | --- |
|  | Gas消耗 | 剩餘Gas量 |
| 交易開始 | 0 | 90 |
| 儲存31到目的地  (store) | 45 | 45 |
| 加兩個值  (add \* 2) | 10 | 35 |
| 儲存總量  (store) | 45 | Out of Gas |
| 交易結束 | 儲存31動作被退還 | |

　　在交易開始時，發起者必須設定其發起資金(startGas × gasPrice)，並叫由第三方託管，而此筆交易在第三方託管的ETH量是：90\*0.02μETH=1.8μETH，這表示了這筆交易沒有完成，並且回歸到了一開始的狀況，改變的只有礦工得到了費用1.8μETH。

**Gas的退回(Gas refund)**

這裡有兩筆在EVM中的負值計算，也就是退款給擁有者

* 清理此合約中剩餘的gas總量：-24000
* 清理先前交易的儲存數據：-15000

　　當Gas的退款成立時，這筆匯款紀錄會被單獨存在一個空間，Gas的退款只會在交易結束或中止時發生，而這最高的返還金額只是Gas量的一半。

　　重要的一點是，每筆交易中的能源是不會上升的，只會跟著運算所減少，所以說確保能源的需求量是很重要的，以免不必要的Gas耗損。如果要確保有退款，避免Gas不足的例外拋出是很重要的。

　　Gas會給予回收的總量，最多就是總Gas量的一半，例如一筆交易總Gas量是60000，裡面共有兩筆合約，每筆各是消耗24000，則礦工依然會收到30000Gas，這可以讓礦工們得到利益，確保不會在運算上面做白工。

**區塊中的Gas限制(Block gas limit)**

　　對於一個區塊最多能夠容納多少交易，從startGas和交易最大消耗能源下去看待，我們會得到一個最大Gas限制，”Block gas limit”。

　　這個Block gas limit是會變動的，而這是為了避免每個區塊會持續的無限增長，在乙太坊的協定就允許礦工用一些factor在其中調整1/1024的Gas限制。這個Gas限制也可以預防外部攻擊者用交易的無窮迴圈的方式去攻擊一個區塊。

**高初始Gas的潛在延遲問題**

**(Potential delays with high startGas)**

　　為了防止Gas沒必要的消耗，大多人都會把Gas量提高到比每筆交易的需求還要多，那為何不要把Gas提升到幾百萬呢？

　　假如有一筆400萬Gas的交易和100筆4萬Gas的交易，礦工會優先選擇後者的交易，假如400萬Gas的交易只有消耗100萬的Gas，礦工就會損失300萬Gas的潛在利益，因此礦工通常會選擇小型的交易，並且帶有充足量的startGas(除非初始Gas([intrinsic gas](https://ethereum.stackexchange.com/questions/1570/mist-what-does-intrinsic-gas-too-low-mean))很高)，這就是為何有些高startGas的交易會有相當高的延遲性。

**繞道至交易所(Detour to Exchanges)**

在乙太坊中主要有兩種帳號

* 使用者帳號(用私鑰管理)
* 合約(用程式控制)

　　發送ETH到使用者的帳戶需要21000Gas的費用，但若用其他方式，例如發送到合約當中的消耗就更高了，這耗費取決於合約中的code和傳送數據，所以有些使用者想發送ETH到他們的合約錢包Ex:Mist(乙太坊錢包)，但只用了最基本的21000Gas量，交易就會因為Gas不足而退回。

**結論**

　　Gas是用來驅使這個世界電腦在乙太坊上做運算的能量來源，提供足夠的能量和需求的費用是不同的，保險起見用多一點的Gas去驅動交易是安全的，在乙太坊中Gas也是一個重要的存在。更多有關乙太交易方面的東西有[Gas估算](https://github.com/ethereum/wiki/wiki/JavaScript-API#web3ethestimategas)、[初始Gas](https://ethereum.stackexchange.com/questions/1570/mist-what-does-intrinsic-gas-too-low-mean)、[Gas價格估算](https://github.com/ethereum/go-ethereum/wiki/Gas-Price-Oracle)、[交易封包大小的影響](https://ethereum.stackexchange.com/questions/1106/is-there-a-limit-for-transaction-size)。