Watts-Strogatz モデルに基づく 大規模ランダムグラフの分散並列生成

安藤 祐斗

富山県立大学　情報基盤工学講座
t815008@st.pu-toyama.ac.jp

May 21, 2021

本研究の背景

背景

近年, ソーシャルネットワークに代表される巨大なデータは「物」を頂点, 「関係」を辺とした大規模なグラフであると捉えられる. これに対して, 解析処理を行うプログラムの開発需要が高まってきている. そのようなプログラムの性能評価には, ランダム生成されたグラフを用いた評価が行われる場合がある. しかし, 逐次的なプログラムによる, 大規模な性能評価用グラフの生成には、非常に時間がかかり, メモリ不足の懸念もあるため, 望ましくない. そこで, 大規模グラフ生成の分散並列化が望まれている.

研究の目的

目的

本研究では, ランダムグラフ生成モデルである Watts-Strogatz モデルを,Hadoop MapReduce を用いて分散並列化を提案する. また, 本手法では, このモデルに生じる再接続辺重複問題を回避するための制約を追加し, それにより従来の Watts-Strogatz モデルに期待される性質を破壊しないことの確認をする.

Watts-Strogatz モデル (1)

グラフ生成の流れ

1．規則的なリング状格子を形成する
2．辺の確率的な再接続を行い, 規則性を乱す.

パラメータ

N:グラフの頂点数 K:平均次数の半分 (各頂点は平均で $2K$ 本の辺をもつ)
P:辺の再接続確率 (グラフの乱れ具合)

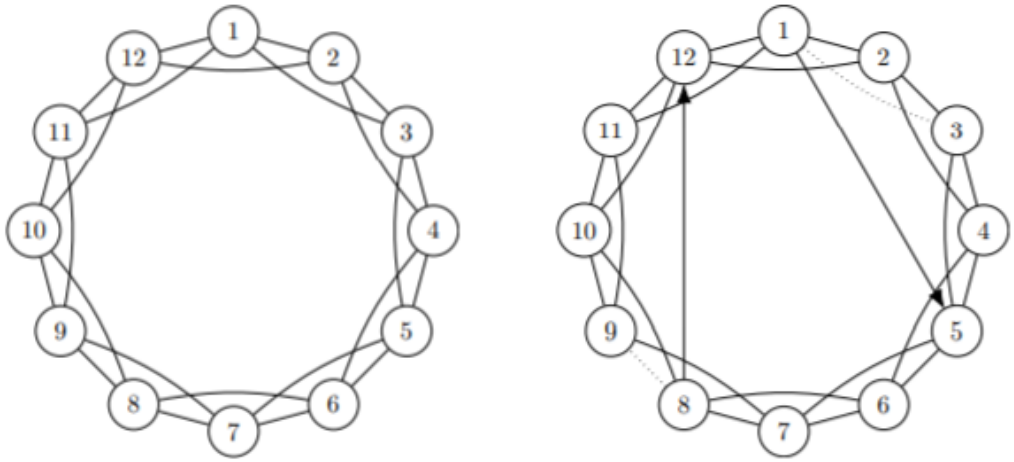

図 1: リング状格子と再接続処理の例（N=12,K=2）

Watts-Strogatz モデル (2)

全頂点を $V = \{1, ..., N\}$,
v の担当する再接続処理対象の辺を
$E(v) = \{(v, v+1), (v, v+2), ..., (v, v+K)\}$,
$rand()$ を $[0, 1]$ の一様なランダムな値を返す関数とする.

foreach $v \in V$
  foreach $e \in E(v)$
    if rand() $> P$ **then**
      e の終点を V' から一様ランダムに選択
      where
        $V' = V \setminus \{v-K, \ldots, v, \ldots, v+K\}$
        $\setminus$ { 再接続で v と繋がった頂点 }

図 2: 疑似コード

Watts-Strogatz モデル (3)

並列化の問題点

・再接続辺重複問題　重複した 2 本の辺は 1 本に同一視され, グラフ全体の辺の数が期待された数より小さくなる

提案

各頂点 v について再接続先を次のようにする.
v から円の中心を見て. 左側の半円の頂点のうち v と頂点番号の偶奇が一致する頂点と, 同じく右半円のうち偶奇の異なる頂点.
N が偶数である場合, 真反対の頂点とは接続できないものとする.

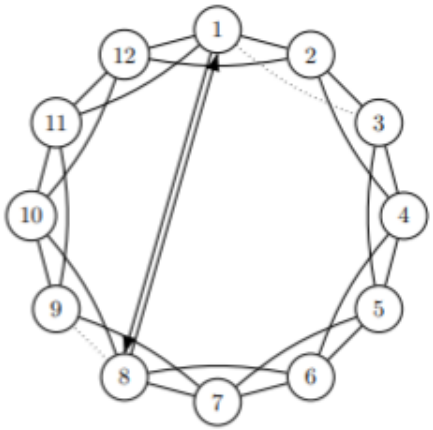

図 3: 再接続処理の並列化時に発生し得る辺の重複

Watts-Strogatz モデル (4)

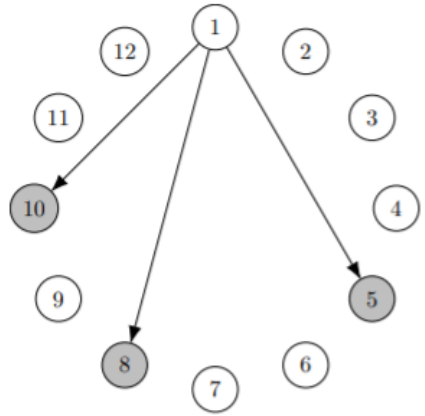

図 4: 頂点 1 から接続可能な頂点

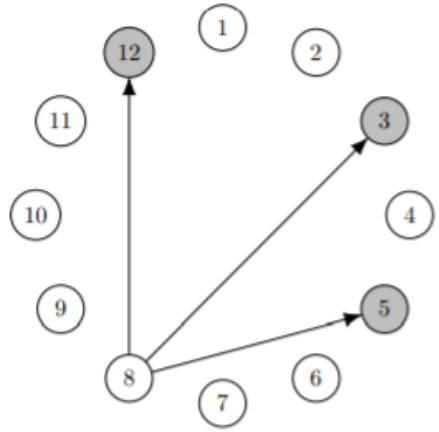

図 5: 頂点 8 から接続可能な頂点

諸性質の保存 (1)

提案手法が,Watts-Strogatz モデルで生成されるグラフに期待される性質を保っていることを示す.
生成されたグラフにおいて, ある頂点が d 本の辺を持つ確率 $P(d)$ を計算する.

頂点の持つ d 本の辺

1．自身が再接続する K 本の辺
2．他の頂点の再接続処理により変わる部分で,$n_i = d - K$ 本存在し, リング上格子の段階で接続されていた辺 $n_{i,1}$ 本と別の頂点の再接続処理によってつながれた $n_{i,2} = n_i - n_{i,1}$ 本とに分けられる

$P_1(n_{i,1})$,$P_2(n_{i,2}) = P_2(d - K - n_{i,1})$ と書くと,$d \geq K$ に対して, 次式で与えられる.

$$P_p(d) = \sum_{n_{i,1}=0}^{min(d-K,K)} P_1(n_{i,1}) P_2(d - K - n_{i,1})$$

諸性質の保存（2）

$P_1(n_{i,1})$ と $P_2(n_{i,2}) = P_2(d - K - n_{i,1})$ は次式によって与えられ,

$$P_1(n_{i,1}) = \binom{K}{n_{i,1}} (1-P)^{n_{i,1}} P^{K-n_{i,1}}$$

$$P_2(n_{i,2}) = \binom{N}{n_{i,2}} \left(\frac{KP}{N}\right)^{n_{i,2}} \left(1 - \frac{KP}{N}\right)^{N-n_{i,2}}$$

N を大きくした際の極限を取れば $\lambda = N \times (KP/N) = KP$ のポアソン分布になり, 次式を得る.

$$P_2(n_{i,2}) = \frac{(KP)^{n_{i,2}}}{n_{i,2}!} e^{-KP}$$

諸性質の保存 (3)

よって, 元の Watts-Strotgz モデルでの生成における $P_p(d)$ は次式によって書き換えられる.

$$P_p(d) = \sum_{n=0}^{min(d-K,K)} \binom{K}{n} (1-P)^n P^{K-n} \times \frac{(KP)^{d-K-n}}{(d-K-n)!} e^{-KP}$$

次に提案手法で改変した生成法に関する $P(d)$ を計算する. 提案手法における $P_2(n_{i,2})$ は,

$$P_2'(n_{i,2}) = \binom{\frac{N}{2}}{n_{i,2}} \left(\frac{2KP}{N}\right)^{n_{i,2}} \left(1 - \frac{2KP}{N}\right)^{\frac{N}{2} - n_{i,2}}$$

となり,N を大きくした際の極限を取ってポアソン分布に近似することを考える
と,$\lambda = N/2 \times 2KP/N = KP$ となり,
$P_2'(n_{i,2}) = P_2(n_{i,2})$ となる. 以上より, 提案手法によるグラフ作成は元のモデルでの接続性を保存していることがわかる.

クラスタ係数の保存 (1)

クラスタ係数の保存を確かめる. 元の Watts-Strogatz モデルでのクラスタ係数を求め, その後提案手法による生成での結果について述べる. クラスタ係数 C は, 各頂点のクラスタ係数 c_v の平均値で定義される.
$C = E(c_v)$
頂点 v の隣接頂点数を d_v, その隣接頂点間に存在する辺の本数を N_v とするとクラスタ係数 c_v は次式で定義される.

$$c_v = \frac{N_v}{d_v(d_v-1)/2}$$

$P = 0$ の時を考える, すなわちそれは再接続が行われずリング状格子が生成結果となる場合である N_v は次式で示される.

$$N_v = \bar{N} = K(K-1) + \frac{K(K-1)}{2} = \frac{3K(K-1)}{2}$$

クラスタ係数の保存 (2)

さらに,$d_v = 2K$ なのでクラスタ係数 $C(0)$ は次式によって表される.

$$
\begin{aligned}
C(0) &= \mathrm{E}\left(\frac{N_v}{c_v(c_v-1)/2}\right) \\
&= \mathrm{E}\left(\frac{3K(K-1)/2}{2K(2K-1)/2}\right) \\
&= \frac{3(K-1)}{2(2K-1)}
\end{aligned}
$$

次に,$P > 0$ の場合を考える. その場合の N_v の期待値を計算する. リング状格子状態から再接続処理を行ったとき, 既存の三角形が保たれる確率すなわち三本の辺すべてが再接続されない確率は $\overline{N}(1-P)^3$,3 点がお互いにお互いを接続先として再接続する確率は $O(N^2) \times O((1/N)\ 3) = O(1/N)$ なので N_v の期待値は次式となる

$$
\mathrm{E}(N_v) = \bar{N}(1-P)^3 + \mathrm{O}\left(\frac{1}{N}\right)
$$

クラスタ係数の保存 (3)

$P > 0$ のときのクラスタ関数 $\overline{C}(P)$ は次式となる.

$$\begin{aligned}\tilde{C}(P) &= \frac{\mathrm{E}(N_v)}{\mathrm{E}(c_v(c_v-1))/2} \\ &= \frac{\bar{N}(1-P)^3}{\mathrm{E}(c_v(c_v-1))/2}\end{aligned}$$

さらに, 分母について平均と演算の順序の入れ替えと,$E(c_i) = 2K$ より, 最終的に次式を得る.

$$\begin{aligned}\tilde{C}(P) &\sim \frac{\bar{N}(1-P)^3}{\mathrm{E}(c_v)\mathrm{E}(c_v-1)/2} \\ &= \frac{\bar{N}(1-P)^3}{2K(2K-1)/2} \\ &= C(0)(1-P)^3\end{aligned}$$

導入した制約により変化しうる部分は再接続により新たな三角形が構成される部分だが, 新たな三角形を構成しうる 2 頂点の幅が半分になろうとも, 再接続により互いが接続され得る確率が倍になろうとも, 新たな三角形が構成される可能性は $O(1/N)$ であることに変わりはない.
よって提案手法に関する解析は元のモデルに関する解析と同じ結果を得る.

平均経路長の保存

$N = 2000, K = 3, P$ の値を $2^{-20}, 2^{-12}, 2^{-10}, 2^{-8}$ と変化させたとき, 複数のグラフについて経路長ごとの割合を測定しプロットしたものである.

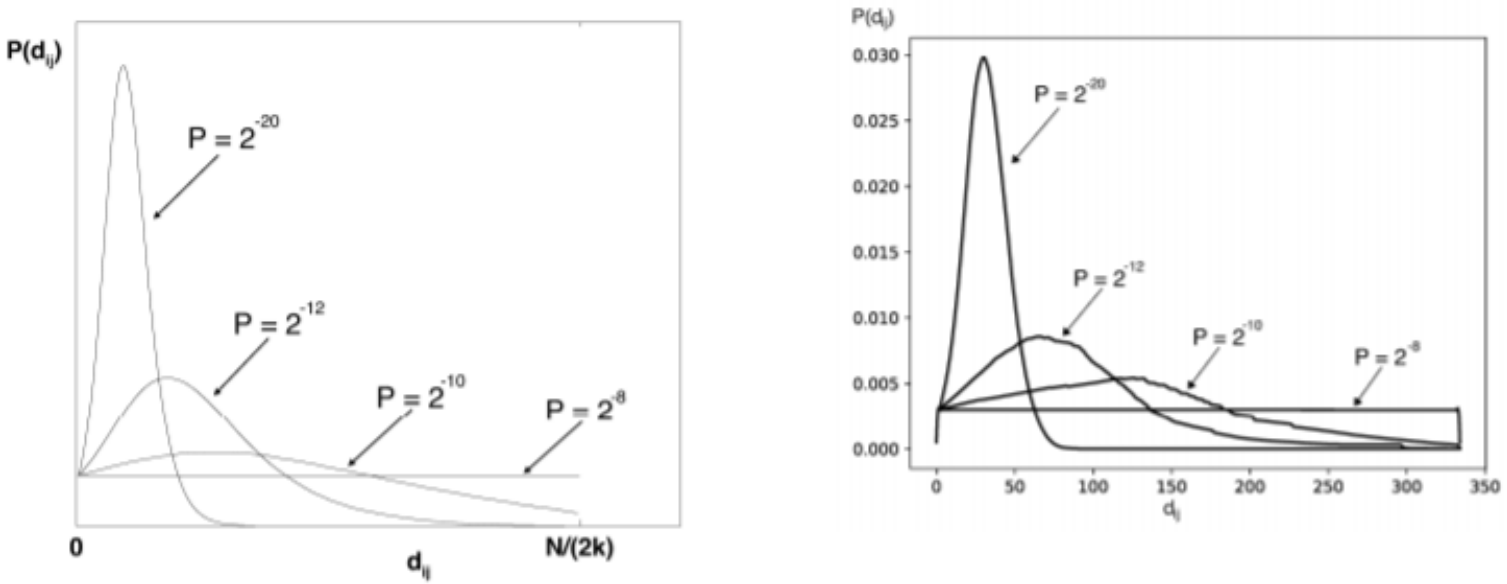

図 6: ノード間距離の割合（左:従来手法, 右:提案手法）

提案手法の実装

Python による実装

逐次的な実装である.Python で実装されたグラフ処理のフレームワークの NetwoekX がある. 提案手法の実装では辺の再接続処理を逐次的に各頂点独立に行った後, 改めて隣接リストの双方向化を行っている. つまり, 頂点の u の持つ辺 (u, v) についてその接続先の頂点 v の隣接リストに u を入れる処理を後で一斉に行う.

Hadoop MapReduce による実装

Map タスクは入力としてそのタスクで処理を担当する頂点番号の範囲 $[s, e)$ やモデルのパラメータ N, K, P を受け取る. そのあと各頂点 $v \in [s, e)$ に対して $2K$ 本の辺の接続先を計算する. そのあと, それらの各々の接続先 ω について,(v,ω) と (ω, v) という key-value ペアを生成する.
Reduce タスクは, 生成させるグラフの各頂点 v 毎に存在する.

実験環境

以下に示す環境の PC16 台からなるクラスタを用いた.HDFS のチャンクサイズは４ MB とした.

CPU：Intel(R) Core(TM) i5 6500 @ 3.20GHz

メモリ：16GB (8GB x2, PC4 17000)

OS：Ubuntu 14.04.5 LTS

Java：Oracle JDK 1.8.0_131

Hadoop：1.2.1

Python：3.4.3

実験

まず,$K = 4, P = 0.4$ と,$N = 100$ 万,1000 万,1 億に対してそれぞれの実装でグラフを生成するのにかかった時間を測定した.

逐次実装の比較

実装そのものが拙いこと、双方向化の処理に時間がかかっているという理由から,1.5 倍ほど提案手法は時間がかかっている.

Hadoop 実装と他の逐次実装の比較

グラフサイズが小さい場合 Hadoop 実装で十分なタスク分割が行われず, 逐次実装のほうが速いがグラフサイズを大きくすると分散並列実行の効果が現れるようになる.

実験結果

表 1　ランダムグラフ生成時間 (秒)

頂点数	NetworkX (既存逐次実装)	Python 実装 (提案逐次実装)	Hadoop 実装 (提案並列実装)				
			1 台	2 台	4 台	8 台	16 台
100 万	6	9	19	17	16	16	17
1000 万	66	92	83	56	30	24	21
1 億	–	–	833	414	213	139	78
10 億	–	–			3811	1288	651

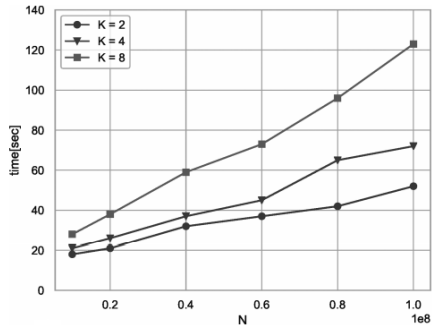

図 5　N だけを変化させたときの生成時間 ($P = 0.4$)

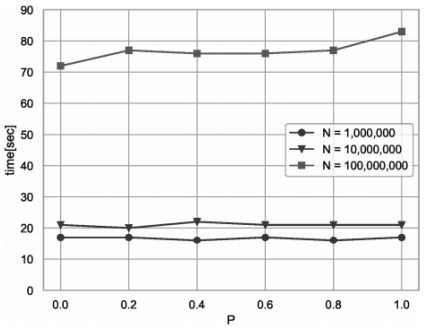

図 6　P だけを変化させたときの生成時間 ($K = 4$)

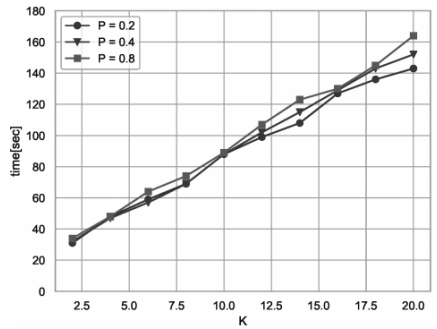

図 7　K だけを変化させたときの生成時間 ($N = 6000$ 万)

表 2　各ブロックサイズにおけるグラフ生成時間 (秒)

生成頂点数	ブロックサイズ				
	1	2	4	8	10
100 万	12	14	13	17	16
1000 万	17	14	17	18	22
1 億	82	62	48	47	55

↑K=2,P=0.4

まとめ

まとめ

・グラフ解析プログラム等の評価に使われる、ランダムグラフ生成モデルの Watts-Strogtz モデルの分散並列化手法を提案した.
・制約を追加しても元のモデルの特徴は破壊していないことを確認した.
・分散並列実行による生成時間の短縮と, 一台の計算機では対応しにくいサイズのグラフ生成が容易になった.

今後の課題

・提案手法のより効率的な実装
・Watts-Strogatz モデルをベースとした派生モデルへの手法の適用