ScaleGraph/Python Pregel Model Computation API

ScaleGraph/Pythonが提供するPregel Model Computation APIについて記載します。

Using API

次のようにxpregelモジュールをimportし、xpregel.XPregelBaseを継承したクラスを記述します。このクラスのインスタンスに対してrun()を呼び出すと、Pregel計算のためのクロージャがシリアライズされます。

> Note: run()を呼び出したときに、Pregel計算がスタートすべきです。改修予定です。

```
import xpregel
```

```
class UserDefinedPregelComputation(xpregel.XPregelBase):
    def compute(self, ctx, messages):
        # Definition of Pregel computation
    def aggregator(self, outputs):
        # Definition of Pregel aggregator
        # returns aggregated value
    def terminator(self, superstep, aggregatedValue):
        # Definition of Pregel terminator
        # returns true when terminate superstep iteration

if __name__ == '__main__':
    pr = UserDefinedPregelComputation()
    pr.run()
```

Class Definition

UserDefinedPregelComputation.compute(self, ctx, messages)

- ctx: Pregel計算時に頂点を表現するVertexCotextクラスのインスタンスが渡されます。
- messages: Pregel計算時に頂点に届いたメッセージのListが渡されます。メッセージの型はfloatです。

> Note: 自由なメッセージ型を利用できるようにすることをめざしたいのですが、ScaleGraphのオリジナルのコードを大きく変更する必要があり、現時点では実装を断念しています。

UserDefinedPregelComputation.aggregator(self, outputs)

- outputs: Pregel計算時にaggregateする値のListが渡されます。この値の型はfloatです。

> Note: 自由なAggregate valueの型を利用できるようにすることをめざしたいのですが、ScaleGraphのオリジナルのコードを大きく変更する必要があり、現時点では実装を断念しています。

UserDefinedPregelComputation.terminator(self, superstep, aggregatedValue)

- superstep: int型でsuperstep の値が入ります。
- aggregatedValue: float型でaggregate結果が入ります。

VertexContext.superstep

int型でsuperstepの値が入ります。

VertexContext.numVertices

int型で計算中のグラフ全体の頂点数が入ります。

VertexContext.outEdges

int型のListでout edgeの頂点IDリストが入ります。

VertexContext.inEdges

int型のListでin edgeの頂点IDリストが入ります。

VertexContext.sendMessage(self, dst_vertexId, message)

out edgeの頂点IDであるdst_vertexIdに向けてdouble型のmessageを送ります。

VertexContext.sendMessageToAllNeighbors(self, message)

out edgeすべての頂点に対してdouble型のmessageを送ります。

VertexContext.getVertexValue(self)

頂点ごとに（superstepをまたいで）保存してある値を取得します。

VertexContext.setVertexValue(self, value)

頂点ごとに（superstepをまたいで）保存する値をセットします。

VertexContext.aggregate(self, value)

あとで、aggregatorに渡される値をセットします。

VertexContext.log(self, *objs)

ログを出力します。

Sample Code

PageRank計算は次のように記述されます。

```
import xpregel

class PageRank(xpregel.XPregelBase):

    def compute(self, ctx, messages):
        if ctx.superstep == 0:
            value = 1.0 / ctx.numVertices
        else:
            value = 0.15 / ctx.numVertices + 0.85 * sum(messages)
        ctx.aggregate(abs(value - ctx.getVertexValue()))
        ctx.setVertexValue(value)
        numOutEdges = len(ctx.outEdges)
        if numOutEdges > 0:
            msg = value / numOutEdges
            for d in ctx.outEdges:
                ctx.sendMessage(d, msg)
#         if numOutEdges > 0:
#             ctx.sendMessageToAllNeighbors(value / numOutEdges)

    def aggregator(self, outputs):
        return sum(outputs)

    def terminator(self, superstep, aggregatedValue):
#         ctx.log("PageRank at superstep " + superstep + " = " +
aggValue + "\n")
        return superstep == 30

if __name__ == '__main__':
```

```
    pr = PageRank()
    pr.run()
```