問題適合型高精度計算ライブラリの開発

プロジェクト代表：
長谷川秀彦@筑波大学図書館情報メディア研究科

発表者：
後 保範@早稲田大学

プロジェクトの目的

- 計算誤り検証付きライブラリの開発
- RSA 暗号(1024ビット)の強度推定: 発表
- 自動チューニング機構付き線形計算ライブラリの開発
- 精度保証付き線形計算ライブラリの開発
- ４倍精度線形計算ライブラリの開発

RSA 暗号の強度推定の計画

- 初年度(発表年度)

  RSA暗号ふるい処理の高速化と評価

- 2年度 (発表年度)

  標数2の線形方程式解法の作成と高速化

- 3年度 (予定)

  現1024ビットRSA暗号の強度推定

RSA 暗号の強度推定の背景

(1) 現在の暗号 (RSA暗号)や認証システムは多数桁数(1024ビット,10進309桁)の因数分解の困難さを利用している。

(2) 現在の多数桁数の因数分解の世界記録は、RSA-768(10進232桁)。2010年1月、NTT他5カ国共同、GNFSを使用。

(3) 「暗号の2010年問題」：RSA暗号を含む現在の暗号システムの変更が必要

RSA-768(232桁)の計算時間

項目	台数･年	比(%)	対象年
ふるい処理	1500	90	昨年度
標数2の線形計算	155	9	今年度
利用関数の探査	20	1	次年度
代数平方根の計算	1	0	次年度
その他	1	0	次年度

注) AMD64 (2.2Ghz, 1コア換算)
行列サイズ：192,796,550 * 192,795,550

標数2の線形方程式解法

多数のふるいデータから$a^2=b^2 \pmod N$が成り立つようにデータを選出するため。解が1の列を選ぶ

(1) 行列のサイズ
- 行数：基底要素(素数)の数(数千万～数億)
- 列数：ふるいで得たデータの数(行数+数百)
- 非ゼロ要素数：50～150/行

(2) 解法の種類
- 直接解法： メモリを反復解法の数十倍使用

反復解法における標数2の特徴

- **利点**
  64**ビット整数で、**64**個の標数が一挙に計算可能**
  ==> **演算量の削減**

- **問題点**
  確率1/2**で内積の値 (スカラー)がゼロになる**
  ==>**スカラーの除算が必要で反復計算できない**
  ==> **「**64**ブロック化」で対処 (確率**$1/2^{4096}$)
  (64**次元の部分行列が正則なら反復可能**)

ブロックランチョス法(64ブロック化)

- 64本の方程式 (同一係数) を同時に解く
- 反復回数
  次元数の1/64以下で64個の解が得られる
- 主要計算(99.9%)
  (1) 疎行列乗算 ($q=A^TAp$)
  (2) 64×64個の内積計算 ($\alpha=p^Tq$)
  (3) 64×64個の積和計算 ($q = \alpha p$)

行列の形と規模(標数2)

- A: m×n次元の疎行列
  m, nは数千万～数億(n=m+数百)
  列当り平均非ゼロ要素数: 50～150
- p, q : n×64次元の行列 (ブロック化)
  64ビット整数のn次元ベクトルに記憶
- α: 64×64次元の行列 (ブロック化)
  64ビット整数の64次元ベクトルに記憶

疎行列乗算 ($q=A^TAp$)

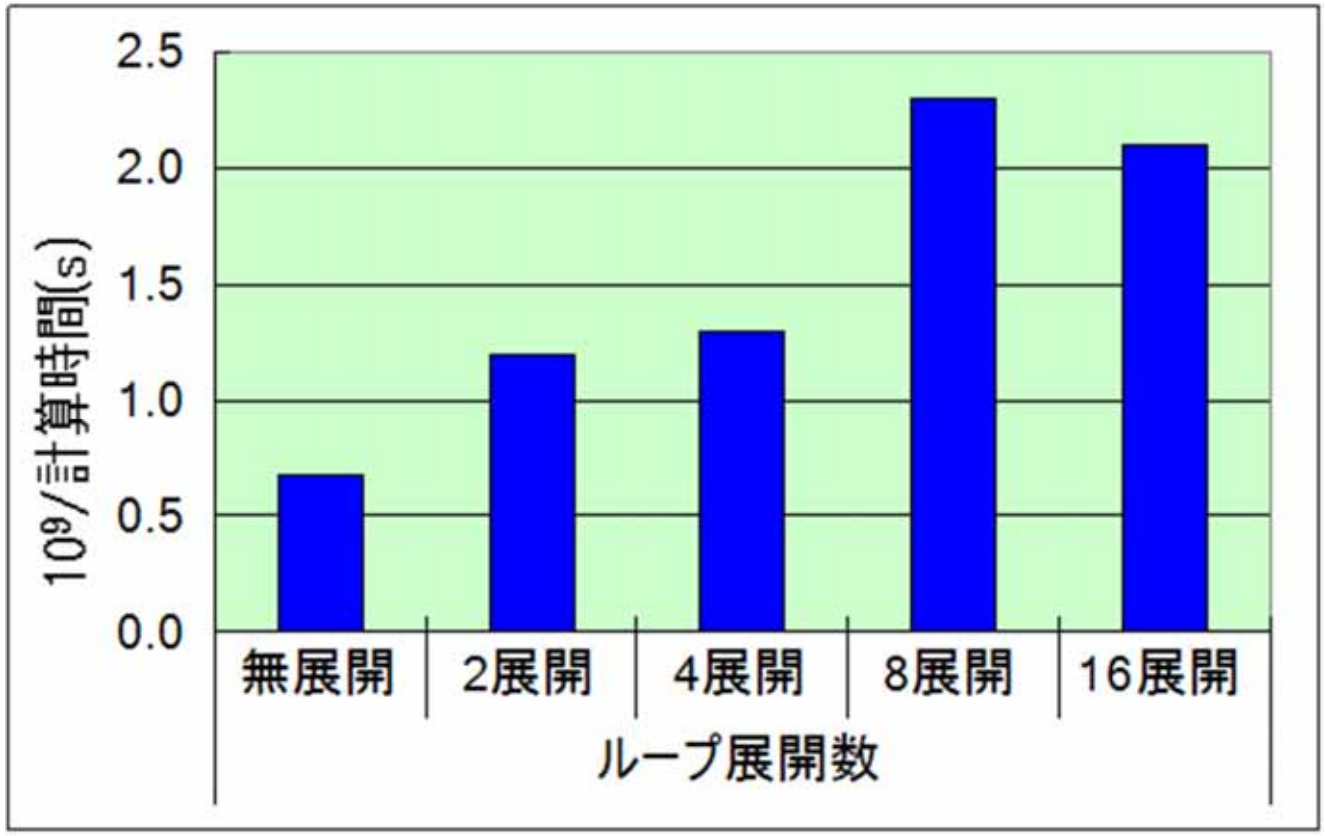
- ベクトル長を長く保つためA^TとAは別に保持
- $w=Ap$, $q=A^Tw$に分けて計算
- MPIの並列処理での処理

  (1) A,A^Tの非ゼロの位置だけ記憶

  (2) 乗算するベクトルが全ての次元数必要で、それ以外のベクトルと行列は並列数で分割可

  (3) A,A^Tの非ゼロ数を均等に分け、ベクトル長を長くするため、最初にソート

$q=A^{T}w$の計算性能(ES2の1CPU)



標数2の内積計算 ($\alpha=p^Tq$)

- 方式A
  64ビット整数の「シフト、乗算及び排他和」で
  64個まとめて計算
  ==> 連続アドレスのベクトル化可能
- 方式B
  分割統治法により演算量を約1/8に
  ==> 重なるアドレスへのストアが必要
  ==> ベクトル化は不可能

標数2の内積計算：方式A

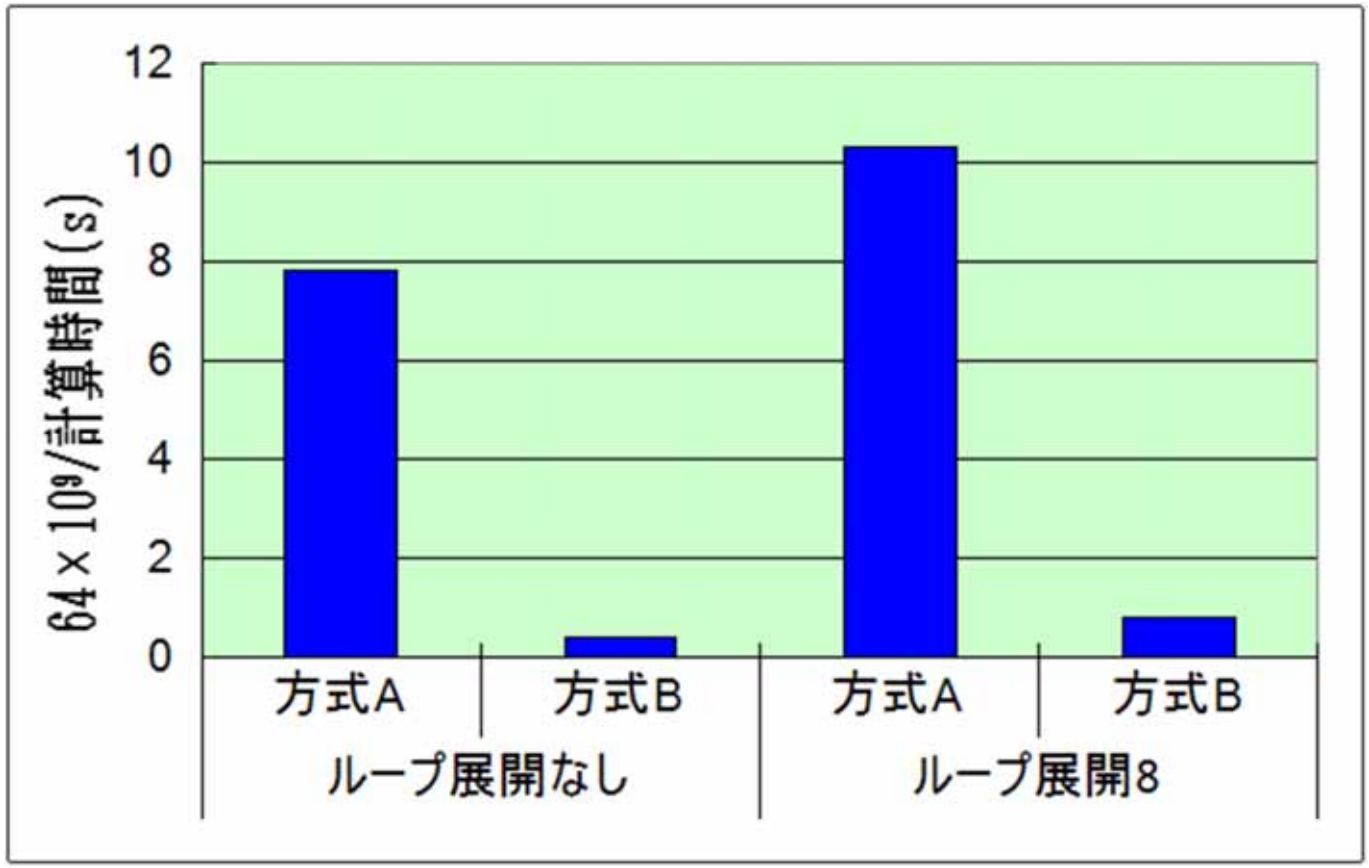
```
for (k=0; k<64; k++)
  { S = 0;
    for (i=0; i<Npr; i++)
      { Wk = (Q[i] >> k) & 1;
        S ^= Wk*P[i]; }
    Alpha[k] = S;
  }
```

NprはNをMPI並列数で分割

標数2取り出し

標数2を64同時処理

標数2の内積計算性能 (ES2の1CPU)



標数2の積和計算 (q=αp)

- 方式A
  64ビット整数の「シフト、乗算及び排他和」で64個まとめて計算可能
  ==> 連続アドレスのベクトル化可能
- 方式B
  分割統治法により演算量を約1/8に
  ==> リストベクトルでベクトル化可能

標数2の積和計算;方式A

```
for (k=0; k<64; k++)
  {  for (i=0; i<Npr; i++)
     {  V= (P[i] >> k) & 1;
        Q[i] ^= V*Alpha[k]; }
  }
```

NprはNをMPI並列数で分割

標数2取り出し

標数2を64同時処理

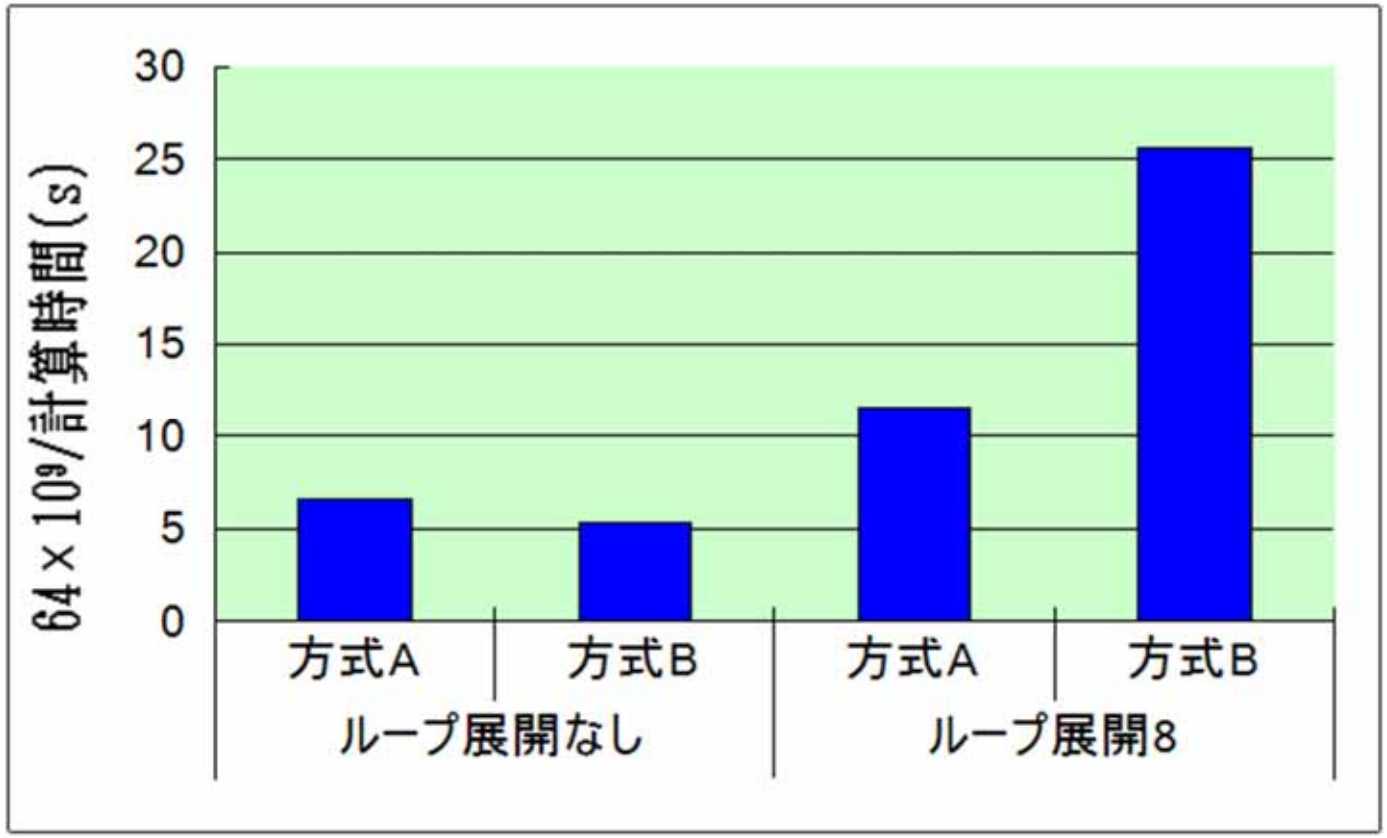
標数2の積和計算：方式B

```
for (j=0; j<8; j++)  {  j8 = j*8;
   for (i=0; i<256; i++)
    {  k = 0;  id = i;  S = 0;
       while (id)  { S ^= (id & 1)*Alpha[k+j8];
                     id  = id >> 1;    k++;    }
       W[i] = S;   }
   for (i=0; i<Npr; i++)
     { Q[i] ^= W[(P[i] >> j8) & 255];  }     }
```

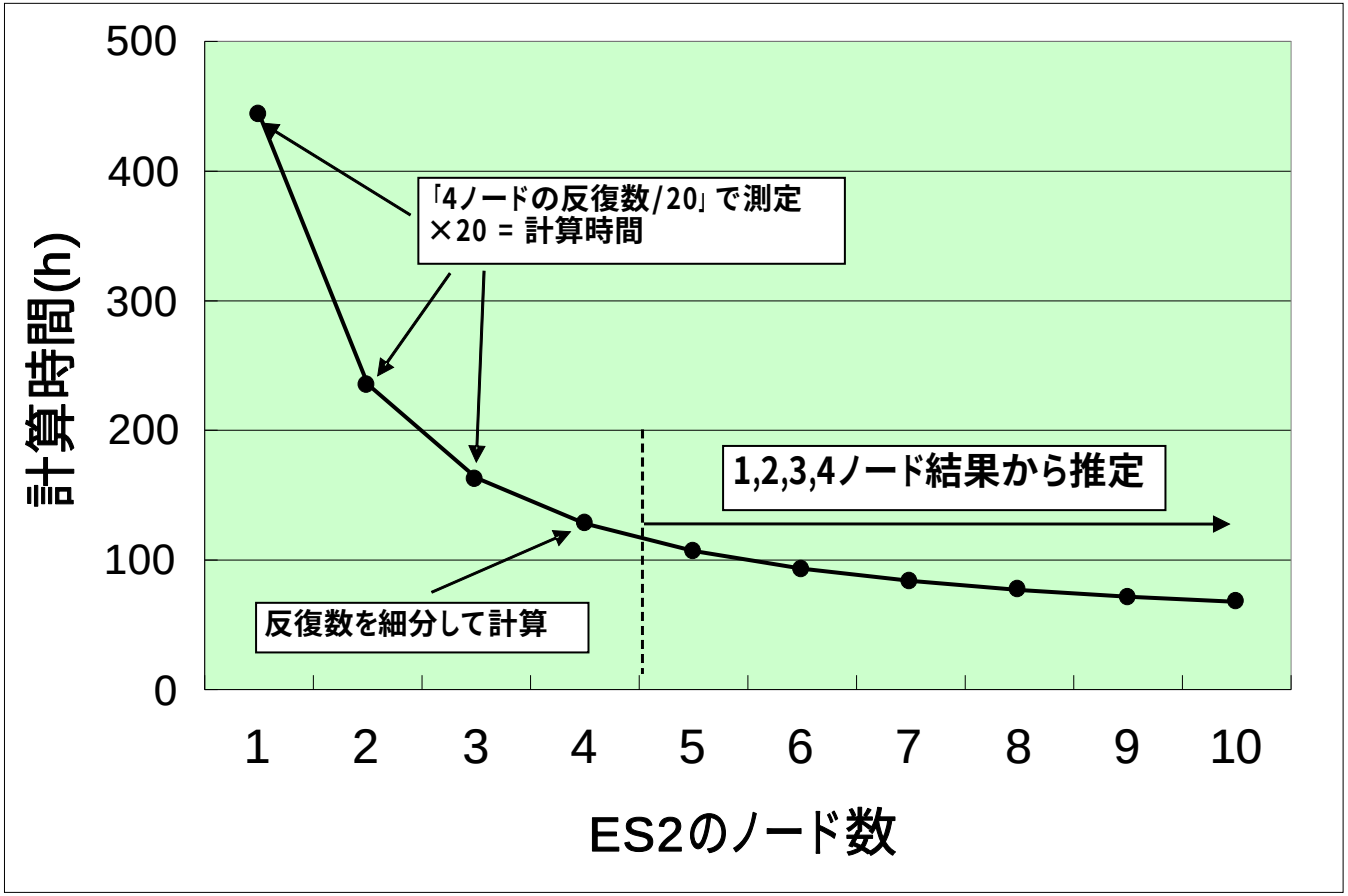
8ビット単位に2^8個のテーブル作成

ビット重ね技法

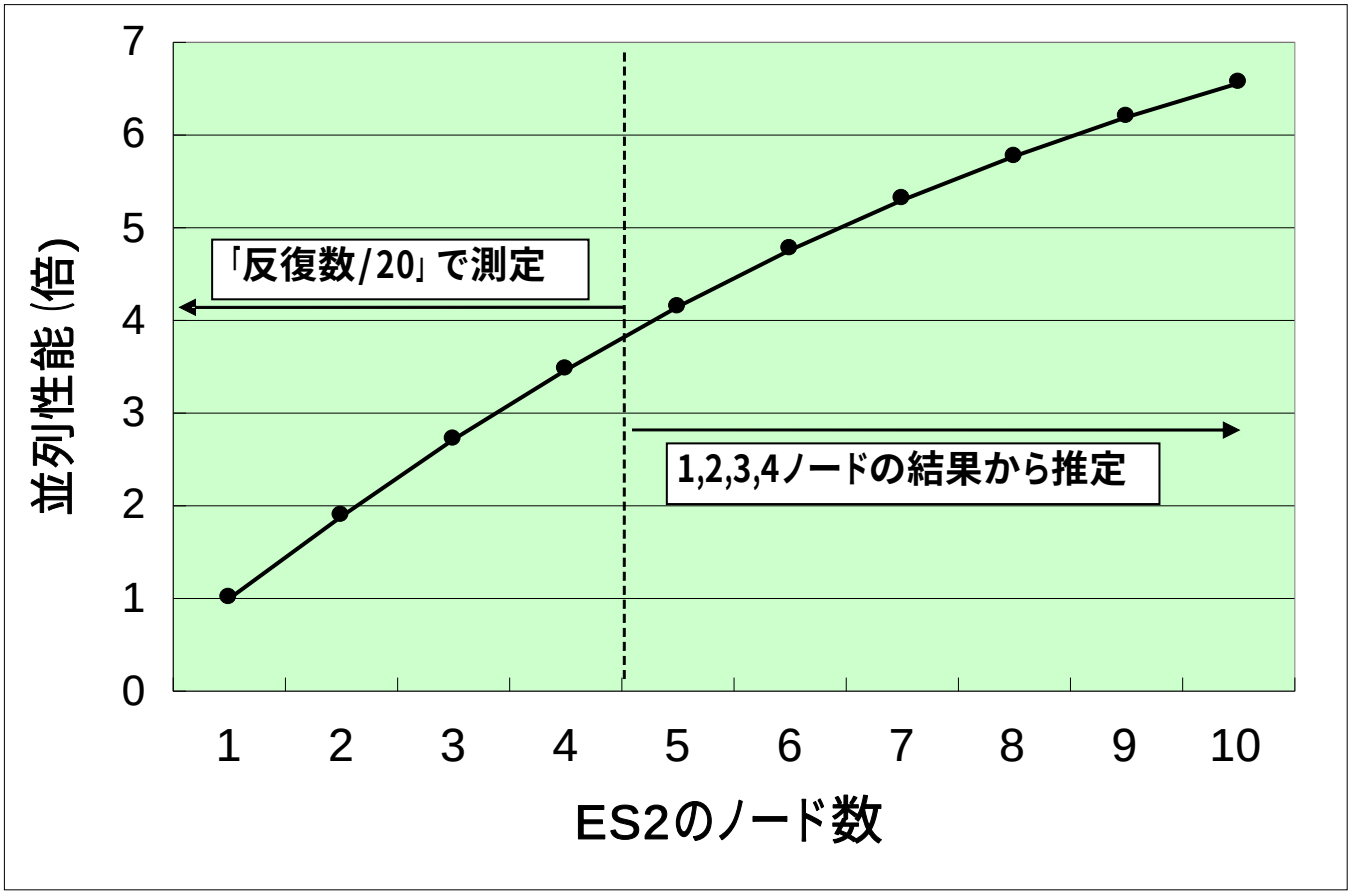
標数2の積和計算性能(ES2の1CPU)



8000万次元の計算時間



8000万次元の並列性能



まとめ

- **行列乗算**(A^TAp)**及び積和計算**(αp)**はループ展開でほぼ十分。内積計算** (p^Tq)**は演算量削減(方式B)が適用できず対スカラー性能が不十分**
- **リストベクトルを多用するので、手でループ展開することが有効**(αp**で**4.8**倍)。ベクトル化率**99.9%
- **疎行列**A**はベクトル長を長くするため、**A**と**A^T**が共に必要。**
- 8000**万次元が**4**ノードで**130**時間 (細切れ利用可)**
- **ハイブリッド並列がメモリ使用量で有利**
- **計算時間は次元数のほぼ**2**乗に比例**

推定性能

- 8000万次元で並列性能は、4ノードで1ノードの3.5倍、8ノードで5.8倍と推定
- 2億次元(RSA-768規模)の標数2の線形方程式解法は、4ノードで約800時間と推定

  ==> 4ノードでAMD(2.2Ghz)の1700台相当

  8億次元(RSA-896規模と推定) はES2の16ノードで、5000時間(8ヶ月)程度と見積