Appendix

A. The iteration processes of DSP

Using the same deducing method as BSP, we get the formalized representation of DSP iteration processes shown in the following steps:

(1) In each iteration, every processor conducts Δ steps of local computation and local data update. To ensure its conciseness and generality, we just show the transformation on the processor whose segment is from x_p to x_q, and the derivation is shown in Table IV.

(2) After Δ steps of local computation and local data update, it follows a global data synchronization. After that, a superstep restarts again. The process could be described as following:

$$
\begin{aligned}
X_\Delta &= (x_{\Delta,0},\ x_{\Delta,1},\ x_{\Delta,2},\ \dots,\ x_{\Delta,n}) \\
X_{\Delta+1}^{(p,q)} &= X_\Delta \otimes F^{(p,q)} \\
&= (\dots,\ x_{\Delta,p-1},\ \biguplus_{i=0}^{n} F_{i,p}(x_{\Delta,p}),\ \dots, \\
&\qquad \biguplus_{i=0}^{n} F_{i,q}(x_{\Delta,q}),\ x_{\Delta,q+1},\ \dots,\ x_{\Delta,n}) \\
&\qquad \vdots \\
X_{2\Delta}^{(p,q)} &= (\dots,\ x_{\Delta,p-1},\ g^{\Delta-1}(\underbrace{\alpha_{2\Delta,p}}_{\boxed{\biguplus_{i=0}^{n} F_{i,p}(x_{\Delta,p})}},\ \underbrace{\beta_{2\Delta,p}}_{\boxed{\biguplus_{i\notin(p,q)} F_{i,p}(x_{\Delta,p})}}),\ \dots, \\
&\qquad g^{\Delta-1}(\alpha_{2\Delta,q},\ \beta_{2\Delta,q}),\ x_{\Delta,q+1},\ \dots,\ x_{\Delta,n}), \\
&\mathit{in\ which\ any\ item\ can\ be\ derived\ as\ follows:} \\
x_{2\Delta,p} &= g^{\Delta-1}(\biguplus_{i=0}^{n} F_{i,p}(\underbrace{x_{\Delta,p}}),\ \biguplus_{i\notin(p,q)} F_{i,p}(\underbrace{x_{\Delta,p}})), \\
&\qquad \boxed{x_{\Delta,p} = g^{\Delta-1}(\alpha_p, \beta_p)} \\
&= g^{\Delta-1}(\biguplus_{i=0}^{n} F_{i,p}\left(g^{\Delta-1}(\alpha,\ \beta)\right),\ \biguplus_{i\notin(p,q)} F_{i,p}\left(g^{\Delta-1}(\alpha,\ \beta)\right))
\end{aligned}
$$

(3) Through the deduction of (1) and (2), we find that the output of current bulk round will be used as the input of next bulk round. Thus one round of iteration process can be expresssed as:

$$h(g(x,y)),$$

in which $g(x,y) = \biguplus_{i=0}^{n}(\biguplus_{i\in(p,q)} F_{i,j}(x),\ y)$.

(4) Through l rounds bulk synchronization, we could get $X_{l\Delta}$:

$$X_{l\Delta} = (x_{l\Delta,0},\ x_{l\Delta,1},\ \dots,\ x_{l\Delta,m}),$$

in which any item can be presented as

$$x_{l\Delta,p} = h^l(x_{\Delta,p}), p = 0, 1, \dots, m$$

$$X_{t0} = (x_{t0,0}, x_{t0,1}, \ldots, x_{t0,n})$$

$$X_{t1}^{(p,q)} = X_{t0} \otimes F^{(p,q)} = (x_{t0,0}, x_{t0,1}, \ldots, x_{t0,n}) \otimes \begin{pmatrix} 1 & 0 & \ldots & 0 & F_{0,p} & \ldots & F_{0,q} & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & F_{1,p} & \ldots & F_{1,q} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 & F_{p-1,p} & \ldots & F_{p-1,q} & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & F_{p,p} & \ldots & F_{p,q} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & F_{q,p} & \ldots & F_{q,q} & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & F_{q+1,p} & \ldots & F_{q+1,q} & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & F_{n,p} & \ldots & F_{n,q} & 0 & \ldots & 1 \end{pmatrix}$$

Using the "matrix multiplication" − like transformation $\otimes$, we get the follows, only x_p to x_q are updated.

$$= (x_{t0,0}, \ldots,\ x_{t0,p-1},\ \biguplus_{i=0}^{n} F_{i,p}(x_{t0,i}),\ \biguplus_{i=0}^{n} F_{i,p+1}(x_{t0,i}),\ \ldots,\ \biguplus_{i=0}^{n} F_{i,q}(x_{t0,i}),\ x_{t0,q+1},\ \ldots,\ x_{t0,n})$$

$$\begin{aligned} X_{t2}^{(p,q)} &= X_{t1}^{(p,q)} \otimes F^{(p,q)} \\ &= (x_{t0,0}, \ldots,\ x_{t0,p-1},\ \biguplus_{i=0}^{n} F_{i,p}(x_{t1,i}),\ \biguplus_{i=0}^{n} F_{i,p+1}(x_{t1,i}),\ \ldots,\ \biguplus_{i=0}^{n} F_{i,q}(x_{t1,i}),\ x_{t0,q+1},\ \ldots,\ x_{t0,n}) \\ &= (x_{t0,0}, \ldots,\ x_{t0,p-1},\ \biguplus(\biguplus_{i\in(p,q)} F_{i,p}(\biguplus_{i=0}^{n} F_{i,p}(x_{t0,i})),\ \biguplus_{i\notin(p,q)} F_{i,p}(x_{t0,i})), \\ &\qquad \ldots,\ \biguplus(\biguplus_{i\in(p,q)} F_{i,q}(\biguplus_{i=0}^{n} F_{i,q}(x_{t0,i})),\ \biguplus_{i\notin(p,q)} F_{i,q}(x_{t0,i})),\ x_{t0,q+1},\ \ldots, x_{t0,n}) \end{aligned}$$

Let $\alpha_p = \biguplus_{i=0}^{n} F_{i,p}(x_{t0,i}),\ \beta_p = \biguplus_{i\notin(p,q)} F_{i,p}(x_{t0,i})$

$$= (x_{t0,0}, \ldots,\ x_{t0,p-1},\ \biguplus(\biguplus_{i\in(p,q)} F_{i,p}(\alpha_p), \beta_p),\ \ldots,\ \biguplus(\biguplus_{i\in(p,q)} F_{i,q}(\alpha_q), \beta_q),\ x_{t0,q+1},\ \ldots, x_{t0,n})$$

Let $g(x,y) = \biguplus(\biguplus_{i\in(p,q)} F_{i,p}(x), y)$

$$\begin{aligned} &= (x_{t0,0}, \ldots,\ x_{t0,p-1},\ g(\alpha_p, \beta_p),\ \ldots,\ g(\alpha_q, \beta_q),\ x_{t0,q+1},\ \ldots, x_{t0,n}) \\ X_{t3}^{(p,q)} &= X_{t2}^{(p,q)} \otimes F^{(p,q)} \\ &= (x_{t0,0}, \ldots,\ x_{t0,p-1},\ \biguplus(\biguplus_{i\in(p,q)} F_{i,p} \biguplus(\biguplus_{i\in(p,q)} F_{i,p}(\alpha_p),\ \beta_p),\ \beta_p), \\ &\qquad \ldots,\ \biguplus(\biguplus_{i\in(p,q)} F_{i,q} \biguplus(\biguplus_{i\in(p,q)} F_{i,q}(\alpha_p),\ \beta_q),\ \beta_q),\ x_{t0,q+1},\ \ldots, x_{t0,n}) \\ &= (x_{t0,0}, \ldots,\ x_{t0,p-1},\ \biguplus(\biguplus_{i\in(p,q)} F_{i,p}(g(\alpha_p, \beta_p)),\ \beta_p),\ \ldots,\ \biguplus(\biguplus_{i\in(p,q)} F_{i,q}(g(\alpha_q, \beta_q)),\ \beta_q),\ x_{t0,q+1},\ x_{t0,q+1},\ \ldots, x_{t0,n}) \\ &\vdots \\ X_{\Delta}^{(p,q)} &= X_{\Delta-1}^{(p,q)} \otimes F^{(p,q)} \\ &= (x_{t0,0}, \ldots,\ x_{t0,p-1},\ \biguplus(\biguplus_{i\in(p,q)} F_{i,p}(\cdots \biguplus(\biguplus_{i\in(p,q)} F_{i,p}(\alpha_p),\ \beta_p), \ldots, \beta_p), \beta_p), \\ &\qquad \ldots,\ \biguplus(\biguplus_{i\in(p,q)} F_{i,q}(\cdots \biguplus(\biguplus_{i\in(p,q)} F_{i,q}(\alpha_q),\ \beta_q), \ldots, \beta_q), \beta_q),\ x_{t0,q+1},\ \ldots,\ x_{t0,n}) \\ &= (x_{t0,0}, \ldots,\ x_{t0,p-1},\ g(g(\ldots g(\alpha_p, \beta_p), \ldots, \beta_p), \beta_p), \ldots,\ g(g(\ldots g(\alpha_q, \beta_q), \ldots, \beta_q), \beta_q),\ x_{t0,q+1}, \ldots, x_{t0,n}) \\ &= (x_{t0,0}, \ldots,\ x_{t0,p-1}, g^{\Delta-1}(\alpha_p, \beta_p), \ldots,\ g^{\Delta-1}(\alpha_q, \beta_q),\ x_{t0,q+1}, \ldots, x_{t0,n}) \end{aligned}$$

TABLE V: Derivation of First Δ Steps of Local Computation and Data Exchange.

in which

$$x_{\Delta,p} = g^{\Delta-1}(\alpha_p,\ \beta_p)$$

in which $\alpha_p = \biguplus_{i=0}^{n} F_{i,p}(x_{t0,i})$ and $\beta_p = \biguplus_{i \notin (p,q)} F_{i,p}(x_{t0,i})$, finally, we can get

$$x_{l\Delta,p} = h^l(g^{\Delta-1}(\alpha_p,\ \beta_p)). \quad (**)$$

B. Implementation of Algorithms appearing in Evaluation

1) Implementation of PageRank:

(S_1) First, check that whether the algorithm variant is convergent when Δ=2. According to the original iteration formula of Pagerank

$$PR(p_i)_{k+1} = \frac{(1-c)}{N} + c \sum_{p_j \in N(p_i)} \frac{PR(p_j)_k}{L(p_j)},$$

when Δ=2, the iteration formula becomes

$$PR(p_i)_{k+2} = \frac{(1-c)}{N} + c \sum_{p_j \in N(p_i)} \frac{\frac{(1-c)}{N} + c \sum_{p_m \in N(x_j)} \frac{PR(p_m)_k}{L(p_m)}}{L(p_m)},$$

we could find that the new iteration also converge.

(S_2) Screen proper Δ.

(S_3) Construct DSP program shown as in Algorithm 3.

Algorithm 3 Pagerank

```
procedure DSP_ALGO(G, v0)
    iter_count← 0
    while True do
        Computing_PR()
        if iter_count % delta == 0 then
            DataExchange()
            if is_convergent() then
                break
        iter_count++
```

2) Implementation of SSSP, Jacobi Method, SGD: Using the guidelines in section **??**, we could get the similar DSP algorithm as in pagerank.