Distributed Calculator

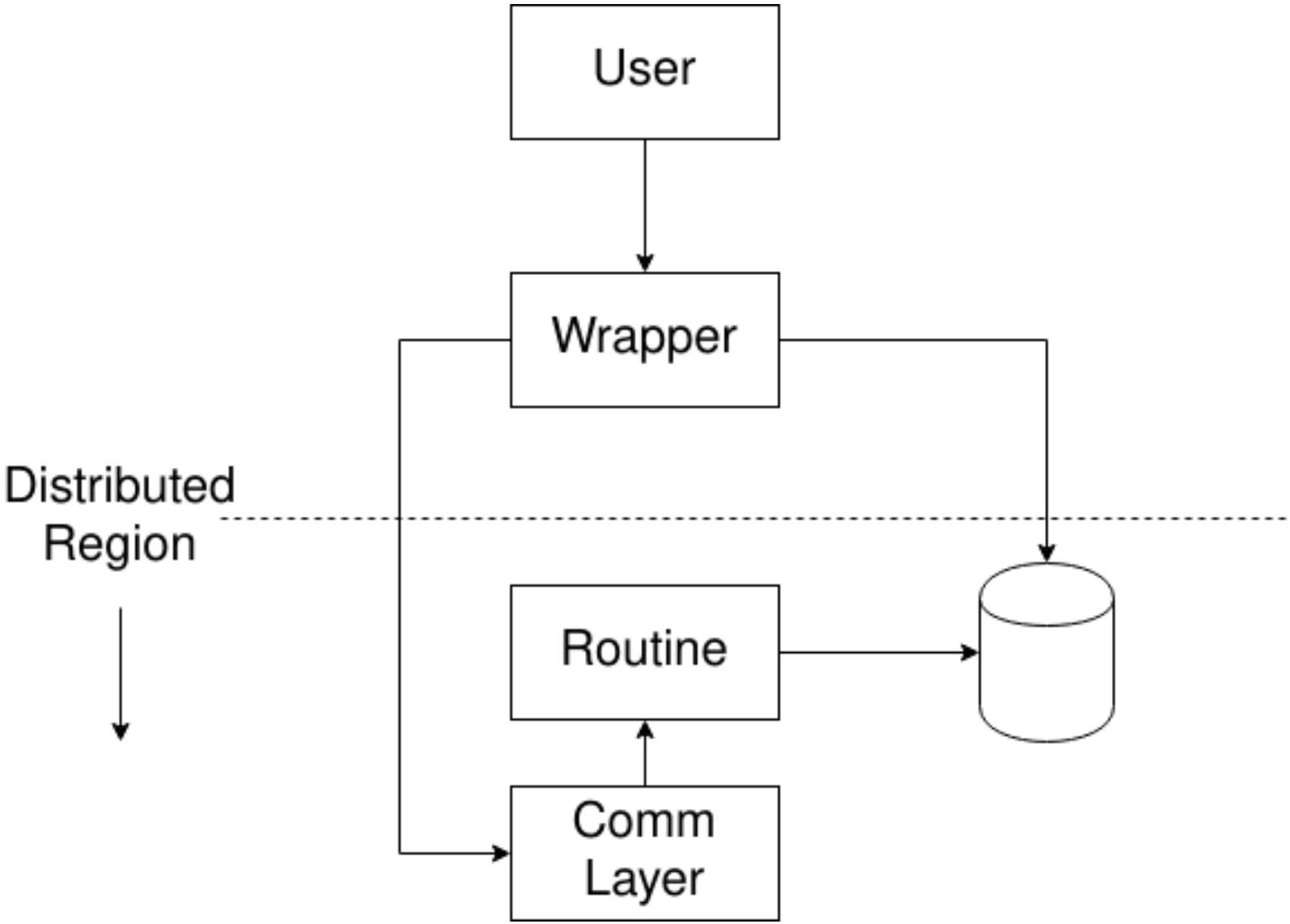
Anuj Singh, Pradeep Kumar
Project Network
Distributed Systems (CS525) [2019-2020-W]
anujs@iitbhilai.ac.in
pradeepk@iitbhilai.ac.in

May 4, 2020

Features

- 8 Distributed Algorithms: Eigenvalues, Gaussian Elimination, Inverse, Transpose, Multiplication, Eigenvectors, QR Decomposition, Determinant
- Complete abstraction of the end user to the distributed part of the computation
- Communication and process creation is handled by our own code
- Minimal set of common Python Packages required to run

Architecture Overview



Cost models

Assume a matrix of size NxN, number of process p with t_s being the startup time and t_w being the communication time.

- Matrix Multiplication:

$$N^3/p + t_s p + t_w N^2 p \tag{1}$$

- Gaussian Elimination:

$$N^3/p + t_s p + 2t_w N(N+1)/p + t_w(N-1) + N^2 \tag{2}$$

- Determinant:

$$N^3/p + t_s p + t_w N^2 p + t_w(N-1) \tag{3}$$

Cost models

- Inverse:

$$2N^3p + t_sp + 4t_wN^2/p + t_w(N-1)$$

- QR Decomposition:

$$T(N,p,t_s,t_w) = t_sp + \frac{(N^2/p^2)(N/2+1)^2}{8}N + \frac{(N/p)(N/p+1)(2N/p+1)}{12}N + \frac{(N/p)(N/p+1)(2N/p+1)}{3}t_w - N/p(N/p+1)t_w$$

- Eigenvalues: Let K be the number of iteration for diagonalization of matrix

$$KT(N,p,t_s,t_w)N^3$$

Cost Models

- Transpose:

$$t_s p + t_w p(N^2/p) + p(N^2/p)$$

- Eigenvectors: Let K be the number of iteration for diagonalization of matrix

$$KT(N, p, t_s, t_w)N^3$$