Parallel Bucket-Brigade Communication Interface for Scientific Applications

Daiki Fukudome and Toshiya Takami
Kyushu University
Bucket-Brigade Communications

One dimensional neighbor collective communication with calculations

\( k \)-th resource in time direction:

- Receives \( x_k \) from the adjacent resource
- Calculates \( x_{k+1} = x_k + y_k \)
- Transfers \( x_{k+1} \) to the next

```c
BBC(Data *recvdata, Data *owndata, int count,
   int root, int dest, MPI_Comm comm) {
    MPI_Recv(Xk-1);
    Xk = Yk+Xk-1;
    MPI_Send(Xk);
}
```
New Implementation of Parareal-in-Time Method


We propose a new configuration of the parareal method.

The standard Parareal-in-Time method
\[ x_{k+1}^{r+1} = g(x_{k+1}^{r+1}) + f(x_k^r) - g(x_k^r) \]

Our method
\[ x_{k+1}^{r+1} = x_k^{r+1} + f(x_k^r) - x_k^r \]
Application to Time Evolution of Burgers Equation

Convergence to the results by the original solver:
- 16 time steps
- \( dt = 0.0004 \) (s)
- Problem size: 100x100

Speedup: compared to the spatial parallel
- Time parallel: 16
- Spatial parallel: 2-32

Burgers equation
\[
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \frac{\partial u^2}{\partial x^2}
\]
is the simplified version of Navier-Stokes equation.

By the original solver \( f(x) \) of the Burgers equation.

for (iteration of \( r \)) {
    BBC\( (x_{k-1}, x_k) \);
    Do \( x_k = f(x_k) - x_k \);
}
Summary & Discussions

We defined the bucket-brigade communication pattern.
   – Performance was measured and its timeline was shown.

We introduced a new configuration of the parareal-in-time method without any coarse solver.
   – Convergence and speedup were demonstrated in time evolutions of 2D Burgers equation.

Discussions:
   – Other applications of Bucket-brigade communications?
   – How can we improve performance of the pattern?