

イジング型コンピュータの通信を考慮した計算時間

木村 晋二^{1,a)} 宮城 翔太¹ 戸川 望¹

概要: イジング型コンピュータは、複数の量子ビットからなるシステムにおいてシステム全体のエネルギーが最小になるように量子ビットが決まるという量子アニーリングに基づく計算機であり、組合せ最適化問題を高速に解くことができることで注目されている。現在、商用のイジング型コンピュータが複数提供されているが、物理的な量子ビットで実現されるものや、量子アニーリングの模擬により実現されるものなど計算方式に違いがある。また、イジング型コンピュータは、通常リモートのサーバー上に設置され、curlなどを用いてAPI経由でアクセスされ、ネットワーク環境により使い勝手が異なる。このため、種々のイジング型コンピュータの特徴に応じた利用が必要である。とくに、組合せ最適化問題をイジングモデルで表した場合、量子ビットの数の二乗に比例する入力データのサイズが非常に大きくなる場合があり、そのデータをリモートサーバーに送る時間も考慮しなければならない。本稿では、いくつかのイジング型コンピュータにおいて、入力データの転送時間を考慮した計算時間についての検討を行ったのでそれについて報告する。

キーワード: イジングモデルの入力データサイズ, 量子アニーリング, データ転送時間, 総計算時間, データ圧縮

Computation Time of Ising Computers Including Data Transfer

KIMURA SHINJI^{1,a)} MIYAGI SHOTA¹ TOGAWA NOZOMU¹

Abstract: Recently Ising computers have been paid attention as a new computation mechanism beyond Moore's law. Ising computers use a quantum annealing to obtain the the lowest total energy of a set of qubits (spins) for solving combinatorial optimization problems. There have been supplied several commercial Ising computers with different mechanisms and usually such computers are served as remote servers via API. Users need to use Ising computers considering the properties of each machine. Ising computers are known to solve combinatorial optimization problems very quickly based on Ising model whose input size is proportional to square of the number of spins. The manuscript is focusing on the total computation time including data transfer time to and from Ising computers. The data size is discussed on several combinatorial problems, and their transfer time are evaluated on several Ising computers. Data compression by gzip when transferring data is also discussed.

Keywords: Input data size of Ising model, Quantum annealing, Data transfer time, Total computation time, Data compression

1. はじめに

近年古典的な計算方法を超越のものとして量子計算が注目をあびている [1]。中でも D-Wave が実現した初の商用

の量子コンピュータおよびその動作原理である量子アニーリングに基づくイジング型コンピュータが注目を集めている。イジング型コンピュータは、組合せ問題の解法に特化した計算機であり、最適解を求めるのに量子効果を用いている。問題によっては古典的なコンピュータに比べて指数的高速化ができると期待されている [2]。

イジング型コンピュータは、量子ビット (スピンとも呼

¹ 早稲田大学 理工学術院
Faculty of Science and Engineering, Waseda University
^{a)} shinji_kimura@waseda.jp

ぶ)が複数あるシステムにおいて、量子ビット間の相互関係および自己エネルギーに応じて量子ビットが変化して、全体のエネルギーが最小になるという物理現象に基づいて計算を行う。量子ビットを物理的に実現する場合もあれば、量子モンテカルロ法を用いて疑似的に実現する場合もある。現在、種々のイジング型コンピュータが実現されている [3], [4], [5]。

イジング型コンピュータを用いて組合せ最適化問題を解く場合には、問題をエネルギーの最小化に帰着する必要がある。組合せ最適化問題は、一般的には、最小化したい式と制約とからなるが、イジングモデルでは制約も最小化の式の一部として表す必要がある。また、変数のとれる値は二値 (1, -1 または 1, 0) である。1, -1 の場合にイジングモデル、1, 0 の場合に QUBO (Quadratic Unconstrained Binary Optimization) と区別して呼ぶこともあるが、相互に変換可能であることが知られている。

イジング型コンピュータを使うには、まずイジングモデルを作成して最小化したいエネルギーの式を生成し、それをイジング型コンピュータで実行する必要がある。扱える量子ビットの数は数千を超えるが、イジングモデルで量子ビット間の相互作用を表す係数は量子ビット数の二乗に比例し、 10^6 のオーダーとなる。

イジング型コンピュータ自体は非常に高速に最小エネルギー解を求めることが知られているが、その入力サイズや入力を送る時間についてはこれまであまり議論がなかった。そこで本稿ではデータの送信時間を考慮した計算時間について、具体的な問題に対していくつかのイジング型コンピュータで評価および考察を行った。

以下、2章ではイジングモデルについて述べる。3章では具体例として巡回セールスマン問題や二次割当問題、グラフの最小カット問題について述べる。4章ではこれらのイジングモデルの転送時間について評価と考察を示す。5章はあとがきである。

2. イジングモデルについて

イジングモデルは二次元に配置された量子ビット (スピン) をモデル化したものである。例えば 9 個のスピンを $\{s_0, s_1, \dots, s_8\}$ と表し、 s_i と s_j の間の相互係数を J_{ij} 、 s_i の自己エネルギーを h_i と表すと、システム全体のエネルギー E は以下の式で表される。自己エネルギーは横磁場と呼ばれることもある。

$$E = -\sum_{i=0}^8 \sum_{j=0}^8 J_{ij} s_i s_j - \sum_{i=0}^8 h_i s_i$$

この E を最小にするように、実際の物理システムでは、 s_i は 1 か -1 の値を決める。 J_{ij} と h_i は正の値であるのが一般的で、 s_i と s_j が同じ値である場合に E が小さくなり、また s_i が 1 の場合に小さくなる。

また、イジングモデルと等価なものとして QUBO

(Quadratic Unconstrained Binary Optimization) が知られている。QUBO では変数が 1 か 0 の二値を取り、変数間の二次式で最小化したい式を表す。イジングモデルのスピン変数 s_i に対し、 $s_i = 2x_i - 1$ で QUBO で最小化するべきエネルギーの式に変換できる。

組合せ最適化問題により、イジングモデルで表すのが適している場合と QUBO で表すのが適している場合がある。種々の組合せ最適化問題で、どのようにイジングモデルを作るか、すなわち制約をどのようにエネルギーの式に組み込むかの検討が行われている。

3. 具体的な組合せ最適化問題のイジングモデル

イジングモデルあるいは QUBO は、最小化したいエネルギーの式の係数で表すことができる。エネルギーは、変数の二次式で表されるので、係数の数は変数の数の二乗に比例したものとなる。係数の行列は多くの場合はゼロ要素の多い疎行列となるので、非ゼロの要素だけをそのインデックス情報とともに表すのが一般的で、各種イジング型コンピュータでも採用されている。

なお、以下ではいくつかの具体的な組合せ最適化問題に対して、QUBO を用いて組合せ問題最適化問題に対するエネルギーの式を考える。

3.1 グラフの最小カット問題

グラフ $G = (V, E)$ の枝に重み ($w: E \rightarrow \mathbb{Z}$, \mathbb{Z} は実数集合) が割り当てられているとする。 V をほぼ等しい数で 2 分割し、異なるグループ間の枝の重みを最小にする問題を考える。各ノード v_i にスピン変数を割り当て、それが 1 か -1 かで異なるグループを表す。異なるグループ間の重みの総和は、 v_i から v_j への枝の重みを w_{ij} と表すと

$$\sum_i \sum_{j>i} w_{ij} (s_i - s_j)^2$$

と表すことができる。実際は重みの総和の 4 倍となる。また、二つのグループのノード数が同数であるという制約は以下の式を追加することで表すことができる。 A は定数で、枝の重みの総和との関係で決められる。

$$A(\sum_i s_i)^2$$

QUBO の変数 x_i に変換すると、以下ようになる。

$$\sum_i \sum_{j>i} w_{ij} (2x_i - 2x_j)^2 + A(\sum_i (2x_i - 1))^2$$

スピン数 4 の場合の QUBO 例を以下に示す。データの最初の行の数字はスピン数を、二行目の数字は係数 J_{ij} の個数を表す。その後、係数 J_{ij} のリストと係数 h_i のリストが続く。係数 J_{ij} は i, j のインデックス情報とともに示されている。

表 1 最小カット問題のデータサイズ

スピン数	係数の数	ファイルのバイト数
100	4,950	83,170
500	124,750	2,289,342
1,000	499,500	9,261,925
2,000	1,999,000	39,259,580
3,000	4,498,500	89,992,471
4,000	7,998,000	161,461,577
5,000	12,497,500	253,664,108

ペナルティ値は 100 としている。 J_{ij} の値としては、ペナルティ値の 2 倍からノード i とノード j の間の枝の重みを引いたものが、また、 h_i はノードのエッジのコストの和に応じた値となっている。

```
4
6
0 1 188.000000
0 2 152.000000
0 3 44.000000
1 2 16.000000
1 3 180.000000
2 3 112.000000
-192.000000
-192.000000
-140.000000
-168.000000
```

枝の重みをランダムに生成し、グラフのノード数に対応するスピンの数を 5,000 までで入力データを作成した場合は表 1 のようになる。係数行列がスピン数の二乗に比例するため、スピン数の二乗に比例してデータのバイト数が増加する。5,000 スピンの場合には、253 MB にもなっている。

3.2 巡回セールスマン問題

巡回セールスマン問題は、相互間の距離が定義された n 個の都市を総距離最小で 1 度だけ訪問する問題である。巡回セールスマン問題は、QUBO で表すのが一般的である。 n 都市を訪問することを表すために、まず n 時刻を考え、都市 a を時刻 t に訪問した時に 1 となる変数 x_{at} を導入する。変数の数は n^2 である。都市 a と b の間の距離を d_{ab} とすると、都市 a に時刻 t で訪問し、都市 b に時刻 $t+1$ で訪問した時に距離 d_{ab} が加算されるので、すべての a, b, t で $d_{ab}x_{at}x_{b(t+1)}$ を加えたものが距離のコストとなる。一方制約としては、ある時刻 t ではどこか一つの都市にしか存在しないこと $\sum_a x_{at} = 1$ と、ある都市 a にはどこか一つの時刻でしか存在しないこと $\sum_t x_{at} = 1$ がある。実際は、すべての時刻およびすべての都市に対してこの制約を追加する必要がある。これらの制約は、ペナルティ変数を A として $A(\sum_a x_{at} - 1)^2$ をエネルギーに追加し、最

表 2 巡回セールスマン問題のデータサイズ

地点数	スピン数	ファイルのバイト数
10	100	30,479
20	400	276,590
30	900	959,160
40	1,600	2,395,052
50	2,500	4,818,302
60	3,600	6,464,752
70	4,900	13,574,862

小化すれば良い。全体として、以下ようになる。なお、実際の記述では、スピンの番号は $a \times n + t$ とする。

$$\sum_a \sum_b \sum_t d_{ab} x_{at} x_{b(t+1)} + \sum_t A (\sum_a x_{at} - 1)^2 + \sum_a A (\sum_t x_{at} - 1)^2$$

2 地点の X-Y 座標をランダムに生成し、最短の距離の経路があるとしてデータを生成すると以下ようになる。データの最初の行の数字はスピン数、つぎは係数 J_{ij} の個数、係数 J_{ij} のリスト、係数 h_i のリストである。ペナルティ値は 87.361319 である。

```
4
8
0 1 87.361319
0 2 87.361319
0 3 10.920165
1 2 10.920165
1 3 87.361319
2 3 87.361319
-87.361319
-87.361319
-87.361319
-87.361319
```

地点の X-Y 座標をランダムとして、地点数を 10 から 70 まで変化させて入力データを作成した。その時のデータサイズを表 2 示す。最小カット問題の場合よりはサイズが小さく、5000 スピン弱で、13 MB 程度のサイズである。

3.3 二次割当問題

二次割当問題では、 n 個の工場に n 個の場所を割り当てる問題を考える。 n 個の工場間には必要な物資の輸送量が存在し、 n 個の場所間には距離が規定される。割り当てた結果の全体のコストとしては輸送量と距離の積の総和となる。工場間の輸送量は上三角行列の $F = (f_{ij})$ で規定され、場所間の距離は上三角行列の $D = (d_{ij})$ で規定される。

二次割当問題の場合も、工場 a を場所 p に割り当てた場合に 1、それ以外で 0 となる変数 x_{ap} を考える。変数の数は巡回セールスマン問題と同様 n^2 である。輸送量と距

表 3 二次割り当て問題のデータサイズ

工場数	スピンの数	ファイルのバイト数
10	100	51,211
20	400	887,926
30	900	4,532,711
40	1,600	14,884,669
50	2,500	37,126,002
60	3,600	77,817,225
70	4,900	145,115,302

離の積で決まるコストは、以下で表せる。

$$\sum_a \sum_p \sum_{b>a} \sum_{q>p} f_{ab} d_{pq} x_{ap} x_{bq}$$

また、制約として、巡回セールスマン問題同様、一つの工場は一つの場所に、また一つの場所は一つの工場にしか割り当てられないので、以下をコストに加える。

$$A \sum_p (\sum_a x_{ap} - 1)^2 + \sum_a (\sum_p x_{ap} - 1)^2$$

A はペナルティを表す定数である。なお、実際の記述では、スピンの番号は $a \times n + p$ とする。

フローと距離を 1,000 までの整数としてランダムに生成し、2 工場の例を示すと以下ようになる。ペナルティ値は 1,000,000 としている。

```

4
3
0 1 -2000000.000000
0 3 209070.000000
2 3 -2000000.000000
-2000000.000000
-2000000.000000
-2000000.000000
-2000000.000000

```

フローと距離をランダムとして、工場数を 10 から 70 まで変化させて入力データを作成した場合のデータサイズを表 3 に示す。巡回セールスマン問題と比べると 10 倍程度大きく、最小カット問題に近いサイズとなっている。

以上 3 つの具体的な問題に対するサイズを示したが、5,000 スピン程度で最大で 200 MB を超える容量となることがわかる。また、入力データの主たる部分がスピン間の係数であることから、実際のデータ量もスピン数に対して二乗で増加している。

4. データのサイズと転送時間を考慮した計算時間

4.1 イジング型コンピュータへのデータの転送時間

前節で示した通り、入力データのサイズは非ゼロの要素をインデックス付きで示す場合に 5000 スピンで 100 MB を超えて大きくなる場合がある。イジング型コンピュータでの計算時間は通常ミリ秒でカウントされ、多くても数千

表 4 入力データ転送時間 (イジング型コンピュータ I、巡回セールスマン問題)

#cities	#spins	data size (B)	time (sec)
10	100	88,167	0.42
20	400	758,527	1.26
30	900	2,602,687	3.46
40	1,600	6,309,847	8.26
50	2,500	12,486,907	13.53
60	3,600	21,740,167	23.89
70	4,900	34,867,627	37.02

ミリ秒である。一方イジング型コンピュータをリモートで使う場合は入力データを送る必要があるため、データの転送時間を考慮する必要がある。

ネットワークはギガビット毎秒の光回線が家庭まで普及しているが、実際の速度は数百 Mbps (Mega bit per second) であり、100 MB のファイル転送には数秒の時間を要する。このため、イジング型コンピュータの実際の稼働時間と入力データの転送時間は、ほぼ同じと言え、総計算時間は単なる稼働時間の 2 倍必要となる。

実際は、個別のサーバーおよびエンドユーザー間の通信回線に依存し、全体の計算時間を予測するためには、通信回線状況を考慮する必要がある。そこで、ここではいくつかのイジング型コンピュータに対してデータ通信時間の評価を行った。評価では、データ転送に Linux 上の curl (client for URL) を用いた。curl は URL を用いてデータ転送を行うためのツールであり、コマンドラインで実行できる。イジング型コンピュータも通常は curl でリモートから実行できるように設定されている。

データの転送に際し、curl では gzip などデータ圧縮法に対応している。サーバー側が対応している場合は、リモートから圧縮したデータを送ることで、データの転送時間を削減することができる。ここでは、どの程度の効果があるかを実際に検証した。

転送時間に影響を与える要因として、入力フォーマットがある。インデックスを指定して係数行列の値を指定する場合に、あるイジング型コンピュータでは人間の読みやすさを考慮して英語で意味を表す単語を併置するなどしている。また、他の場合でも i, j, J_{ij} を括弧などで囲むなどして、読みやすくすることもある。いずれの場合も、入力データのサイズは大きくなる。以下で示すデータサイズは各イジング型コンピュータの入力書式に従った場合のもので、前節で示したサイズよりは大きくなる。

データ転送時間は、基本的にはデータサイズをネットワークの回線の速度で割ったものとなる。回線の速度を評価するため、イジング型コンピュータのサーバーに対し、WiFi 接続のノート PC (Intel(R) Core(TM) i-78665U, 16 GB 主記憶) からギガビットの光回線を経由してデータを

表 5 入力データ転送時間 (イジング型コンピュータ II、巡回セールスマン問題)

#cities	#spins	data size (B)	time (sec)
10	100	40,745	1.10
20	400	369,805	1.19
30	900	1,278,461	1.29
40	1,600	3,156,525	1.35
50	2,500	6,310,585	1.47
60	3,600	11,046,845	1.59
70	4,900	17,683,305	1.90

表 6 入力データ転送時間 (イジング型コンピュータ II、最小カット問題)

#spins	data size (B)	time (sec)
1,000	12,272,765	1.57
2,000	51,283,418	2.71
3,000	117,029,309	6.81
4,000	209,511,416	19.58
5,000	328,726,938	20.92

送った場合の速度の測定を行った。入力データとしては巡回セールスマン問題とグラフの最小カット問題を用いた。データ転送には Linux 上の curl をデータ圧縮無しで用い、通信の時間は Linux OS の tcsh 上の time コマンドで curl の実行時間として計測した。

まず巡回セールスマン問題に対する結果を表 4 に示す。入力データのサイズに応じた転送時間がかかることと、この接続方式では 1MB の転送に 1 秒程度必要 (8Mbps (Mega bit per second) 弱の速度) であることがわかる。

同じ巡回セールスマン問題に対し、別のイジング型コンピュータに送った場合のデータ転送時間の評価を示す。条件は先の場合と同じである。結果を表 5 に示す。こちらの場合も、データ量に応じた転送時間となっているが、増加が緩やかで、60 Mbps 程度の速度であることがわかる。

また、同じスピン数の場合にデータ量が大きくなる最小カット問題の場合でも通信速度の評価を行った。通信時間のことを考慮し、2 番目のイジング型コンピュータのみで評価を行った。結果を表 6 に示す。

4.2 データ転送時の圧縮について

2 番目にテストを行ったイジング型コンピュータでは、curl での転送で gzip での圧縮転送に対応していたので、その効果も評価した。curl の実行時に以下のオプションを追加して gzip で圧縮したファイルを転送した。これで、転送中はデータが圧縮され、サーバー側で展開されて使用される。

```
-H 'Content-Encoding: gzip' \  
--data-binary @file_name.gz
```

巡回セールスマン問題でデータ量の大きい 60 都市と 70

表 7 gzip 後入力データ転送時間 (イジング型コンピュータ I、巡回セールスマン問題)

#cities	#spins	org size (B)	data size (B)	time (sec)
50	2,500	6,310,585	8,83,804	1.21
60	3,600	11,046,845	4,397,596	1.38
70	4,900	17,683,305	6,327,010	1.57

表 8 gzip 後入力データ転送時間 (イジング型コンピュータ II、最小カット問題)

#spins	gzip data size (B)	time (sec)
1,000	1,823,063	1.30
2,000	7,010,781	1.67
3,000	15,910,247	2.24
4,000	28,406,708	3.40
5,000	44,508,237	4.40

表 9 gzip 並行入力データ転送時間 (イジング型コンピュータ II、最小カット問題)

#spins	gzip time (sec)	transfer gzipped file (sec)	"gzip -c -" (sec)
1,000	0.37	1.30	1.65
2,000	1.40	1.67	3.14
3,000	2.79	2.24	4.90
4,000	4.66	3.40	7.61
5,000	6.76	4.40	11.60

都市の場合に適用した結果を表 7 に示す。表からわかるように、データ量は gzip で約 40% に減り、データ転送時間もそれに依りて削減されていることがわかる。

最小カット問題に対して gzip したデータを送った場合の転送時間を表 8 に示す。この表からは、通信回線は 80 Mbps 程度の速度と評価でき、圧縮された分だけ転送の時間が減っている。

なお、圧縮データの転送の場合は、Linux のパイプ機能を用いて以下のように gzip で圧縮しながらデータを送ることもできる。

```
gzip -c file_name | curl -X POST \  
--data-binary @- -H "Content-Encoding: gzip"
```

表 9 にこのように gzip を行いながら並行して圧縮データ転送を行う場合の評価結果を示す。表では、gzip をファイルに行う場合の実行時間と gzip した後のファイルを転送する時間と、今回の gzip と転送を同時に行う場合の時間を示している。

現在使用している UNIX 環境では、表からわかるように、プレーンなファイルに gzip をしながらファイル転送する場合は、gzip でファイルを作成する時間と転送時間の和にほぼ等しい時間が必要であることがわかった。gzip の時間を含めて考えると、全体では、転送時間は約半分となっている。これがプロセスの並列化などで削減できるかどうかは今後の課題である。

5. おわりに

本稿では、入力データの転送時間を考慮したイジング型コンピュータの計算時間について評価と考察を行った。イジング型コンピュータはサーバーとして提供されることが一般的で、エンドユーザーはインターネット経由で使う必要がある。curl などを用いて URL 経由でデータ転送およびイジング型コンピュータの起動ができるので、リモートからの使用は非常に便利になっている。またイジング型コンピュータの実働時間は非常に短くて良いことがわかっている。しかし、イジングモデルを表す入力データはスピンの数とともに、その二乗に比例して大きくなる。通信回線の速度は一定であるため、リモートサーバーへの転送には、データ量に応じた時間を必要とし、イジング型コンピュータの計算時間を評価する場合には、イジング型コンピュータの実働時間だけでなく、データ転送の時間を考慮する必要がある。

今回 2 種類のイジング型コンピュータでデータの転送時間について評価を行ったが、ユーザー側からは同じインターネットの環境で、同じ入力データであったとしても、通信時間には非常に大きな差がでることが分かった。

また、サーバー側で curl 転送時に gzip による通信時のデータ圧縮に対応している場合の効果を評価した。結果として、gzip で圧縮したデータを送ることで、削減量に比例した通信時間の短縮効果が得られることを確認した。問題のサイズが大きいほど gzip による圧縮率は高くなり、巡回セールスマン問題の場合は最大で 35.7% に、それよりもサイズの大きい最小カット問題や二次割当問題では 21.2% 程度に圧縮できることがわかった。

5,000 スピンでも、数百 MB の入力サイズになることもあり、一般的な gzip でもデータ圧縮は転送時間の短縮に有効であるといえる。また、通信回線の状況や、データのサイズによっては、単なる実行時間の数倍から数十倍の時間を入力データの転送に用いるため、イジング型コンピュータ毎にデータの転送時間を事前に計測することは総実行時間を見積もる上で重要であると言える。

謝辞 日ごろからご討論いただき、早稲田大学・電子物理システム学科・情報システム研究室の柳澤政生教授、史又華教授、吉増俊彦教授をはじめとする研究室のメンバーに心から感謝します。とくに、情報システム研究室の植田圭氏にはイジング型コンピュータの入力データの作成およびフォーマット変換で一部ご協力いただきました。なお、本研究(の一部)は、内閣府総合科学技術・イノベーション会議の戦略的イノベーション創造プログラム(SIP)「光・量子を活用した Society 5.0 実現化技術」(管理人:量子科学技術研究開発機構)によって実施されました。

参考文献

- [1] Shu Tanaka, Yoshiki Matsuda, Nozomu Togawa, "Theory of Ising Machines and a Common Software Platform for Ising Machines," Proc. of ASP-DAC 2020, pp. 659-666, Jan. 2020.
- [2] Elizabeth Crosson, Aram W. Harrow, "Simulated Quantum Annealing Can Be Exponentially Faster than Classical Simulated Annealing," Proc. of IEEE 57th Annual Symposium on Foundations of Computer Science, pp.1-10, Oct. 2016.
- [3] Satoshi Matsubara, Motomu Takatsu, Toshiyuki Miyazawa, Takayuki Shibasaki, Yasuhiro Watanabe, Kazuya Takemoto, Hirofumi Tamura, "Digital Annealer for High-Speed Solving of Combinatorial Optimization Problems and Its Applications," Proc. of ASP-DAC 2020, pp. 667-672, Jan. 2020.
- [4] Chihiro Yoshimura, Masato Hayashi, Takashi Takemoto, Masanao Yamaoka, "CMOS Annealing Machine: A Domain-Specific Architecture for Combinatorial Optimization Problem," Proc. of ASP-DAC 2020, pp. 673-678, Jan. 2020.
- [5] Takuya Okuyama, Masato Hayashi, and Masanao Yamaoka, "An Ising Computer based on Simulated Quantum Annealing by Path Integral Monte Carlo Method," Proc. of IEEE International Conference on Rebooting Computing, pp.1-6, Nov. 2017.