Design of Parallel Algorithm for Kalman Filter on SW26010 Processors

Aiqiang Yang¹*, Dandan Xu²
¹College of information and intelligence, Hunan Agricultural University, Changsha, Hunan 410128, China
²College of Information Science and Engineering, Hunan University, Changsha, Hunan 410008, China

*yangaiqiang99@163.com

Abstract. Kalman filter algorithm, an effective data processing algorithm, has been widely used in space monitoring, wireless communications, tracking systems, the financial industry, and so on. On the Sunway TaihuLight platform, we present an improved Kalman filter parallel algorithm which is according to the new architecture of the SW26010 many-core processors (260 cores) and new programming mode (master and slave heterogeneous collaboration mode). Furthermore, we propose a pipelined parallel mode for the KF algorithm based on a seven-level pipeline of the SW26010 processor. The vector optimization strategy and double buffering mechanisms are provided to improve the parallel efficiency of Kalman filter parallel algorithm on SW26010 processors. The vector optimization strategy can improve data concurrency in parallel computing. In addition, the communication time can be hidden by double buffering mechanisms of SW26010 processors. The experimental results show that the performance and scalability of the parallel Kalman filter algorithm based on SW26010 are greatly improved compared with the CPU algorithm for five different data sets, and is also improved compared to the algorithm on GPU.

Keywords: Heterogeneous computing; Kalman Filter; Parallel algorithm; Many-core processors.

1. Introduction
Kalman Filter (abbreviated as KF), the most effective data algorithms in information processing, is one of the powerful tools to solve the estimation and prediction of the state-space model. It does not need to store historical data and can be used to optimize the fitting of the state-space model with computer programs [1]. At present, with the increase of state space, the order of the state transition matrix of the KF algorithm becomes larger and larger, which leads to the increase of the computational complexity of KF algorithm rising. The traditional serial algorithm is slow and can not meet the demand for data processing efficiency. Therefore, the parallel advantage of a high-performance computer is utilized to speed up the computational efficiency of a complex KF algorithm. In recent years, universities have successively carried out the construction of a smart campus. Smart classrooms have improved the efficiency of teachers’ teaching and students’ learning, electronic examination rooms have
reduced the labor cost required for invigilation, video structured analysis system has effectively main-
tained the campus security, and so on.

The KF algorithm has five steps: predicted (or priori) state estimate, predicted (or priori) estimate
covariance, calculated optimal Kalman gain, updated (or posteriori) state estimate, and updated (or
posteriori) estimate covariance. The whole calculation process is recursive, and the current step must
use the results of the previous step. This kind of operation process with data dependence is not suitable
for parallel computing. However, there are a lot of matrix calculations in the iteration for the large
scale of data [2]. Optimization. So the performance of KF algorithm can be significantly improved by
the parallel matrix computations.

The Sunway TaihuLight supercomputer was once the world's number one supercomputer. The core
computing components of the Sunway TaihuLight supercomputer are the SW26010 processors, which
provide 3.06 TFlops peak performance[3]. Compared to GPU (Graphics processing unit, GPU) and
MIC (Many Integrated Core) chips, the memory bandwidth on-chip and buffer size of SW26010 are
relatively limited. KF parallel algorithm based on the SW26010 processors will have certain ad-
vantages over CPU and GPU.

1) We implement a parallel algorithm of KF on different parallel architectures, such as multi-
core CPU using OpenMP, GPU with enabled-CUDA (Compute Unified Device Architecture),
and the SW26010 processors.

2) A parallel mode for KF algorithm is provided by pipelining technology based on SW26010
processor.

According to experimental results for 5 test cases, the performance of our algorithm on SW26010
processors is greatly improved.

The paper is organized listed hereafter.
In Section 2, Review related works on the KF algorithms and design of parallel algorithms.
In Section 3, Give an introduction to the SW26010 processors.
In Section 4, Present an introduction to the KF algorithms.
In Section 5, Describe the parallel implementation of our algorithm on SW26010 processors.
In Section 6, Demonstrate the test results of performance in our experiments.
In Section 7, Conclude the paper.

2. Related Research
There are many improved variants for KF, such as Extended Kalman Filter (EKF) [4] and Unscented
Kalman Filter (UKF) [5] since KF was proposed in the 1960s [6], which were utilized in a nonlinear
system. The KF can work well in the condition that priori statistics of the measurement models and
stochastic errors are assumed to be available in both dynamic processes, but it has particular difficulty
in practical applications, which include the measurement noise covariance matrix R [7]. Also, the
adaptive Kalman filter (AKF) [8] used some adaptive mechanisms to solve this problem. According to
the filtering performance, the change of process or measurement noises can adaptively be adjusted by
estimating its noise statistics parameters for AKF. The different methods of the adaptive filter were
classified into four categories [9]: maximum likelihood, correlation, Bayesian, and covariance match-
ing. An attitude estimation method was introduced for low-sample-rate global navigation satellite sys-
tem (GNSS) velocity and position measurements using parallel AKF[10], which could estimate the
AVP error of the integration method to address the issues of the GNSS low sampled rate and abnormal
measurements.

For motion prediction, Azuma and Bishop [11] improved the registration in two areas, and one is a
new system that demonstrated accurate static registration across a wide variety of positions and view-
ing angles. The other is dynamic errors which were reduced by predicting future head locations when
the user moved its head. For navigation and global positioning systems, Amor and Chebbi [12] intro-
duced that the tone jamming signal was modeled by an autoregressive (AR) process and whitened by
estimating its AR coefficients jointly with the time delay. With the improvement of species concentra-
tions in chemical transport models, three-dimensional variational(3 DVar) DA and model output sta-
Statistics based on one-dimensional KF were implemented separately and simultaneously to investigate their performance for the model forecast [13]. In the image treatment, Erturk [14] presented a real-time stabilization system that used KF to retain smooth gross movements by removing short-term image fluctuations. A magnetic sensing-based parallel orientation Kalman filter (KF) was proposed [15], which had advantages over the KF with state-augmentation in the calculation efficiency.

3. An Introduction to SW26010 Processors
SW26010 many-core processors were developed by Shanghai High Performance Integrated Circuit Design Center through independent technology, which adopted a heterogeneous many-core architecture composed of on-chip computing array clusters and distributed shared memory [16]. The architecture of the SW26010 processor is shown in Fig.1.

![Fig. 1 Teaching system adaptive evolution system logic architecture](image)

The SW26010 many-core processor consists of four heterogeneous groups (computing cores). The heterogeneous group consists of a master core, 64 slave cores, heterogeneous group interfaces, and memory controllers. The chip contains 260 computing cores in total. The master and slave core support a vectorization operation of 256 bits. The SW26010 processor also integrates a system interface bus, which connects the standard PCIe interface to realize direct connection and interconnection on the chip to achieve system management, maintenance, and testing. Storage sharing and communication are implemented between four heterogeneous groups by the transmission network within the group. The computing power of the SW26010 processor is shown in Tab. 1.

| Parameters                               | Master cores | Slave cores |
|-----------------------------------------|--------------|-------------|
| peak performance of double-precision floating point | 24GFLOPS     | 16.5GIPS    |
| peak performance of single-precision fixed-point | 12GFLOPS *64 | 13.5GIPS *64 |
| frequency                               | 1.45 GHz     | 1.45 GHz    |
| memory                                  | 32KB L1 cache, 256KB L2 cache | 64KB * 64 |

**Tab. 1** Parameters of SW26010 processor

4. Kalman Filter Algorithm
4.1. Mathematical Model Of Kalman Filter Algorithm

The system state and its error covariance can be estimated by KF for the uncertainties in transition and observation models with hidden states for linear dynamic systems. The equations of the KF are divided into two groups. The first group is time update equations, and the second group is measurement update equations. The current state estimate executes ahead in time on the first step of KF, and the measurement equations update adjustments of the projected estimate for an actual measurement on the following step of KF. Furthermore, the time update equations can be used as corrector equations to predictor update equations. Indeed the final a predictor-corrector is resembled in the estimation algorithm for solving numerical problems as shown in Fig. 2.

KF can estimate a process with feedback control, which obtains feedback in the form of measurements by the estimate of the process state synchronously. The priori estimates of the next time step can be obtained by projecting forward the current state and error covariance estimates using the time update equations. The improved estimates can be obtained by the measurement equations [6].

The equations of the time and measurement updates are presented by Eq. (1) to (5):

\[
\begin{align*}
\dot{X}_k &= AX_{k-1} + BU_{k-1} \quad (1) \\
\dot{P}_k &= AP_{k-1}A^T + Q \quad (2) \\
K_k &= \dot{P}_kH^T(H\dot{P}_kH^T + R)^{-1} \quad (3) \\
X_k &= \dot{X}_k + K_k(Z_k - H\dot{X}_k) \quad (4) \\
P_k &= (I - K_k)\dot{P}_k \quad (5)
\end{align*}
\]

The state and covariance estimates can be obtained by the time update equations forward from time step \(k-1\) to step \(k\) by Eq. (1) and (2).

The first task is to compute the Kalman gain, \(K_k\) during the measurement update, which is calculated by Eq. (3). Secondly, the posteriori state estimate is obtained by incorporating the measurement using Eq. (4). And Eq. (4) generates a final estimate after many iterations. Finally, a posteriori error covariance can be obtained using Eq. (5) in each iteration [6].

The commonly used symbols and their descriptions are summarized in Tab. 2.

| Symbols   | Descriptions                              | Symbols   | Descriptions                              |
|-----------|-------------------------------------------|-----------|-------------------------------------------|
| \(X_{k-1}\) | the n* n priori state estimate            | \(\dot{P}_k\) | the n* n predictive estimate error covariance |
| \(\dot{X}_k\) | the n* n predictive state estimate        | \(P_k\)   | the n* n posteriori estimate error covariance |
4.2. An Example Of Kalman Filter

An example of KF will be introduced to illustrate the basic practical application in trajectory measurements in this section. The trajectory refers to the route of an object moving from a starting position to an ending position. It is represented by the direction of trajectory, the form of trajectory, and the range of motion. In the two-dimensional plane, we study the problem of prediction for object trajectories using KF. Given the parameters of the object trajectories:

\[
A = \begin{pmatrix} 1 & 0.01 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},
\]

\[
H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad R = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.
\]

For 100 measurements of the object, and the object trajectories can be predicted by KF. The estimates and measurements are shown in Fig. 3, which have good consistency.

| Symbol | Description |
|--------|-------------|
| $X_k$  | the $n^*$ n posteriori state estimate |
| $A$    | the $n^*$ n state transition model |
| $B$    | the $n^*$ n control-input model |
| $U$    | the $n^*$ n control matrix |
| $P_{k-1}$ | the $n^*$ n priori estimate error covariance |
| $Z$    | the $n^*$ n observation model |
| $Q$    | the $n^*$ n covariance of process noise |
| $R$    | the $n^*$ n covariance of observation noise |
| $H$    | the $n^*$ n observation model |
| $I$    | the $n^*$ n unit matrix |

**Tab. 2 Frequently Used Symbols and Their Descriptions**

5. Design Of Kf Parallel Algorithm Based On Sw26010

5.1. Master-Slave Core Heterogeneous Parallel Mode

The heterogeneous system based on master-slave cores provides flexible heterogeneous programming models. There are mainly four kinds.
1. Master-slave synchronous parallel mode: The master core mainly processes computing and communication tasks, which can not handle by slave cores. Nevertheless, the master core is in a blocking state when the slave cores are computing.

2. Master-slave asynchronous parallel mode: In order to improve the parallel efficiency of master-slave cooperation, the master core completes other computing, communication, or I/O operations while the slave core accelerates the computation.

3. Master-slave collaborative parallel mode: The master core and the slave core perform parallel computing as peers, and distribute the load according to their respective computing power to perform the calculation of the tasks together.

4. Master-slave dynamic parallel mode: The tasks are allocated through the master core. Selecting the appropriate heterogeneous programming model according to the program characteristics can achieve a better acceleration effect.

For the calculation process of KF algorithm, we choose the master-slave cooperative parallel model as the programming model, which is shown in Fig. 4. In the parallel algorithm design of KF, it is necessary to write the master codes and the slave codes. The master codes implement the data reading, calling the slave kernel function, data communication, and partial calculation, while the slave codes implement the task division, data calculation, and result access.

5.2. The Basic Calculation Process Of KF
The basic process of KF calculation is shown in Algorithm 1, in which A and B are parameters matrices for the multi-model system, H is the measurement parameter matrix for the multi-measurement system, R is the noise covariance of measurement, Q is the covariance of processing noise, $X_{k-1}$ is prior state estimation, $P_{k-1}$ represents prior estimation error covariance, $Z_k$ represents measurement value and K represents iteration steps. X is a state matrix in Algorithm 1. Because a lot of matrix computations are involved in the calculation process, the computational steps in the iterative process of the algorithm can be parallelized.

![Algorithm 1](image)

Algorithm 1 Basic algorithm of KF:

**Require:** Basic parameters of KF, $A$, $B$, $H$, $Q$ and $R$, $X_{k-1}$, $P_{k-1}$, $k$, $Z_k$.

**Ensure:** Optimal estimation of state $K$, $X_k$.

1. while $k \leq n$ do
2. //Predict state
3. $X_k \leftarrow AX_{k-1} + BU_{k-1}$;
4. //Predict estimated covariance
5. $P_k \leftarrow AP_k^{-1}A^T + Q$;
6. //Calculate the optimal Kalman gain
7. $K_k \leftarrow P_kH^T(HP_kH^T + R)$;
8. //Update state estimation
9. $X_k \leftarrow X_k + K_k(Z_k - HX_k)$;
10. //Update covariance estimation
11. $P_k \leftarrow (T - K_k)P_k$;
12. end while
13. return $X_k$.

![Fig. 5](image)

Fig. 5 The pipeline of KF algorithm on SW26010 processor

5.3. Design Of Heterogeneous Parallel Algorithm Of KF
KF algorithm can be divided into two groups of the calculation process, one is the time update equation, the other is the measurement update equation. The state estimation of KF algorithm needs to pro-
cess (prediction and update) 18 matrices. So as to improve the computational efficiency, we use the SW26010 processor with four groups of cores to accelerate matrix operations. The parallel KF algorithm can integrate the original 18 steps into 10 steps, as shown in Algorithm 2.

Fig. 5 shows the parallel processing of KF algorithm on the SW26010 processor in the form of a pipeline. Each group of cores in the SW26010 performs different matrix operations, and then after eliminating correlation, each group of cores divides its tasks into a master core and 64 slave cores. And the same operation is performed on each slave core to maximize the benefits of the slave cores and improve the parallel efficiency of the master-slave structure. Matrix multiplications, matrix transpositions, matrix inversions, matrix additions, and matrix subtractions are divided into slave cores to execute. Parallelization is carried out according to the rule that the same cores perform the same operations. Some tasks that cannot be parallel are assigned to the main core for execution to maximize computational efficiency.

6. Experimental Evaluation

6.1. Experiment Settings

Two computers are used in the experiment. The first computer is equipped with Intel E5-2640 CPU running at 2.30 GHz and a Tesla K20m GPU of NVIDIA. The GPU has 2496 CUDA cores and possessing 5 GB global memory. The test machine runs at Redhat 6.4. The second computer is a fully connected cluster with four SW26010 processors in Wuxi State Key Laboratory of Mathematical Engineering and Advanced Computing. The hardware parameters of the computers are shown in Table 4 and 5. We select five data sets (100 * 100, 200 * 200, 300 * 300, 400 * 400, 500 * 500) to evaluate the performance of KF algorithm. These tested sets are randomly produced from trajectory measurements.
In the experiment, the KF parallel algorithms are implemented on CPU, GPU, and SW26010 processors respectively. The test results of CPU, GPU, and SW26010 are shown in Fig. 6, where the numbers under "CPU" and "SW26010" represent the numbers of threads of CPU and SW26010 processors respectively. From Fig. 6, we can find that the single SW26010 processor runs faster than CPU but slower than GPU. Because the core frequency of SW26010 cores is low and the number of GPU threads is much larger than that of SW26010 slave cores, the performance of a single SW26010 node is worse than that of a single GPU, but the 4 SW26010 processors can make up a fully connected cluster, which shows that the complex numerical simulation application program has better scalability in SW26010 architecture.

In addition, for a single SW26010 processor, the running time of the program decreases with the increase of the number of slave cores, but the speed of reduction becomes slower and slower. This is due to the limitation of the computing capacity of slave cores on a single processor (determined by the size of memory). When the number of cores reaches 16, the computing capacity of slave cores can be saturated if the size of the matrix is large enough. However, when the size of transmission data between master core and slave core does not change greatly, it has little effect on program performance.
Fig. 6 Performance comparison of Kalman filter algorithm on multi-core CPU, GPU, and SW processors

7. Concluding Remarks
This paper mainly introduces the design and implementation of KF parallel algorithm on the platform of SW26010 processors. Experiments were carried out on multi-core CPU, GPU, and 4-node SW26010 processors cluster. The performance differences of parallel algorithms were compared and the factors affecting the performance were analyzed. From the experiment, we can find that SW26010 processors have good scalability and provide an ideal computing platform for high-performance computing. It can be used to develop application systems for large data sets and complex computing.

References
[1] Welch Greg, Bishop Gary 2006 J. An Introduction to the Kalman Filter. Proc Siggraph Course. 8.
[2] Xu D, Xiao Z, Li, D and Wu F 2016 J.Optimization of parallel algorithm for Kalman filter on CPU-GPU heterogeneous system. 12th International Conference on Natural Computation and 13th Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). IEEE.
[3] Jack, Dongarra 2016 J. Sunway taihu light supercomputer makes its appearance, National Science Review 3 pp265–266
[4] B Sadeghi, B Moshiri 2007 J.Second-order ekf and unscented kalman filter fusion for tracking maneuvering targets in IEEE International Conference on Information Reuse and Integration
[5] Chang L, Hu B, Li A, and Qin F, 2013 J.Transformed unscented kalman filter, IEEE Transactions on Automatic Control 58 pp252–257
[6] R E Kalman 1960 J.A new approach to linear filtering and prediction problems Journal of Basic Engineering Transactions 82 pp35–45
[7] Whittle J, Schumann J. 2004 J.Automating the implementation of kalman filter algorithms ACM Transactions on Mathematical Software 30 pp434-453
[8] K D Sebesta, N Boizot 2014 J.A real-time adaptive high-gain ekf, applied to a quadcopter inertial navigation system IEEE Transactions on Industrial Electronics 61 pp495–503
[9] R K Mehra 1972 J.Approaches to adaptive filtering, Automatic Control IEEE Transactions on 17 pp693–698
[10] Xiong L., Xia X, Lu Y, Liu W, Gao L, Song S and Yu Z 2020 J.Imubased automated vehicle body sideslip angle and attitude estimation aided by gnss using parallel adaptive kalman filters IEEE Transactions on Vehicular Technology pp1–10
[11] R Azuma, G Bishop 1994 J.Improving static and dynamic registration in a see-through hmd in Conference on Computer Graphics and Interactive Techniques
[12] N Amor, S Chebbi 2017 J.Performance comparison of particle swarm optimization and extended kalman filter methods for tracking in non-linear dynamic systems in International Conference on Control
[13] Chaqun M, Wang T, Zang Z and. Zhijin L 2018 J.Comparisons of three-dimensional variational data assimilation and model output statistics in improving atmospheric chemistry forecasts Advances in Atmospheric Sciences 35 pp813–825
[14] S Erturk 2002 J.Real-time digital image stabilization using kalman filters Real-Time Imaging 8 pp317–328
[15] J K Lee 2019 J.A parallel attitude-heading kalman filter without stateaugmentation of model-based disturbance components IEEE Transactions on Instrumentation and Measurement 68 pp2668–2670