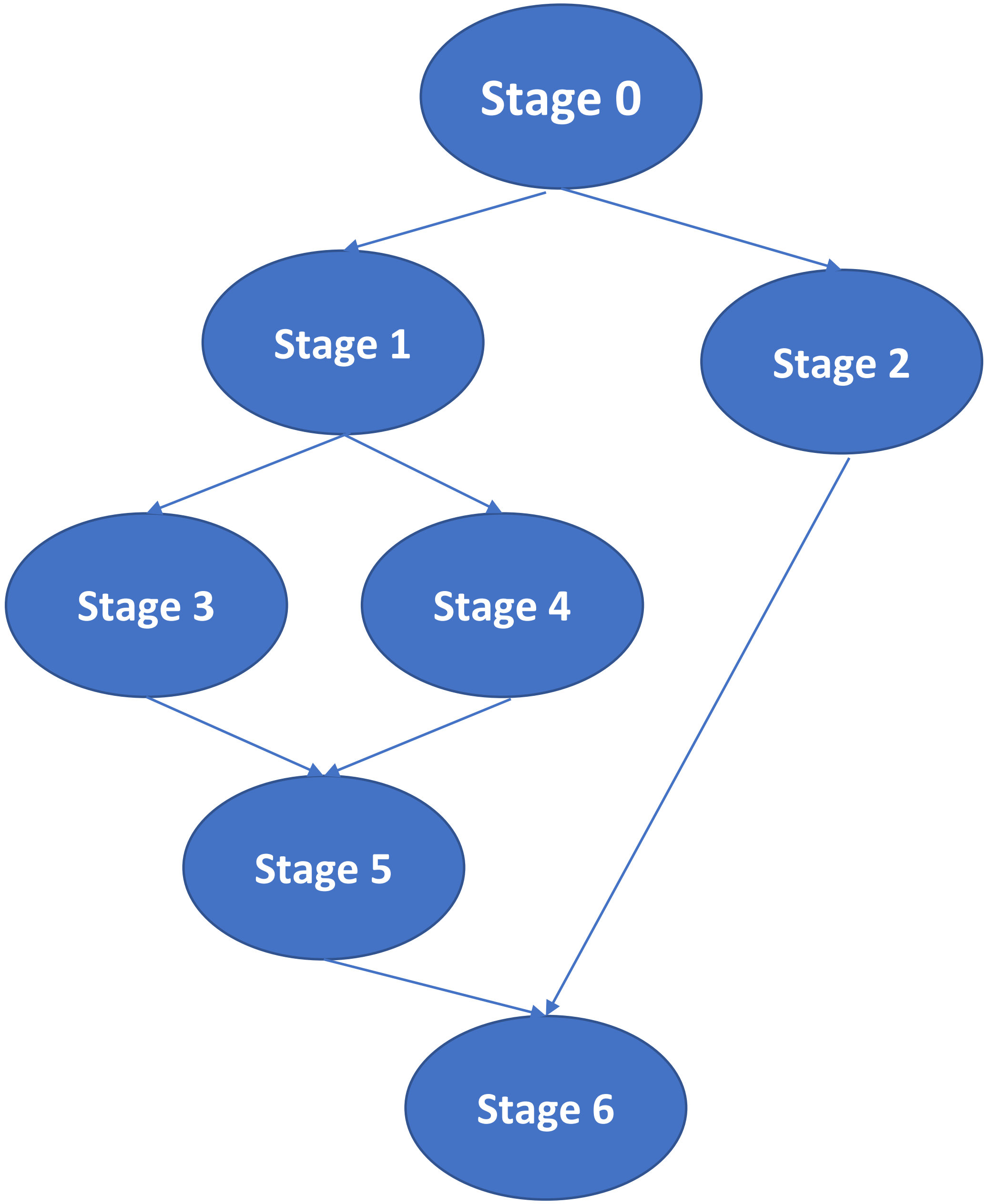
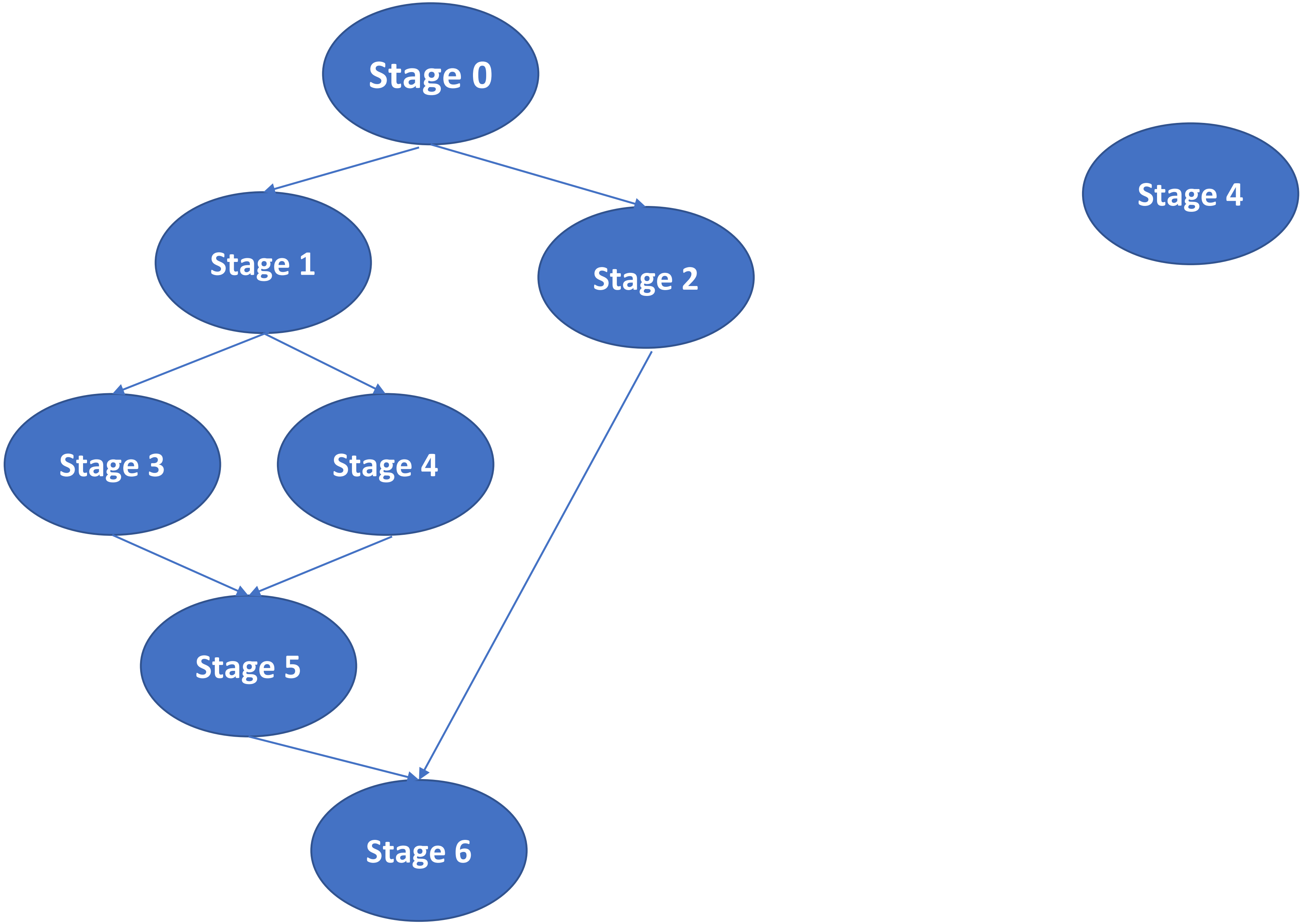
Libra and the Art of Task Sizing in Big-Data Analytic Systems

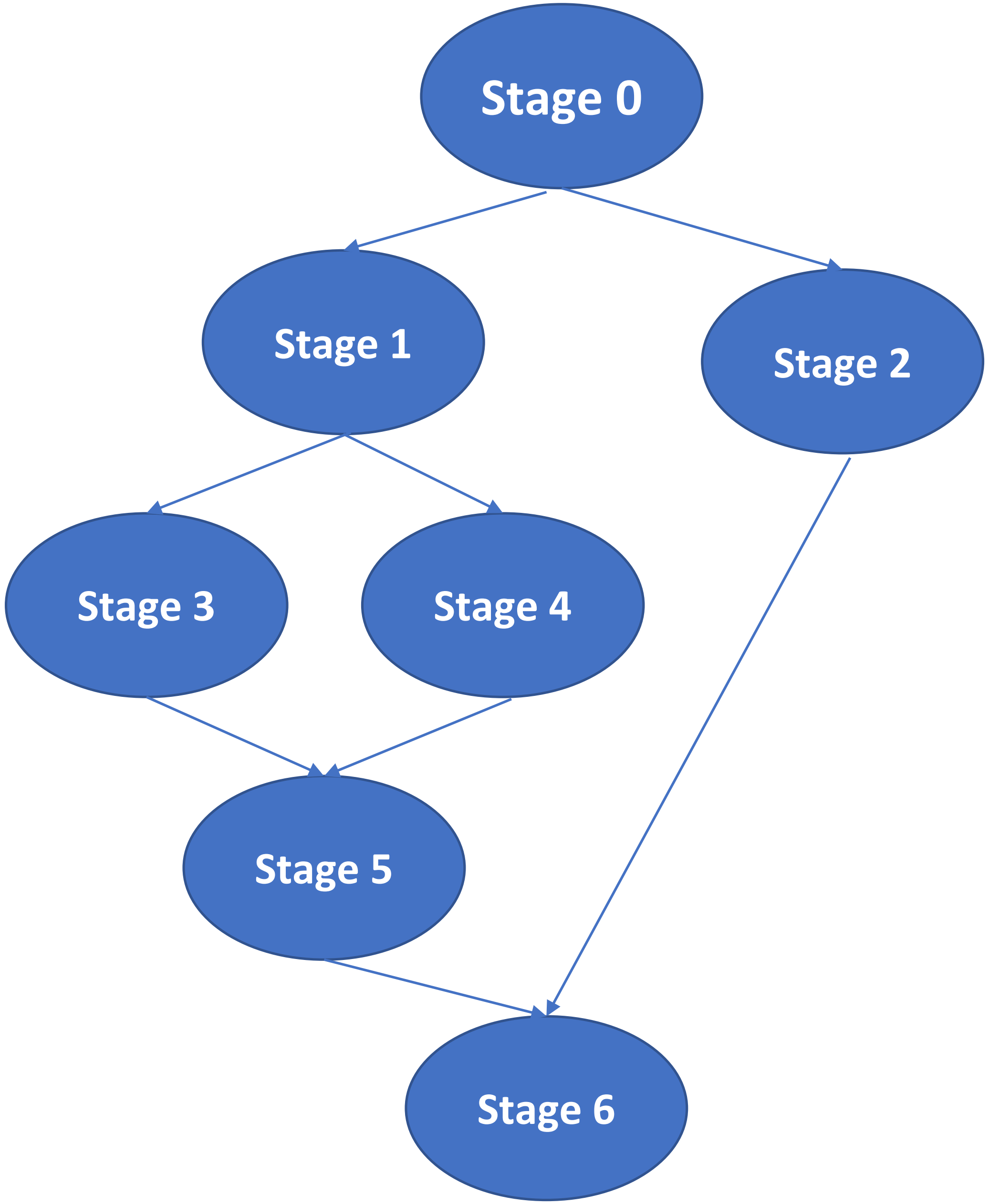
Rui Li, Peizhen Guo, Bo Hu, Wenjun Hu

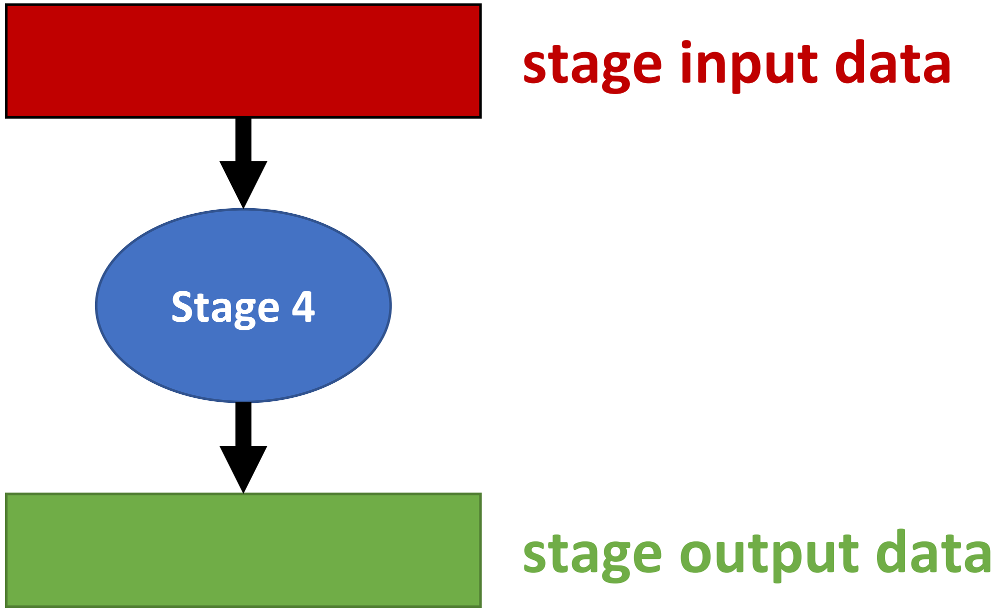
Yale University

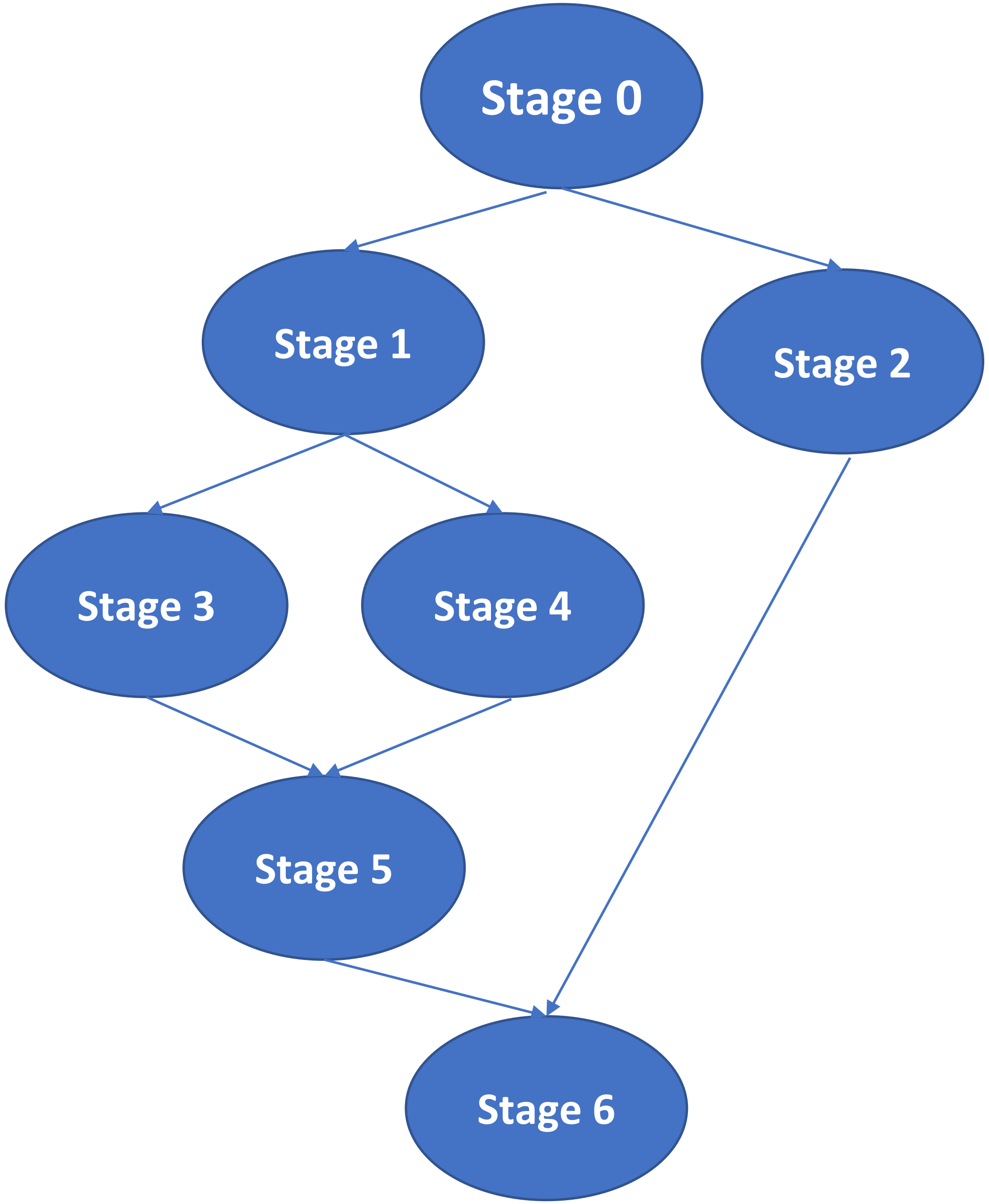
Background

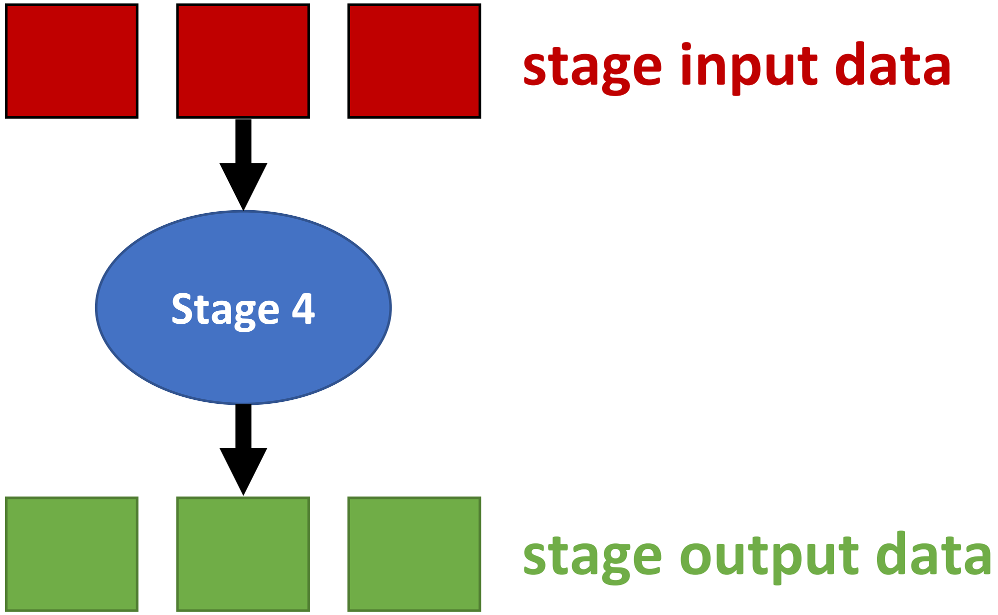

Stage 0
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Stage 6

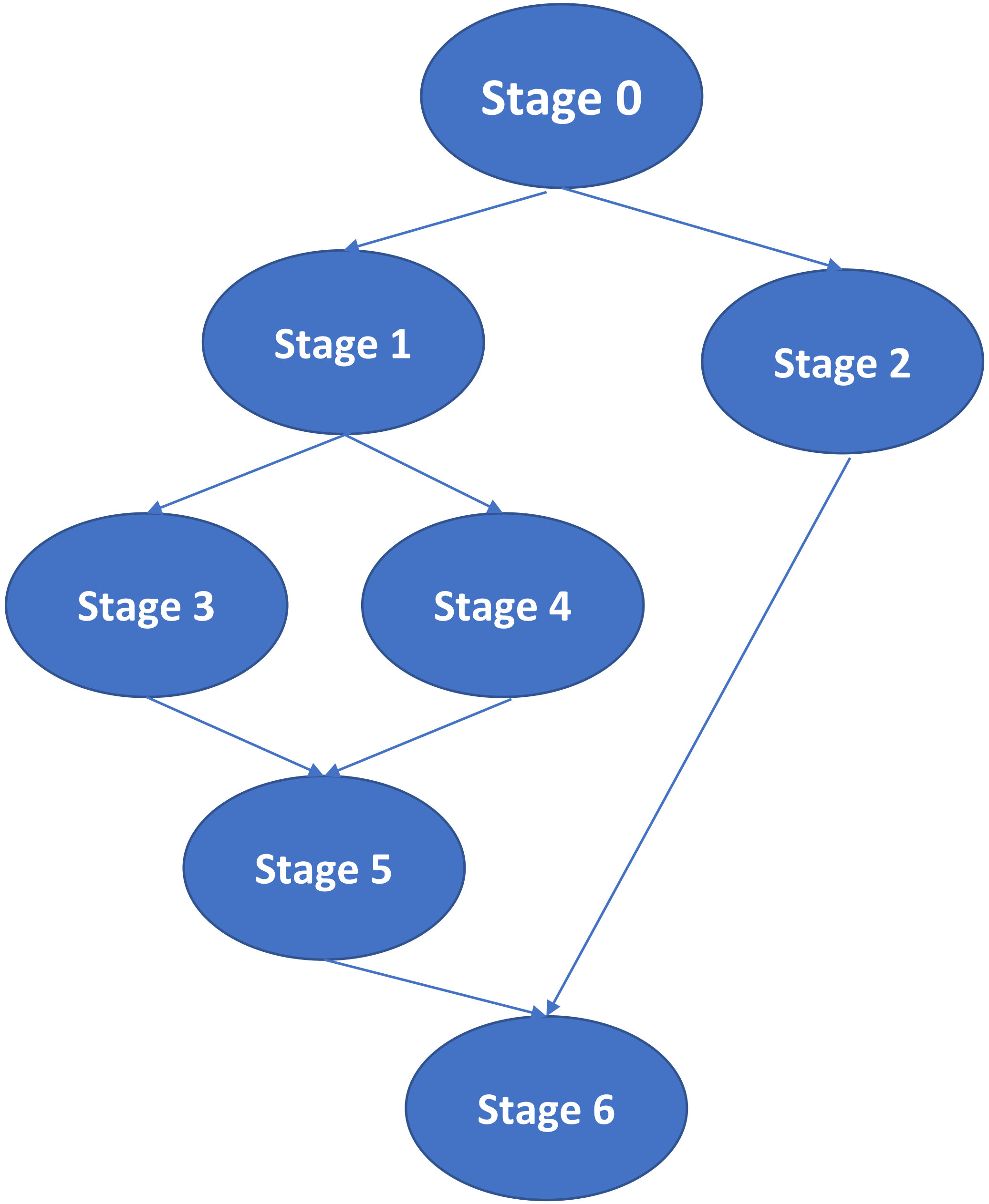

Stage 0
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Stage 6
Stage 4

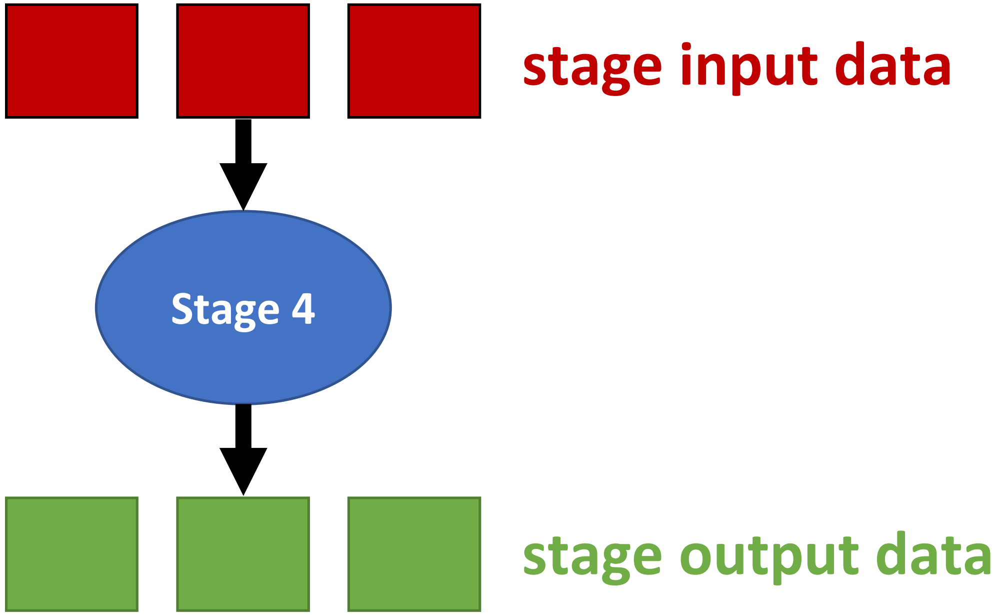

Stage 0
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Stage 6


stage input data
Stage 4
stage output data

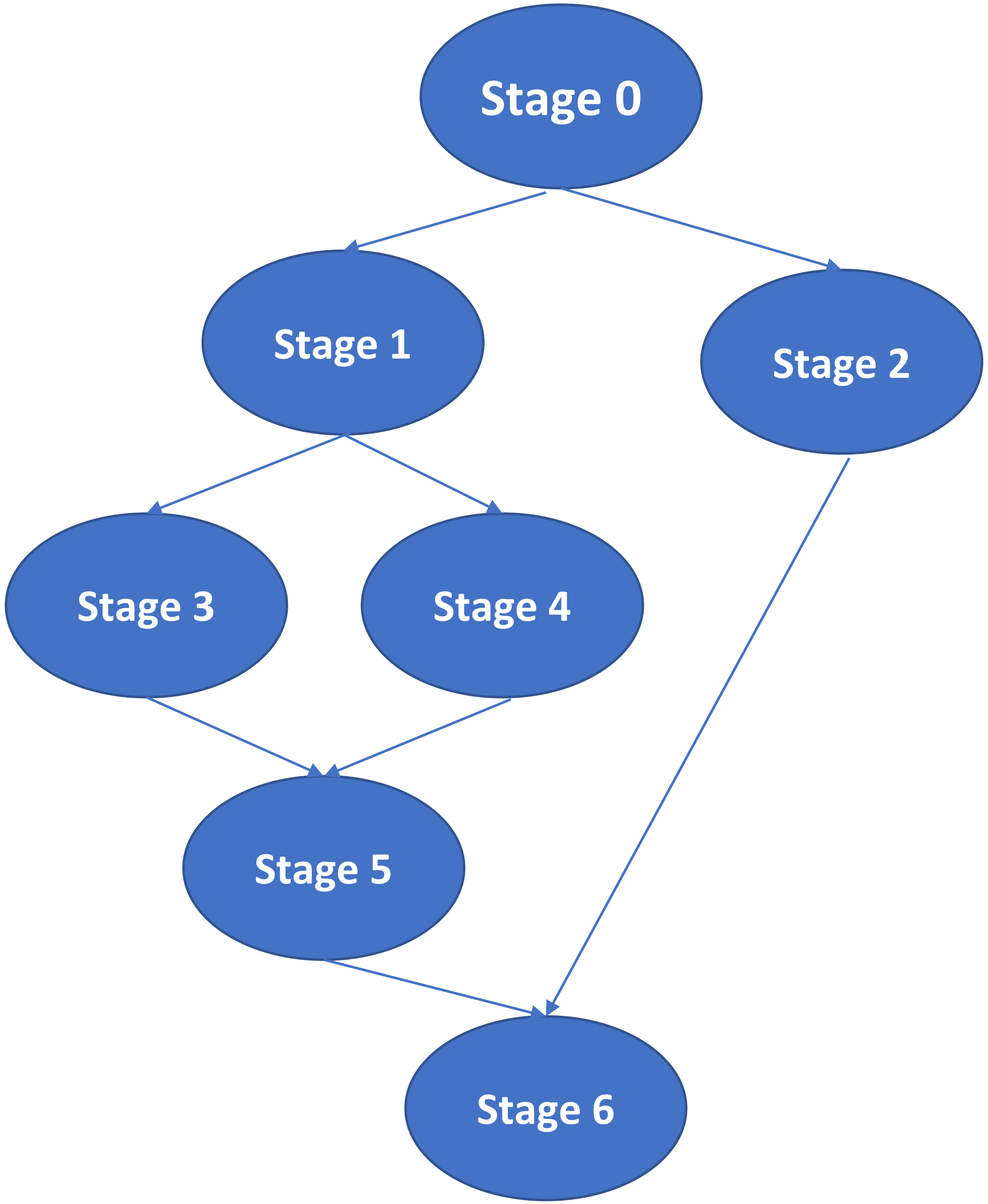

Stage 0
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Stage 6

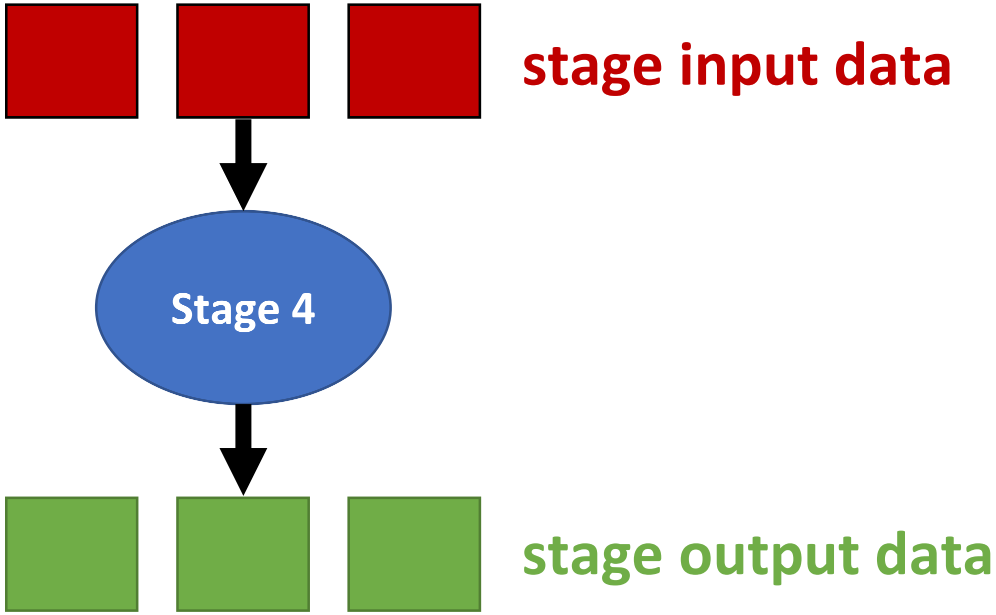

stage input data
Stage 4
stage output data





How to set task size?



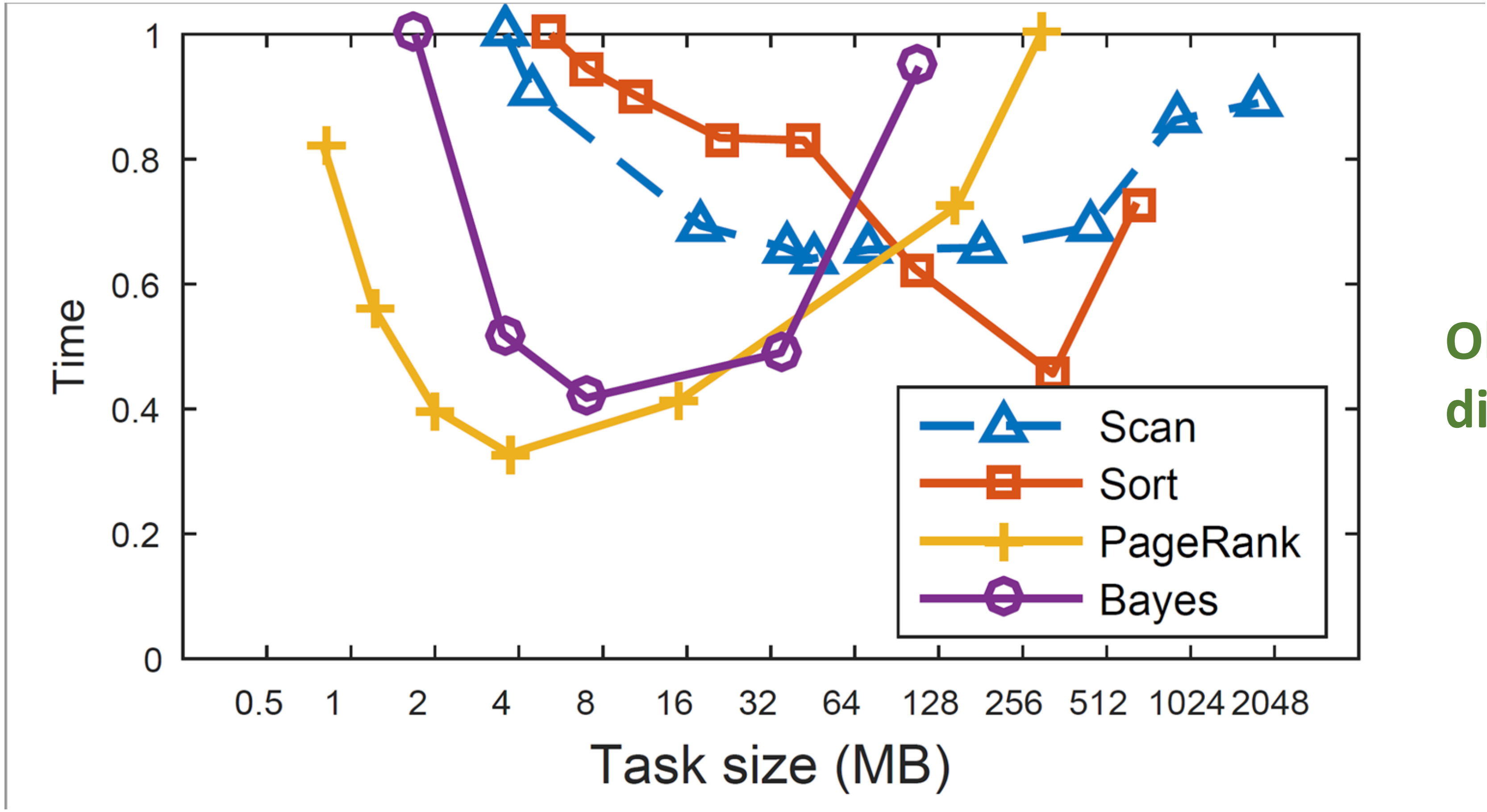


How to set task size?

-- User experience

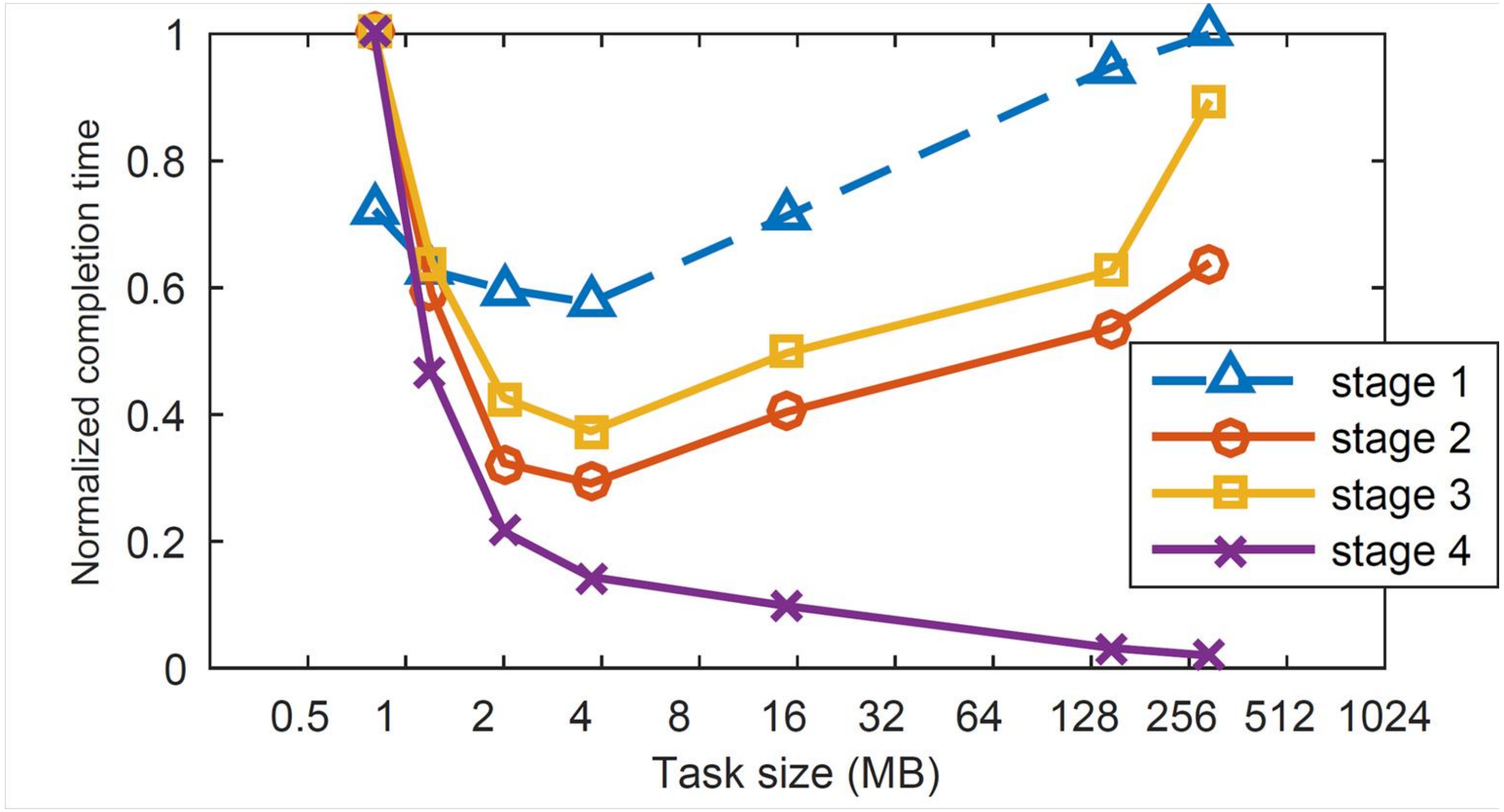
-- System default value

The importance of task sizing



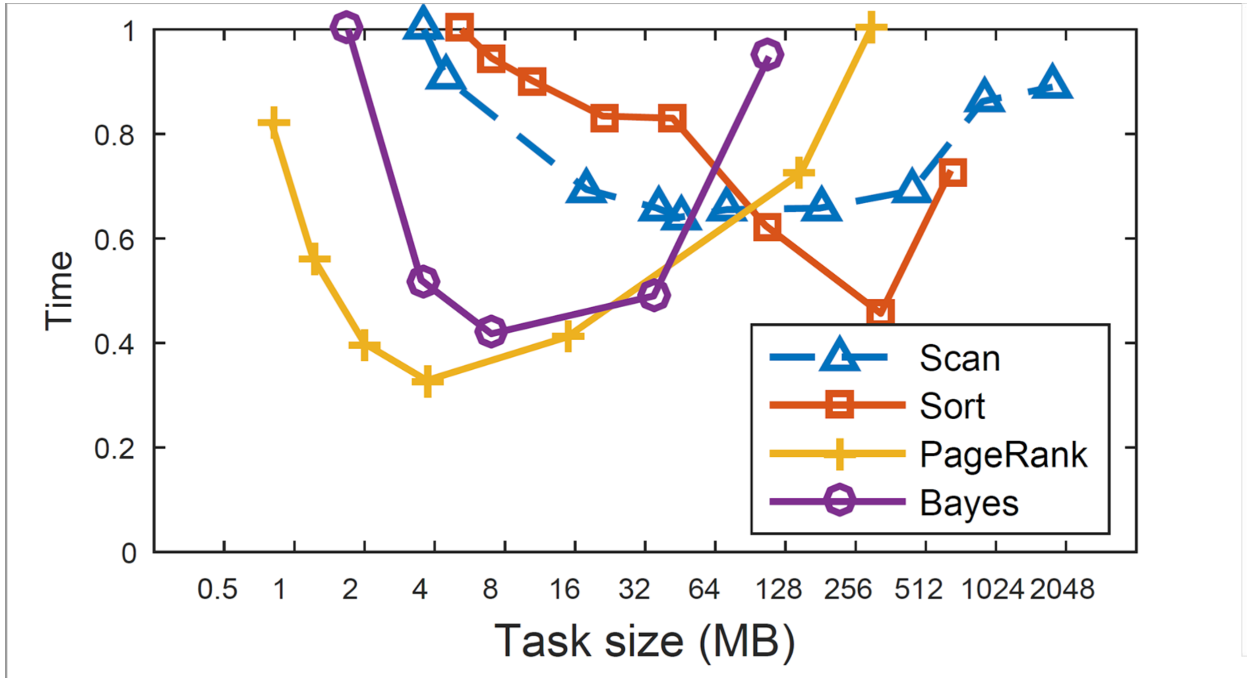
Normalized stage completion time vs task size

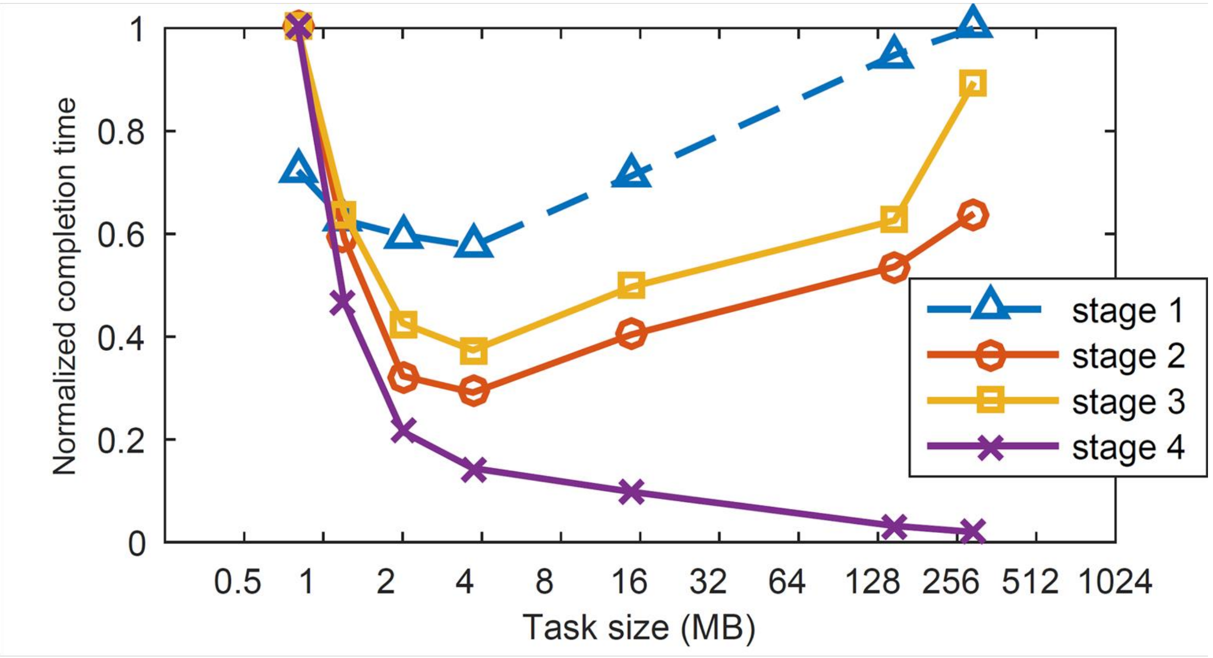
Observation 1: diff jobs have diff optimal task sizes



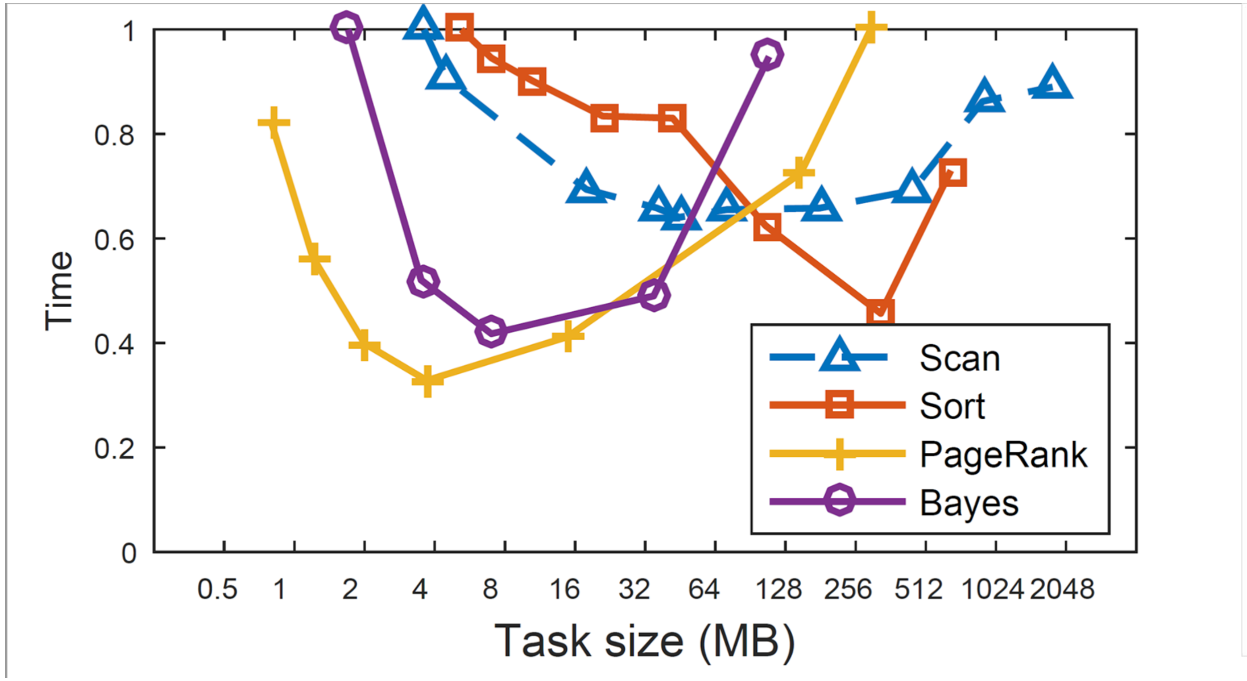
Observation 2: diff stages have diff optimal task sizes

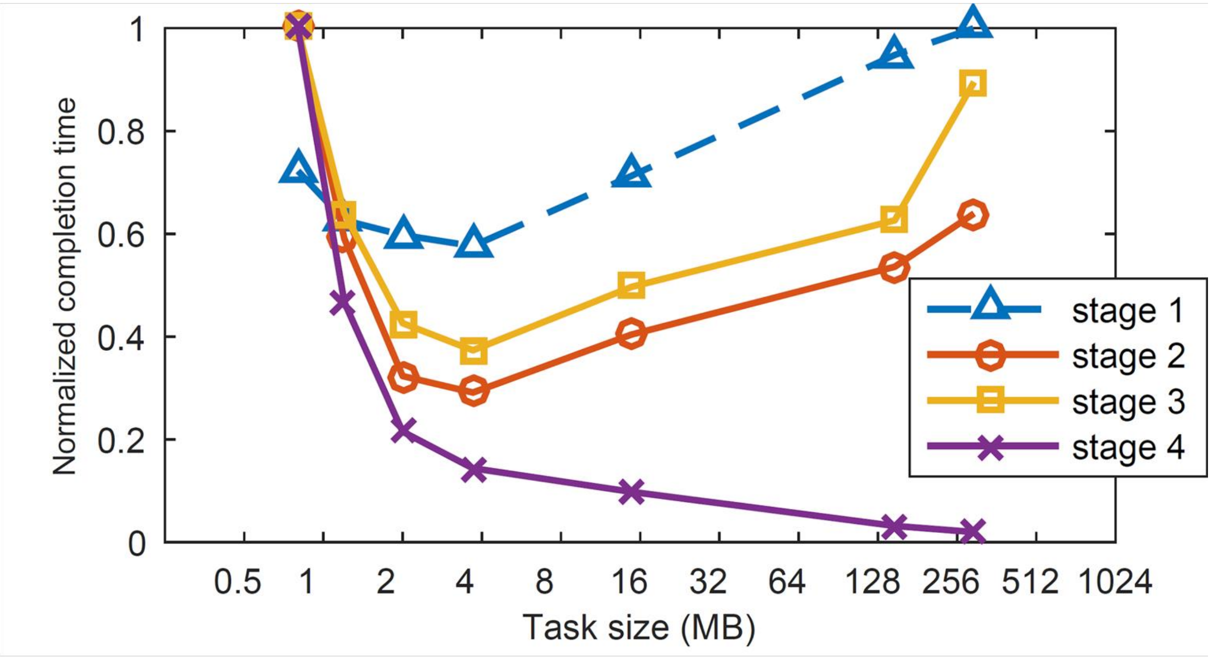
PageRank stage completion time vs task size





1. Proper task sizing is important

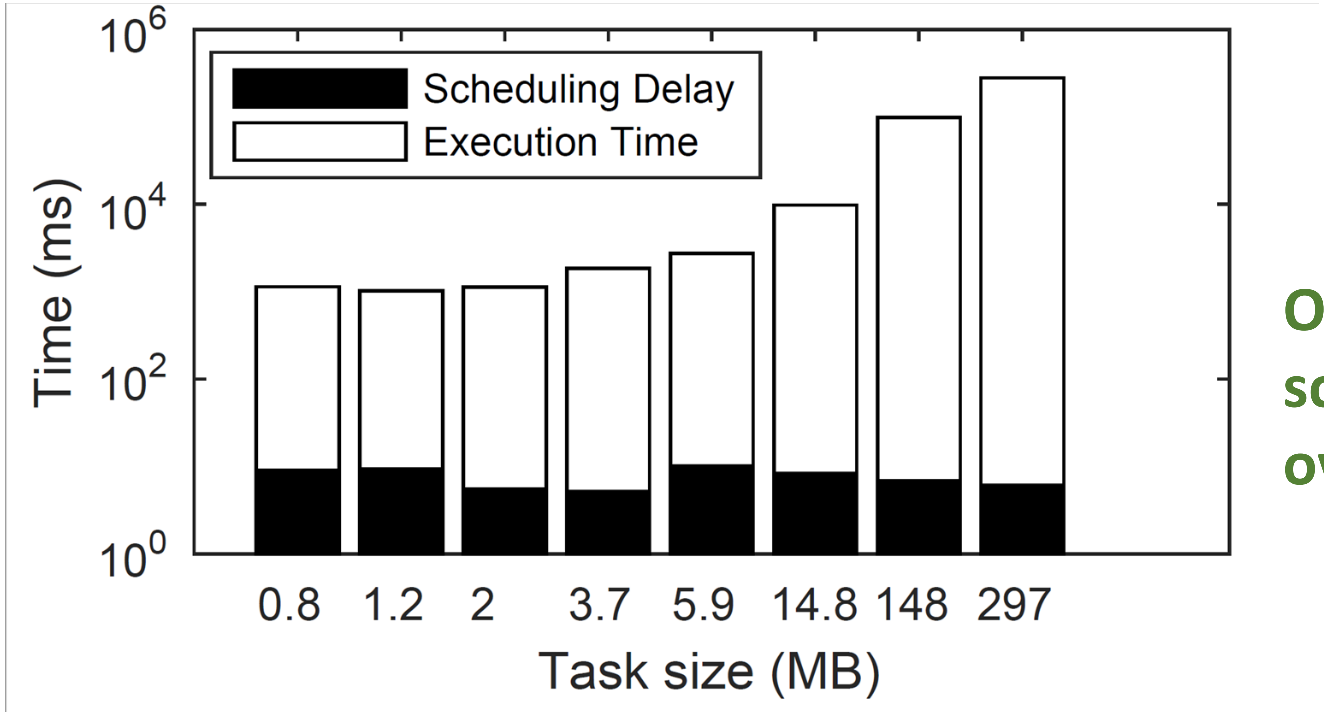




1. Proper task sizing is important

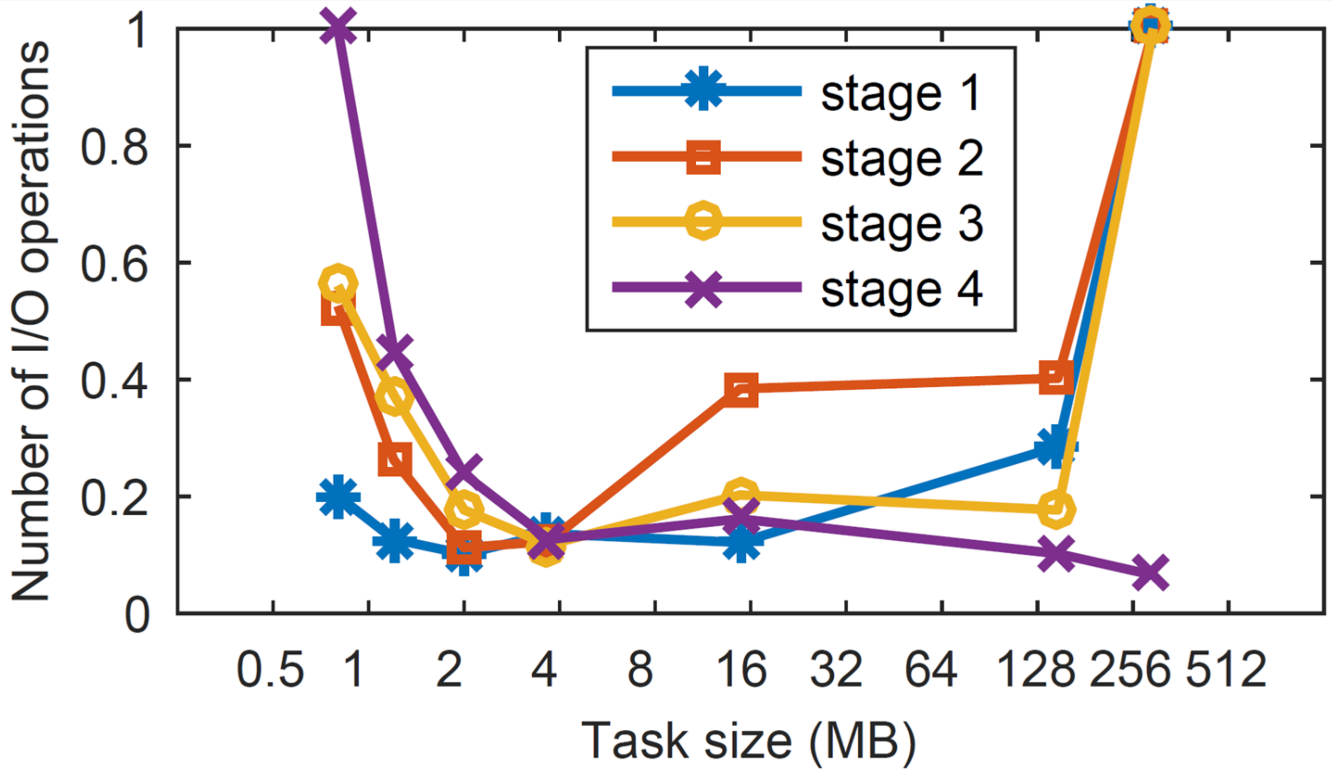
2. U-curve pattern

Analysis of U-curve pattern



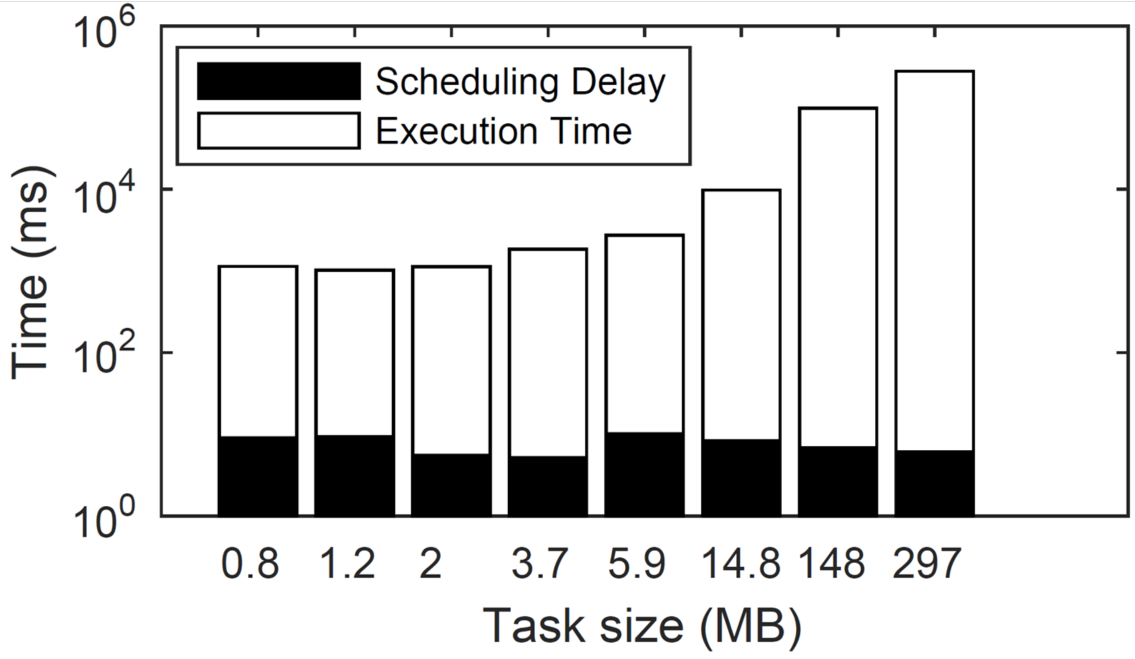
Per-task overhead for PageRank stage 1

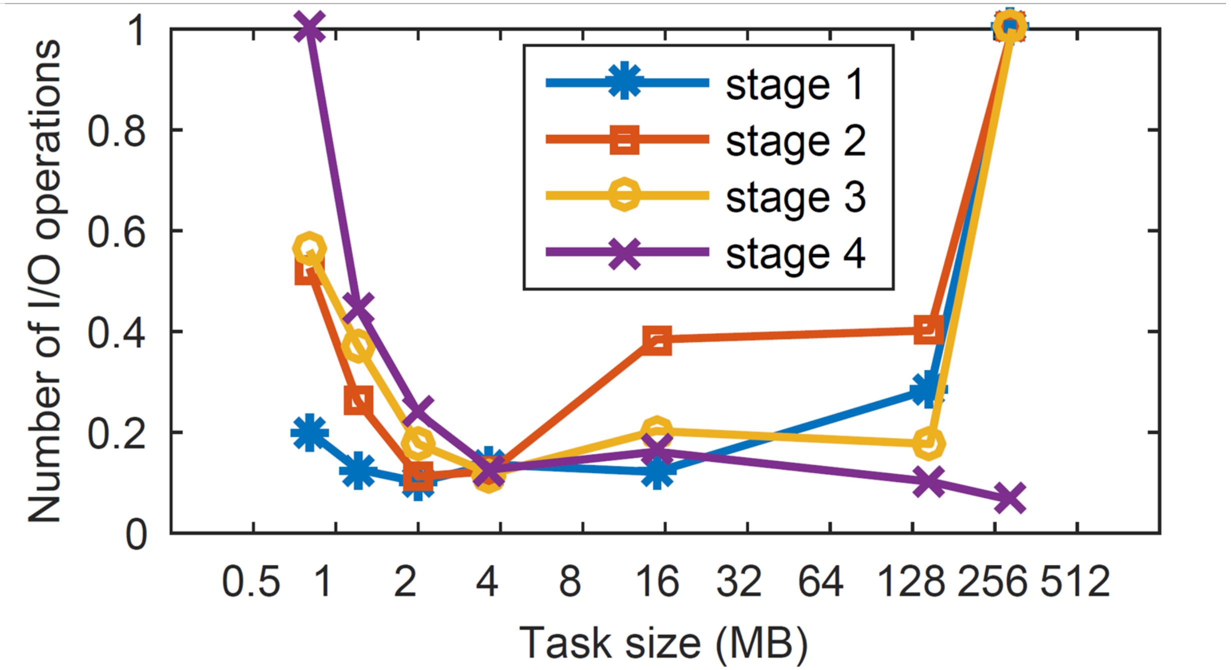
Observation 3: tasks have similar scheduling delay and system overhead regardless of task sizes



Observation 4:
small size => fail to do batch processing
large size => memory swapping

# of IO ops for different stages of PageRank





Small task size => high aggregated overhead, no batch processing

Large task size => memory swapping

System design

- Strawman solution

$$S_{k+1} = S_k + \alpha \times dS_k$$

$$dS_k = \frac{R_k - R_{k-1}}{S_k - S_{k-1}}$$

R_k: processing rate of task k
S_k: input size of task k

Refinement 1: ADAM optimization

$$S_{k+1} = S_k + \alpha \times dS_k$$

$$dS_k = \frac{R_k - R_{k-1}}{S_k - S_{k-1}}$$

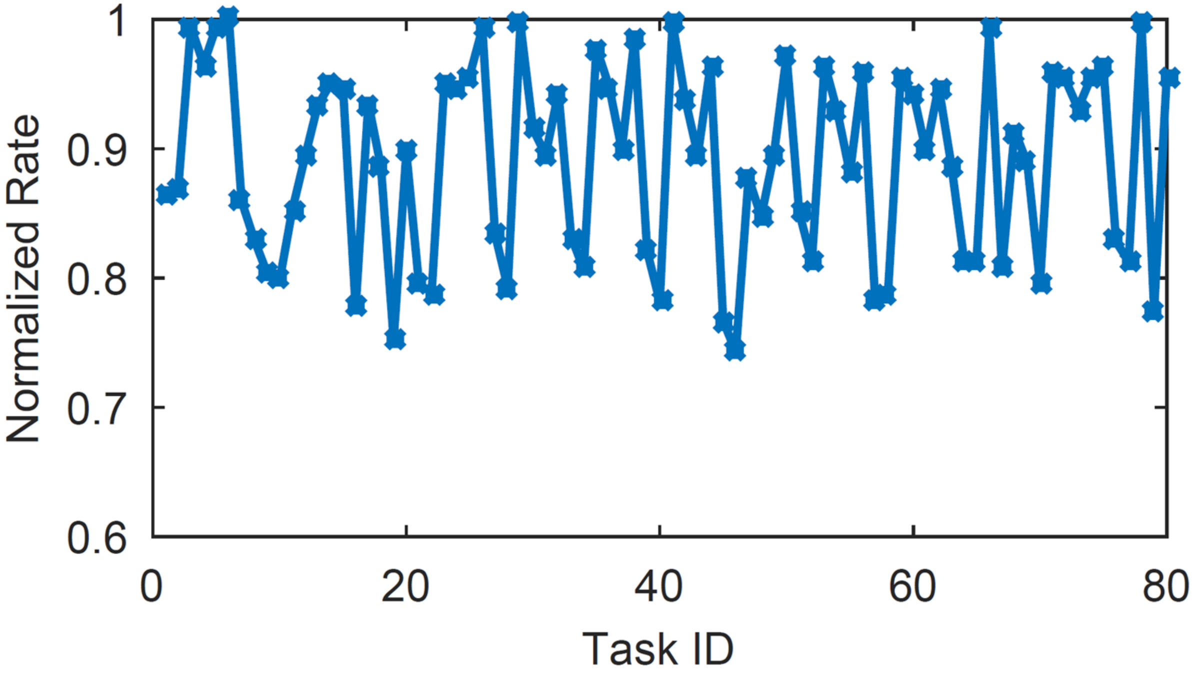
$$S_{k+1} = S_k + \alpha_k \times \frac{m_k}{\sqrt{v_k} + \epsilon}$$

$$\alpha_k = \frac{\alpha_0}{\sqrt{k}}$$

$$m_k = \beta_1 \times m_{k-1} + (1 - \beta_1) \times dS_k$$

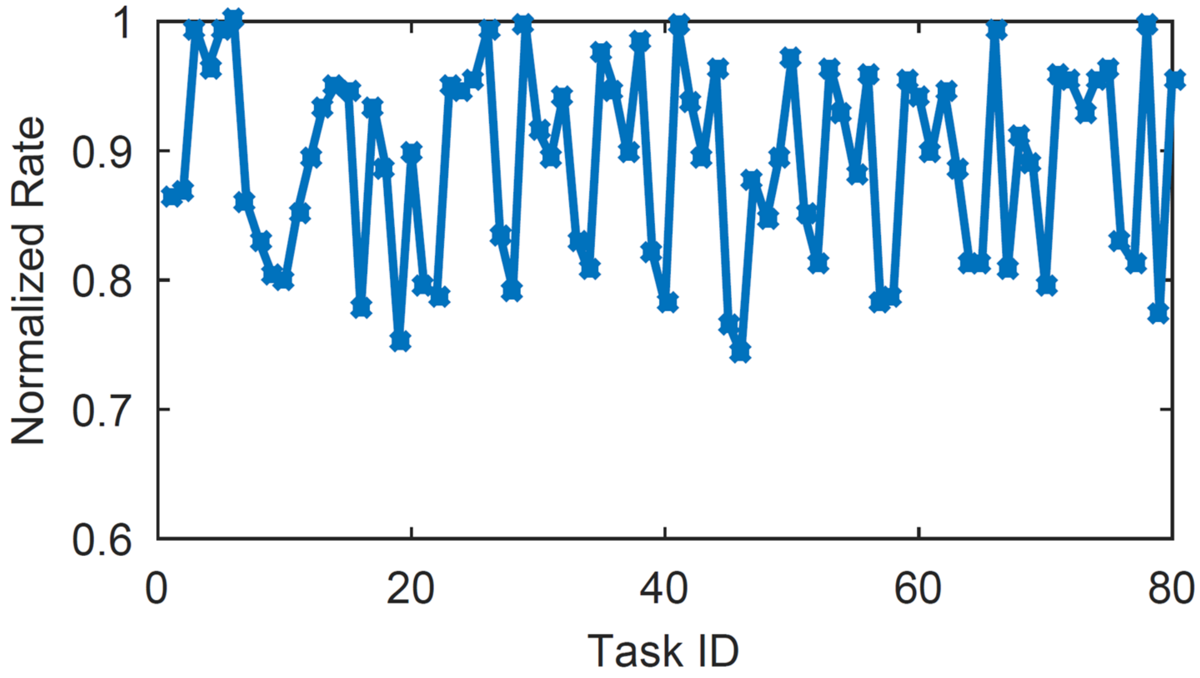
$$v_k = \beta_2 \times v_{k-1} + (1 - \beta_2) \times dS_k^2$$

Refinement 2: noise filtering



Task processing rate fluctuation for stage 1 of PageRank

Refinement 2: noise filtering

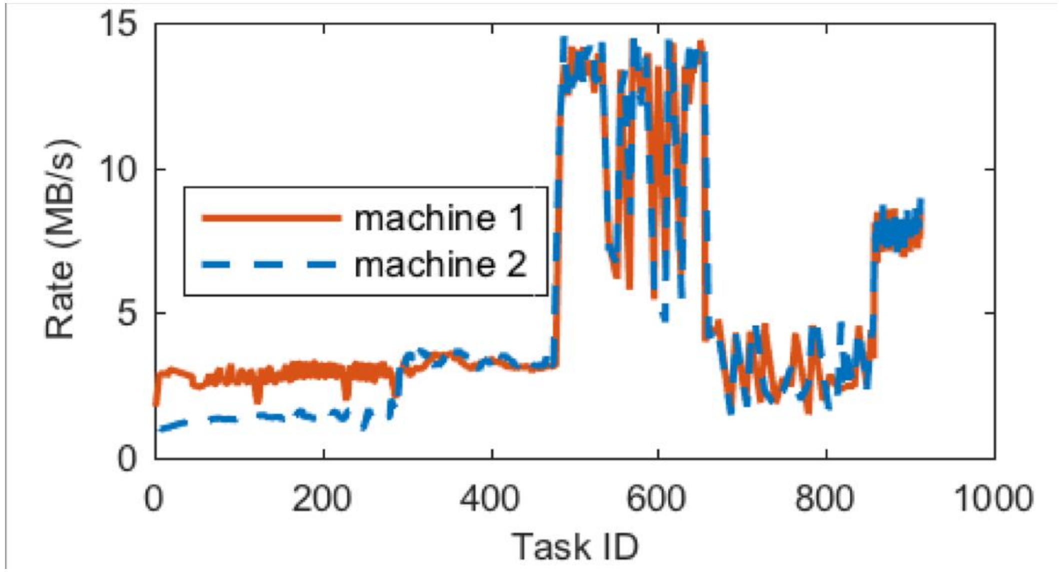


Task processing rate fluctuation for stage 1 of PageRank

$$C_{S_k} = \alpha \times C_{S_{k-1}} + (1 - \alpha) \times R_{S_k}$$

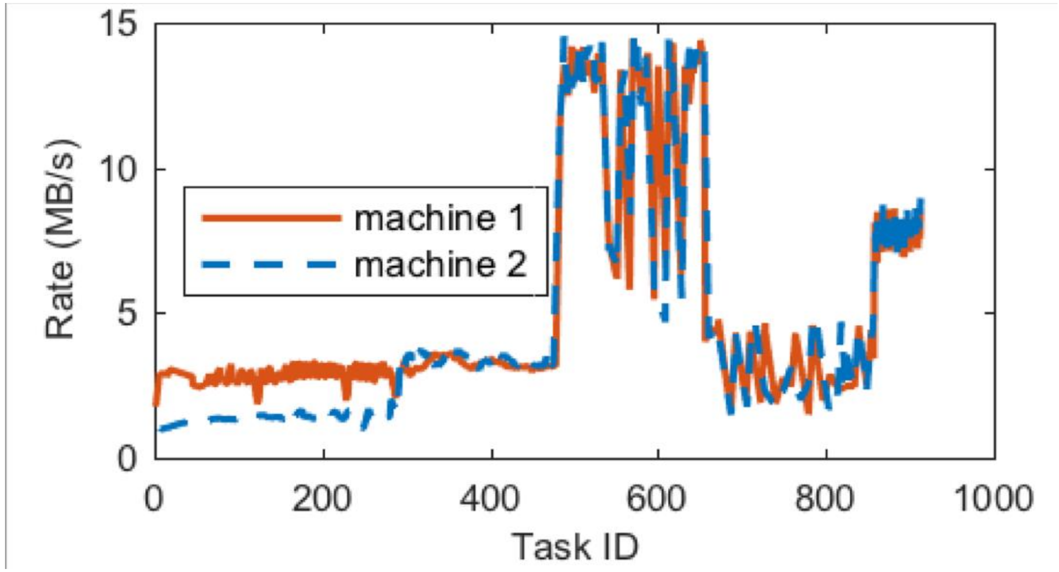
$$R_{S_k} = \frac{\sum_{i=1}^{X} R_i}{X} \text{ (task } 1 \sim X \text{ all have size } S_k\text{)}$$

Refinement 3: contention avoidance



PageRank over two machines

Refinement 3: contention avoidance



PageRank over two machines

$$C_j = \alpha \times C_j + (1 - \alpha) \times R_i$$

(task i is running on machine j)

$$C = \alpha \times C + (1 - \alpha) \times C_j$$

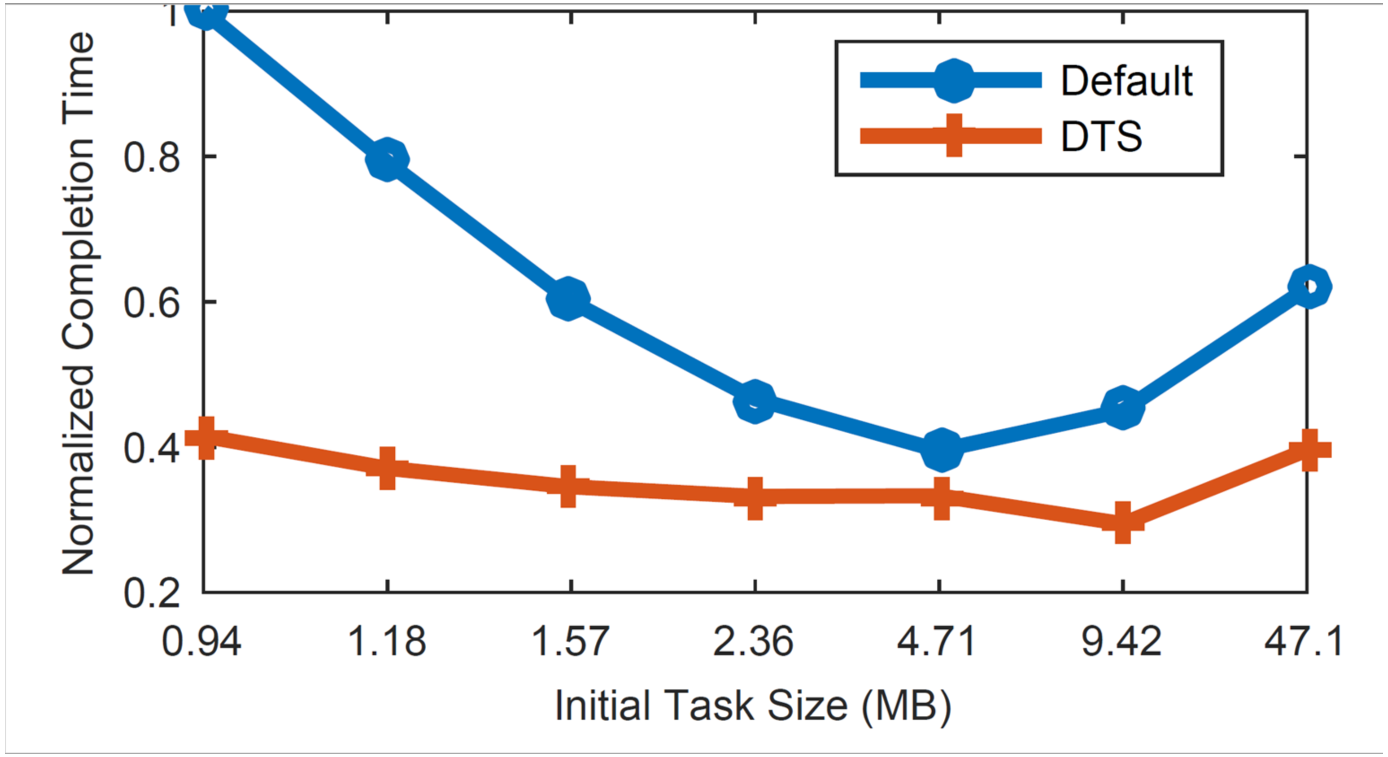
(C is the avg rate of all machines)

if $C_j < (1 - thres) \times C$,
switch machine j

Evaluation

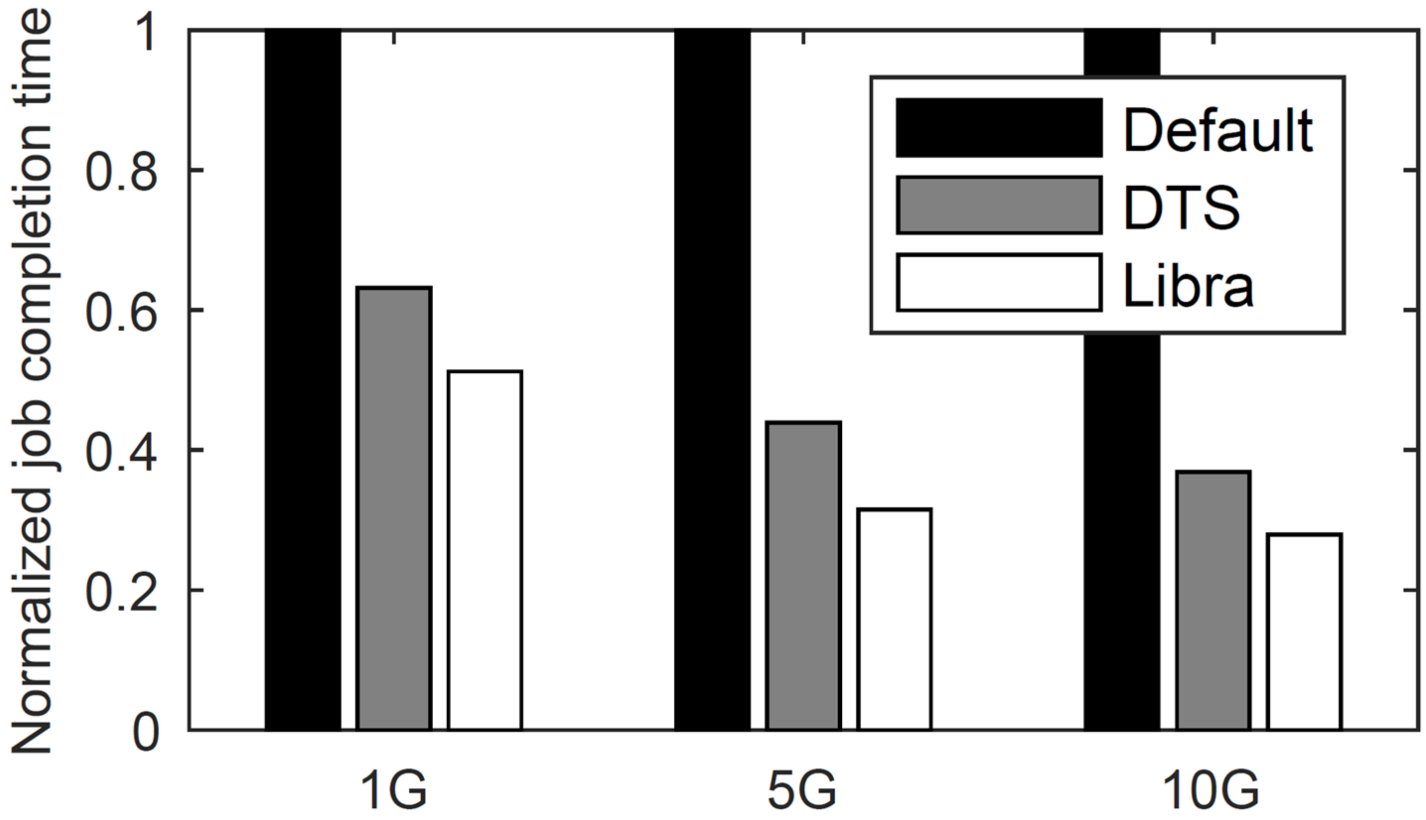
- 8 m4.xlarge VMs from EC2
- Workloads generated from HiBench

Initial task size effect



PageRank completion time over diff. initial task size

Libra performance



PageRank completion time with diff. input data size

Q&A