Parallel Dynamic Programming[1]

Kenneth L. Judd, Hoover Institution

January 20, 2014

[1] Joint work with Yongyang Cai, Greg Thain and Stephen Wright.

Parallel DP Algorithm

- DP Problems:

$$V_t(x,\theta) = \max_{a\in\mathcal{D}(x,\theta,t)} \quad u_t(x,a) + \beta\mathbb{E}\{V_{t+1}(x^+,\theta^+)\},$$

- Parallelization in Maximization step in NDP: Compute

$$v_{i,j} = \max_{a\in\mathcal{D}(x^i,\theta^j,t)} \; u_t(x^i,\theta^j,a) + \beta\mathbb{E}\{\hat{V}(x^+,\theta^+;\mathbf{b}^{t+1})\}, \qquad (1)$$

  for each $x^i \in \mathbb{X}_t$ and $\theta^j \in \Theta$, $1 \le i \le N_t$, $1 \le j \le D$
- Condor Master-Worker system: distributed parallelization, two entities: Master processor, a cluster of Worker processors.

Type-I Parallel DP Algorithm for Master

Initialization. Given a finite set of $\theta \in \Theta = \{\theta^j : 1 \leq j \leq D\} \subset \mathbb{R}^d$. Set $\mathbf{b}^T$ as the parameters of the terminal value function. For $t = T-1, T-2, \ldots, 0$, iterate through steps 1 and 2.

Step 1. Separate the maximization step into D tasks, one task per $\theta \in \Theta$. Each task contains parameters $\mathbf{b}^{t+1}$, stage number t and the corresponding task identity for some θ^j. Then send these tasks to the workers.

Step 2. Wait until all tasks are done by the workers. Then collect parameters $\mathbf{b}_j^t$ from the workers, for all $1 \leq j \leq D$, and let $\mathbf{b}^t = \left\{\mathbf{b}_j^t : 1 \leq j \leq D\right\}$.

Type-I Parallel DP Algorithm for Worker

Step 1. Get parameters $\mathbf{b}^{t+1}$, stage number t and the corresponding task identity for one $\theta^j \in \Theta$ from the master, and then choose the approximation grid, $\mathbb{X}_t = \{x_t^i : 1 \leq i \leq N_t\} \subset \mathbb{R}^n$.

Step 2. For this given θ^j, compute

$$v_{i,j} = \max_{a \in \mathcal{D}(x^i, \theta^j, t)} u(x^i, \theta^j, a) + \beta \mathbb{E}\{\hat{V}(x^+, \theta^+; \mathbf{b}^{t+1})\},$$

for each $x^i \in \mathbb{X}_t$, $1 \leq i \leq N_t$, where the next-stage discrete state $\theta^+ \in \Theta$ is random with probability mass function $\mathbb{P}(\theta^+ = \theta^{j'} \mid \theta^j) = p_{j,j'}$ for each $\theta_{j'} \in \Theta$, and x^+ is the next-stage state transition from x^i and may be also random.

Step 3. Using an appropriate approximation method, compute $\mathbf{b}_j^t$ such that $\hat{V}(x, \theta^j; \mathbf{b}_j^t)$ approximates $\{(x^i, v_{i,j})$: $1 \leq i \leq N_t\}$, i.e., $v_{i,j} \approx \hat{V}(x^i, \theta^j; \mathbf{b}_j^t)$ for all $x^i \in \mathbb{X}_t$.

Step 4. Send $\mathbf{b}_j^t$ and the corresponding task identity for θ^j to the master.

Type-II Parallel DP Algorithm for Master

Step 1. Separate $\mathbb{X}_t$ into M disjoint subsets with almost equal sizes: $\mathbb{X}_{t,1}, \ldots, \mathbb{X}_{t,M}$, and separate the maximization step into $M \times D$ tasks, one task per $(\mathbb{X}_{t,m}, \theta^j)$ with $\theta^j \in \Theta$, for $m = 1, \ldots, M$ and $j = 1, \ldots, D$. Each task contains the parameters $\mathbf{b}^{t+1}$, the stage number t and the corresponding task identity for $(\mathbb{X}_{t,m}, \theta^j)$. Then send these tasks to the workers.

Step 2. Wait until all tasks are done by the workers. Then collect all $v_{i,j}$ from the workers, for $1 \leq i \leq N_t$, $1 \leq j \leq D$.

Step 3. Using an appropriate approximation method, for each $\theta^j \in \Theta$, compute $\mathbf{b}_j^t$ such that $\hat{V}(x, \theta^j; \mathbf{b}_j^t)$ approximates $\{(x^i, v_{i,j})\colon 1 \leq i \leq N_t\}$, i.e., $v_{i,j} \approx \hat{V}(x^i, \theta^j; \mathbf{b}_j^t)$ for all $x^i \in \mathbb{X}_t$. Let $\mathbf{b}^t = \left\{\mathbf{b}_j^t : 1 \leq j \leq D\right\}$.

Type-II Parallel DP Algorithm for Worker

Step 1. Get the parameters $\mathbf{b}^{t+1}$, stage number t and the corresponding task identity for one $(\mathbb{X}_{t,m}, \theta^j)$ with $\theta^j \in \Theta$ from the master.

Step 2. For this given θ^j, compute

$$v_{i,j} = \max_{a \in \mathcal{D}(x^i, \theta^j, t)} u(x^i, \theta^j, a) + \beta \mathbb{E}\{\hat{V}(x^+, \theta^+; \mathbf{b}^{t+1})\},$$

for all $x^i \in \mathbb{X}_{t,m}$, where the next-stage discrete state $\theta^+ \in \Theta$ is random with probability mass function $\mathbb{P}(\theta^+ = \theta^{j'} \mid \theta^j) = p_{j,j'}$ for each $\theta^{j'} \in \Theta$, and x^+ is the next-stage state transition from x^i and may be also random.

Step 3. Send $v_{i,j}$ for these given $x^i \in \mathbb{X}_{t,m}$ and θ^j, to the master process.

Parallelization in Optimal Growth Problems

- Problem size: 4D continuous state k, 4D discrete state θ with $7^4 = 2401$ values
- Performance of Type-I Parallel DP:

Table: Statistics of Parallel DP under HTCondor-MW for the growth problem

Wall clock time for all 3 VFIs	8.28 hours
Wall clock time for 1st VFI	0.34 hours
Wall clock time for 2nd VFI	3.92 hours
Wall clock time for 3rd VFI	4.01 hours
Total time workers were up (alive)	16.9 days
Total cpu time used by all workers	16.5 days
Number of (different) workers	50
Average Number Present Workers	49
Overall Parallel Performance	98.6%

Parallel efficiency for various numbers of worker processors

# Worker processors	Parallel efficiency	Average task wall clock time (second)	Total wall clock time (hour)
50	98.6%	199	8.28
100	97%	185	3.89
200	91.8%	186	2.26

Parallelization in Dynamic Portfolio Problems

- Problem size: 6 stocks plus 1 bond, transaction cost, stochastic interest rate with 5 values, number of task = 3125 (each discrete state has 625 tasks).
- Performance of Type-II Parallel DP:

Table: Statistics of Parallel DP under HTCondor-MW for the 7-asset portfolio problem with stochastic interest rate

Wall clock time for all 6 VFIs	3.6 hours
Wall clock time for 1st VFI	4.8 minutes
Wall clock time for 2nd VFI	43.4 minutes
Wall clock time for 3rd VFI	40.6 minutes
Wall clock time for 4th VFI	41.5 minutes
Wall clock time for 5th VFI	42.9 minutes
Wall clock time for 6th VFI	43.7 minutes
Total time workers were up (alive)	29.3 days
Total cpu time used by all workers	27.4 days
Number of (different) workers	200
Average Number Present Workers	194
Overall Parallel Performance	94.2%