

# Hybrid parallel programming and DP – hands-on examples

Simon Scheidegger  
[simon.scheidegger@gmail.com](mailto:simon.scheidegger@gmail.com)

August 3<sup>rd</sup>, 2017

Open Source Macroeconomics Laboratory – BFI/UChicago

Including adapted teaching material from books, lectures and presentations by  
B. Barney, B. Cumming, W. Gropp, G. Hager, M. Martinasso, R. Rabenseifner, O. Schenk, G. Wellein

# Outline

## **I. Discrete state dynamic programming**

- OMP & MPI
- MPI groups

# 1. DSDP – the model

$$V_{new}(k, \Theta) = \max_c (u(c) + \beta \mathbb{E}\{V_{old}(k_{next}, \Theta_{next})\})$$

$$\text{s.t. } k_{next} = f(k, \Theta_{next}) - c$$

$$\Theta_{next} = g(\Theta)$$

## States of the model:

- $k$  : today's capital stock → **There are many independent  $k$ 's**
- $\Theta$  : today's productivity state → **The  $\Theta$ 's are independent**

## Choices of the model:

- $k_{next}$

→  $k$ ,  $k_{next}$ ,  $\Theta$  and  $\Theta_{next}$  are limited to a finite number of values

# solver.cpp – the critical loops

```
for (int itheta=0; itheta<ntheta; itheta++) {
```

2). \*split MPI communicator

```
/*  
Given the theta state, we now determine the new values and optimal policies corresponding to each  
capital state.  
*/
```

```
for (int ik=0; ik<nk; ik++) {
```

1). distribute k's via OpenMP & MPI

loops to worry about

```
// Compute the consumption quantities implied by each policy choice  
c=f(kgrid(ik), thetagrid(itheta))-kgrid;
```

```
// Compute the list of values implied by each policy choice  
temp=util(c) + beta*ValOld*p(thetagrid(itheta));
```

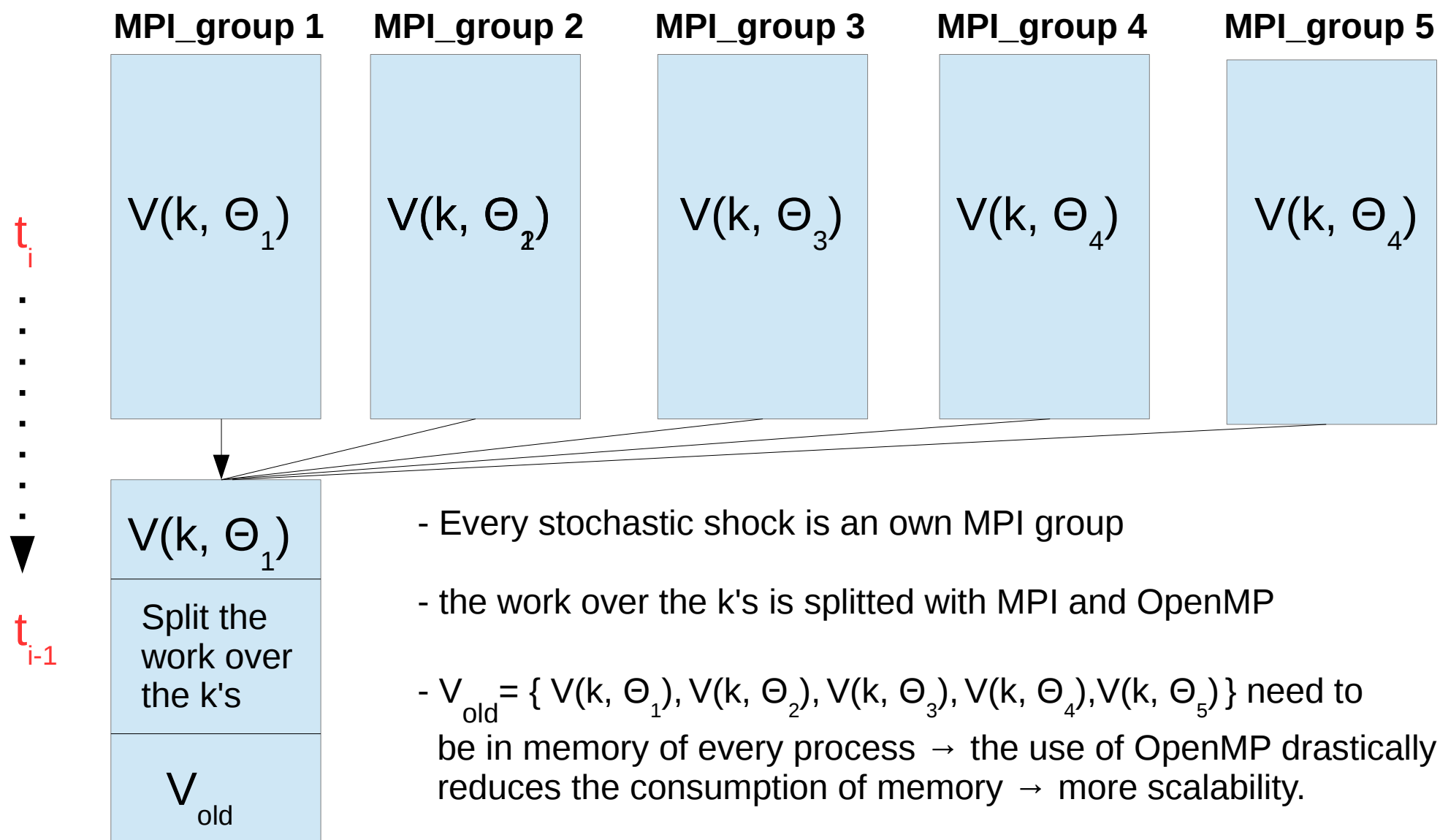
```
/* Take the max of temp and store its location.  
The max is the new value corresponding to (ik, itheta).  
The location corresponds to the index of the optimal policy choice in kgrid.  
*/
```

```
ValNew(ik, itheta)=temp.maxCoeff(&maxIndex);
```

```
Policy(ik, itheta)=kgrid(maxIndex);
```

```
    }  
}
```

# The parallelization scheme



Let's look at the code and run it

cd /project2/osmlab/DP\_MultComms

>vi main.cpp

>vi solver.cpp

Compile

>make (notice the compilation flags for OpenMP)

>run on Midway (submit\_hybrid\_midway.sh)