DOI 10.34185/1562-9945-2-139-2022-11
UDC 004.032.24

D.M. Moroz

AGGREGATION ARRANGEMENT FEATURES OF NETWORK INTERFACE CHANNELS IN MULTIPROCESSOR COMPUTING SYSTEMS

Abstract. The paper identifies ways to increase the multiprocessor computing system performance by reorganizing the architecture of its network interface. It is shown that the computational parallelization performance significantly depends on many factors, the most essential of which is the data transfer between the boundary nodes of a multiprocessor system, which is the algorithm's slowest part and can significantly reduce the effect of increasing the number of processors. Hence, it was established that improving the multiprocessor systems performance by reorganizing the network interface structure is relevant interesting nowadays, and its study is at the active development stage.

The research aims at the aggregation arrangement of network interface channels in multiprocessor computing systems. It is shown that the main channel aggregation mode advantage is that the data interchange speed is significantly increased, as well as the reliability of the cluster system.

Keywords: aggregation, network interface, multiprocessor system, nodes, performance, parallelization.

Problem statement. The parallel computer systems application is one of the strategic directions of information technology development. That can be explained by the constant increase in the number of application tasks for the solution of which current computing tools lack capabilities [1 – 5]. Such tasks include, for instance, numerical modeling of hydrodynamics and metallurgical thermophysics processes, pattern recognition, optimization computations with a huge number of parameters, and others. It is obvious that highly efficient clusters provide a successful way to perform relevant tasks of a wide class.

Special interest in multiprocessor parallel computing systems (clusters) design is due to the publicly available components and technologies [6, 7]. Significant competition between manufacturers of computing equipment leads to the fact that prices for its components are declining quite dynamically. The same refers to the release of new computer models. Given the diversity and scale of production of current information and computing devices, it is safe to say that standard components can build powerful computer systems in a very short time and with the entire possible

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

needs of different users. Depending on the tasks type and budget for a project, there are many different options for configuring the operating system.

Analysis of multiprocessor systems design approaches has shown that computer manufacturers recently offer devices based on blade technologies [8 - 10]. Under such circumstances, designing a multiprocessor system based on blade servers, get a ready-made chassis equipped with the necessary controls and network interface.

Each blade of the modular system operates under the control of a standard operating system. The power and composition of the nodes may be different within one system. The MPI standard with the open base implementation mpich-1 was chosen as a software tool for parallel computing.

It is obvious that a new qualitative stage in the development of multiprocessor computing systems is in the field of new modern network technologies application. That can be explained by the significant differences between the cluster computer network and the workstation network. Although a cluster design requires conventional network cards and switches in the workstation network arrangement, the cluster computing system has one fundamental feature. Thus, the computer system network is not designed to connect computers but to connect computing processes. Then, the higher the bandwidth of the system's computing network, the faster the user-defined parallel tasks are performed. Therefore, the technical characteristics of the computer network are of paramount importance when it comes to multiprocessor computer systems. Considering the above, it was chosen to apply InfiniBand network interface technology.

Therefore, the InfiniBand standard arranges the data interchange between nodes of the multiprocessor system. In addition, the developed system has the following basic features: network loading of processors, VLAN mode support, redundancy mechanism of module key components, specially designed mode of data interchange between system nodes in the network of InfiniBand switches.

However, the computational parallelization performance depends on many factors, the most important of which is the data transfer between the nodes of the multiprocessor system, which is the slowest part of the algorithm and can significantly reduce the effect of increasing the number of processors. That circumstance suggests that one of the main ways to increase the efficiency of multiprocessor systems is to aggregate the network interface channels in the data interchange network between the slave nodes of the system [11, 12]. From such positions in the given research, ways of increase of efficiency of multiprocessor computing systems are estimated.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Thus, it is established that improving the multiprocessor systems performance by reorganizing the network interface structure today is relevant, interesting and its study is at the active development stage.

Analysis of recent research and publications. Analysis of the network interface operation modes of multiprocessor systems allows formulating the following problem: how the architectural design features of the network interface of multiprocessor cluster systems can increase their efficiency and speed.

That problem can be solved as follows: the "linking channels" procedure or channel bonding technology is recommended to increase the cluster bandwidth, which allows combining several network adapters into a single-speed channel [13]. The data interchange between the cluster's computing nodes is run in a separate network that operates based on the mentioned channel bonding technology [14]. The proposed approach should increase the data interchange speed between the slave nodes of the multiprocessor system and reduce the load on the channel that connects the cluster nodes.

The channel aggregation problem in modular multiprocessor cluster systems is currently not solved properly. In addition, there are critically few studies examining the impact of cluster system network architecture on computational parallelization performance.

The developed technology of linking the network interface channels of a multiprocessor cluster system allows connecting the cluster nodes into a network so that each of them is connected to the switch with more than one channel. The described technology is similar to the trunking mode when connecting switches, which increases the data rate between two or more switches. The application of the channel binding procedure allows achieving uniform load distribution (data reception/transmission) between them in a multiprocessor system and increases the speed of data interchange between its nodes.

The research purpose. The paper identifies ways to increase the multiprocessor computing system performance by reorganizing the architecture of its network interface. The proposed approach allowed increasing the computations parallelization performance and significantly reduced the time of their implementation. Those results were achieved by reducing the duration of data interchange between neighboring computing nodes of the multiprocessor system. Thus, the research aims to organize the network interface channels aggregation in multiprocessor computer systems.

The main advantage of the channel aggregation mode is that the speed of data interchange is significantly increased, and the cluster system reliability. Finally, it is provided that in case an adapter fails, the traffic is sent to the next working adapter without service interruption. When the adapter starts working again, data transfer is resumed.

Main research material presentation

Arrangement and configuration of the multi-channel network interface of the multiprocessor system. Note some important arrangement features of the network interface according to channel bonding technology in the application to the first blade of a multiprocessor system. To do this, consider the principle of connecting several adapters in parallel. Suppose there are four adapters: ib1, ib2, ib3, and ib4. They must be combined into a pseudo-adapter bond0. The system recognizes those aggregate adapters as one. They are all configured for a single MAC address, so remote servers treat them as single devices. The pseudo-adapter bond0 can be configured for a single IP address. Then any program refers to it as a normal adapter, which is twice as fast. Network interface aggregation protocols specify which ports are used for outbound traffic or which specific port receives inbound traffic. The interface state is used to check the link; it can be active or not.

Given the described features of channel linking, we define the requirements to be met by the hardware and software of a multiprocessor cluster system. Channel bonding technology implementation requires at least two physical subnets. Nevertheless, the combined channel can be built based on three or more network cards. To bind those maps into a single channel (one virtual card), it is necessary to compile the kernel of the OS with support for channel bonding mode or to load the kernel module into the OS bonding.o.

To configure the merged channel, one must use the standard ifconfig command or the optional ifenslave command. That is because ifenslave allows copying the settings of the first interface to the rest of the additional interfaces.

These studies implement the formation mode of two subnets with a single virtual channel. Shutting down network interfaces in a multiprocessor system is done by a command /etc/rc.d/init.d/network stop.

Then proceed to the actual configuration of the connected (combined) channel. The first step is to modify the /etc/modules.conf file by adding the following line:

```
alias bond0 bonding
```

In order not to reboot the system, the modprobe bonding module is activated. That tells the system that one needs to load the bonding.o module, which is identified by the bond0 command.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Then go to the directory /etc/sysconfig/network-scripts and rename the description file of the first interface ifcfg-ib1 to ifcfg-bond0 as follows:

```
cp ifcfg-ib1 ifcfg-bond0
```

The generated ifcfg-bond0 file needs to be edited so that it looks like this:

```
DEVICE=bond0IPADDR=192.168.1.1NETMASK=255.255.0.0NETWORK=
192.168.0.0BROADCAST=192.168.2.255ONBOOT=yesBOOTPROTO=
noneUSERCTL=no
```

In this case, the user must specify own data: *IP* address, mask, network address. In addition, one needs to provide information used to link the channels of the proposed multiprocessor cluster system. It should be noted that no lines were deleted in the provided file, with necessary changes. Therefore, the virtual network interface description file was edited. The next step is to create the description files of the real physical interfaces *ib1*, *ib2*, *ib3*, and *ib4*, which indicate their affiliation to the connected channel. Table 1 shows the contents of the files *ifcfg-ib1*, *ifcfg-ib2*, *ifcfg-ib3*, and *ifcfg-ib4*.

Table 1

Contents of network interface description files

file ifcfg – ib1	file ifcfg – ib2	file ifcfg – ib3	file ifcfg – ib4
DEVICE=ib1	DEVICE=ib2	DEVICE=ib3	DEVICE=ib4
USERCTL=no	USERCTL=no	USERCTL=no	USERCTL=no
ONBOOT=yes	ONBOOT=yes	ONBOOT=yes	ONBOOT=yes
MASTER=bond0	MASTER=bond0	MASTER=bond0	MASTER=bond0
SLAVE=yes	SLAVE=yes	SLAVE=yes	SLAVE=yes
BOOTPROTO=none	BOOTPROTO=none	BOOTPROTO=none	BOOTPROTO=none

Acoording to Table 1, the interface configuration files have the following contents:

file ifcfg – ib1:

```
DEVICE=ib1IPADDR=192.168.2.1NETMASK=255.255.0.0NETWORK=192.
168.0.0BROADCAST=192.168.2.255ONBOOT=yesBOOTPROTO=noneUSERCTL=
no
file ifcfg – ib2:
DEVICE=ib2IPADDR=192.168.2.2NETMASK=255.255.0.0NETWORK=192.
168.0.0BROADCAST=192.168.2.255ONBOOT=yesBOOTPROTO=noneUSERCTL=
no
```

file ifcfg – ib3:

```
DEVICE=ib1IPADDR=192.168.2.3NETMASK=255.255.0.0NETWORK=192.
168.0.0BROADCAST=192.168.2.255ONBOOT=yesBOOTPROTO=noneUSERCTL=
no
```

file ifcfg – ib4:

```
DEVICE=ib1IPADDR=192.168.2.4NETMASK=255.255.0.0NETWORK=192.168.0.0BROADCAST=192.168.2.255ONBOOT=yesBOOTPROTO=noneUSERCTL=no
```

Next, perform the network interface initialization stage by the command

```
/etc/rc.d/init.d/network start
```

If the system distribution does not allow master/slave notification, then to configure the network interfaces, one needs to run the connected channel interface manually by executing this sequence of commands:

/sbin/ifconfig bond0 192.168.1.1 up netmask 255.255.0.0 /sbin/ifenslave bond0 ib0 /sbin/ifenslave bond0 ib1 /sbin/ifenslave bond0 ib2 /sbin/ifenslave bond0 ib3 /sbin/ifenslave bond0 ib4

In order not to execute each of the listed commands manually each time, it is recommended to write them in some startup-script (it can be */ets/rs.d/rs.losal*) or to replace with them the part of the script responsible for initialization of the network interface */etc/rc.d/init.d/network*.

The resulting compiled file must be copied to the folder */usr/sbin*.

When, for any reason, all network drivers need to initialize bonding driver downloads, the following line should be added

```
probeall bond0  ib1 ib2 ib3 ib4 ibbonding
```

to the */etc/modules.conf* file. That instructs the system that in the case of *bond0* interface initialization, the modprobe utility must first load the drivers of all network interfaces.

Channel bonding configuration gets completed at this stage. If the network interface was initialized without errors, one can check this by the *ifconfig* command. After running it without entering parameters, the user sees the following messages on the terminal screen:

```
[root]# /sbin/ifconfig
bond0      Link encap: InfiniBand  HWaddr 00: C0 : F0 :1 F :37: B4
           inet addr :192.168.1.1 Bcast :192.168.2.255  Mask :255.255.0.0
           UP BROADCAST RUNNING MASTER MULTICAST  MTU :1500 MEtrIc :1
           RX packets :7224794 errors :0 dropped :0 overruns :0 frame :0
           TX packets :3286647 errors :1 dropped :0 overruns
```


ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

```
:1 frame :0
              collisions :0 txqueuelen :0
    ib1       Link encap: InfiniBand  HWaddr 00: C0 : F0 :1 F
:37: B4
              inet  addr  :192.168.2.1    Bcast  :192.168.2.255
Mask :255.255.0.0
              UP  BROADCAST  RUNNING  SLAVE  MULTICAST   MTU  :1500
MEtrIc :1
              RX packets :7224794 errors :0 dropped :0 overruns
:0 frame :0
              TX packets :3286647 errors :1 dropped  :0 over-
runs :1 carrier :0
              collisions :0 txqueuelen :100
              Interrupt :10 Base address :0 x1080
    Ib2       Link encap: InfiniBand  HWaddr 00: C0 : F0 :1 F
:37: B5
              inet  addr  :192.168.2.2    Bcast  :192.168.2.255
Mask :255.255.0.0
              UP  BROADCAST  RUNNING  SLAVE  MULTICAST   MTU  :1500
MEtrIc :1
              RX packets :7224794 errors :0 dropped :0 overruns
:0 frame :0
              TX packets :3286647 errors :1 dropped  :0 over-
runs :1 carrier :0
              collisions :0 txqueuelen :100
              Interrupt :9 Base address :0 x1400
    Ib3       Link encap: InfiniBand  HWaddr 00: C0 : F0 :1 F
:37: B6
              inet  addr  :192.168.2.3    Bcast  :192.168.2.255
Mask :255.255.0.0
              UP  BROADCAST  RUNNING  SLAVE  MULTICAST   MTU  :1500
MEtrIc :1
              RX packets :7224794 errors :0 dropped :0 overruns
:0 frame :0
              TX packets :3286647 errors :1 dropped  :0 over-
runs :1 carrier :0
              collisions :0 txqueuelen :100
             Interrupt :9 Base address :0 x1400
    Ib4       Link encap: InfiniBand  HWaddr 00: C0 : F0 :1 F
:37: B7
              inet  addr  :192.168.2.4    Bcast  :192.168.2.255
Mask :255.255.0.0
```

```
              UP BROADCAST RUNNING SLAVE MULTICAST  MTU :1500
MEtrIc :1
              RX packets :7224794 errors :0 dropped :0 overruns
:0 frame :0
              TX packets :3286647 errors :1 dropped  :0 over-
runs :1 carrier :0
              collisions :0 txqueuelen :100
              Interrupt :9 Base address :0 x1400
    lo        Link encap: Local Looppback
                    inet addr :127.0.0.1  Mask :255.0.0.0
    UP LOOPBACK RUNNING  MTU :1500  MEtrIc :1
    RX packets :1110 errors :0 dropped :0 overruns :0 frame :0
    TX packets :1110 errors :0 dropped :0 overruns :0 frame :0
    collisions :0 txqueuelen :0
```

According to the method described above, the network interfaces and other nodes of the multiprocessor system are initialized. One can then conclude that the linked channel was successfully configured. It is clear that the *IP* and *MAC* addresses of all network interfaces in the given version turned out to be the same. For the switch to work properly with such a channel, one needs to configure the *Link Aggr*egation mode.

Conclusions

The paper identifies ways to increase the multiprocessor computing system performance by rearranging the architecture of its network interface. Thus, the operation modes analysis of the network interface of multiprocessor systems allowed outlining ways to increase their performance and speed due to the architecture design features of their network interface. It should be borne in mind that the network of a computing system is not designed to connect computers but to connect computing processes. Then, the higher the system's computer network bandwidth, the faster the user-defined parallel tasks will be performed. Therefore, the computer network technical features are of paramount importance for multiprocessor computing systems.

The problem outlined in the paper is solved as follows: to increase the bandwidth of the cluster network, it is recommended to apply the "linking channels" procedure, or channel bonding technology, which allows combining multiple network adapters into a single-speed channel. In this case, the data interchange between the cluster's computing nodes is done in a separate network, which operates based on the mentioned channel bonding technology. The proposed approach should help increase the speed of data interchange between the slave-nodes of the multiprocessor

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

system and reduce the load on the channel that connects the cluster nodes.

Considering the above, it was decided to use *InfiniBand* network interface technology. Therefore, the data interchange between nodes of the multiprocessor system is arranged by *InfiniBand* standard. In addition, the developed system has the following basic features: network loading of processors, *VLAN* mode support, key components redundancy mechanism of the module, specially designed mode of data interchange between system nodes in the network of *InfiniBand* switches.

The main advantage of the channel aggregation mode is that the speed of data interchange is significantly increased, and the cluster system reliability. Finally, it is provided that in case of the adapter's failure, the traffic is sent to the next working adapter without service interruption. When the adapter restarts again, the data transfer resumes.

The proposed approach allowed increasing the efficiency of computations parallelization and significantly reduced their implementation time. Those results were achieved by reducing the duration of data interchange between neighboring computing nodes of the multiprocessor system. Thus, the research aims to arrange the aggregation of network interface channels in multiprocessor computing systems.

Those features of the developed system allowed improving it with the acquisition of certain differences from available systems, namely:

– first, the *InfiniBand* technology implementation has the following advantages: low latency and high speed, which allows solving problems that involve advanced computing segments;

– secondly, based on the *RDMA* principle of *InfiniBand* technology, direct data interchange runs between the *RAM* of multiprocessor system nodes that increase computing speed and provides high-speed access to the memory of its slave nodes, and data interchange in-between, unloading *CPU* system during data interchange and reducing the channel load that passes between the nodes of the computer system;

Third, the application of the adapters such as *ConnectX* creates fundamentally new opportunities for the "connectivity" of such a system with other computing environments; in particular, it allows unloading the *CPU* (by servicing *InfiniBand* traffic) as a consequence, increases entire multiprocessor system performance.

REFERENCE

1. Ivaschenko V.P., Shvachych G.G., Sobolenko O.V. The system of automated control of temperature regimes of thermal treatment of steel products. Metallurgical and Mining Industry. 2015. № 1. P. 142 –147.
2. Shlomchak G., Shvachych G., Moroz B., Fedorov E., Kozenkov D. Automated control of temperature regimes of alloyed steel products based on multiprocessors computing systems. Metalurgija. 2019. №58. P. 299 – 302.
3. Shvachych G., Mamuzic I., Ivaschenko O., Moroz B., Hulina I. Maximum parallel forms of difference scheme algorithms in applied problems of metallurgical thermal physics. Materials and Metallurgy: 14 International Symposium of Croatian Metallurgical Society SHMD. CROATIA. 2020. P. 442.
4. Shvachych G., Ivaschenko O., Busygin V., Fedorov E. Parallel computational algorithms in thermal processes in metallurgy and mining. Naukovyi visnyk NHU. Dnipro. 2018. № 4. P. 129–137.
5. Ivaschenko V., Shvachych G., Ivaschenko O., Busygin, V. Processors selection problem in the module computing system when making new technological processes. Information Technology in Metallurgy and Machine Building : materials and Technical International Conference. 2018. P. 106.
6. Shvachych G.G. Prospects of construction highly-productive computers systems on the base of standard technologies. Strategy of Quality in Industry and Education: IV International Conference. May 30 – June 6. 2008, Varna, Bulgaria, 2008. Vol. 2. P. 815 – 819.
7. Voevodin V.V., Voevodin Vl.V. Parallel computations. Sankt Peterburg, 2002. 600 p.
8. Shvachych G., Pobochii I., Kholod E., Ivaschenko O., V. Busygin Multiprocessors computing systems in the problem of global optimization. Structural transformations and problems of information economy formation: monograph. Ascona Publishing United States of America. 2018. P. 281 – 291.
9. Shvachych G., Pobochii I., Ivaschenko O., Busygin V. Research of compatibility in the multi-processing compound systems. Science Review. Poland. 2018. № 2 (9). Vol. 1. P.15–19.
10. Ivaschenko V., Shvachych G., Ivaschenko O., Busygin, V. Processors selection problem in the module computing system when making new technological processes. Information Technology in Metallurgy and Machine Building : materials and Technical International Conference, Dnipro, March 27-29. 2018 Dnipro. 2018. P. 106.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

11. Shvachich G., Sobolenko O., Ivashchenko O. Mode of aggregation of network interface channels of multiprocessor computing systems. Quality strategy in industry and education: Conference proceedings. Varna, July 5-8, 2017. Varna, Bulgaria. 2017. Vol. 2. P. 452–457.
12. Ivaschenko V., Shvachych G., Ivaschenko O., Busygin, V. Improving the efficiency of multiprocessors system through in-line interface network aggregation. System technologies. Dnipro. 2018. № 2(115). P. 84–93.
13. Shvachych G., Moroz B., Ivaschenko O., Sushko I., Pobochii I. The implementation features of the aggregation mode of network interface in the multiprocessors computing system. Computer modeling and optimization of complex systems: Proceedings of the IV international scientific-practical conference, Dnipro, November 1-2., Dnipro. 2018. P. 200–202.
14. Shvachich G., Kholod E., Tkach M. Features of using the INFINIBAND network interface in multiprocessor systems with a distributed computing domain. Information processing systems. 2016. No. 3(140), P. 175 – 181.

Received 16.02.2022.
Accepted 18.02.2022.

Особливості організації агрегації каналів мережевого інтерфейса в багатопроцесорних обчислювальних системах

Визначаються шляхи підвищення ефективності багатопроцесорної обчислювальної системи за рахунок реорганізації архітектури її мережевого інтерфейсу. Показано, що ефективність паралелізації обчислень істотно залежить від багатьох чинників, найважливіший з яких – особливості пересилання даних між граничними вузлами багатопроцесорної системи, що являє собою найповільнішу частину алгоритму виконуваних обчислень і може суттєво знизити ефект від збільшення кількості задіяних процесорів. Запропоновано режим агрегації каналів мережевого інтерфейса в багатопроцесорних обчислювальних системах. Відзначено, що основна перевага режиму агрегації каналів полягає в тому, що істотно підвищується швидкість обміну даними, а також зростають показники надійності функціонування кластерної системи.

Мороз Дмитро Максимович – аспірант, НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Dmytro Moroz – PhD student, Dnipro University of Technology, Dnipro, Ukraine.