CHiPPS-64 Status – A Report

Shyam Khobragade

Abstract

The stability, performance, system software update done and to be done and problems of CDoT High Performance Parallel Processing System (CHiPPS-64) are reported here.

1. System Stability

The CHiPPS-64 has been extensively used for a period of one week between 21st Sept. 1992 and 26th Sept. 1992 for various applications to check up its *stability* and *performance*. The benchmark applications involved 1K x 1K Complex-2D-FFT and pulsar search. The benchmark programs were such that they use all *communication paths* and all *computational processors* of the system. The system found to be reasonably *stable* by its own, right from the begining.

In the period of one week, the *stability test* was rigorously done every day 3 to 4 hours continuously and also 2 over-nights (under supervision). The system didn't crash by its own for any reasons related to the system software or hardware failures, except in few cases which were then sorted out and abolished.

On third day of stability test, it failed once after 85 iterations (one hour of real time) of the benchmark program giving an error for *ppmdcp* call, which was then reported to Shri. Periasamy. On the fourth and fifth days, day time was utilised for *performance evaluation* and whole night time again for *stability* check up.

During night time of 4th day, program went through for 350 iterations (about 5 hrs of real time) and failed once for a call *ppmprv*. (This problem was then solved next day). In the same night, program continued further for 308 iterations till morning.

During next over-night, program went through for about eleven and half hours with a single failure for a call *ppmpfr*.

These over-night stability checks were performed under the supervision of some of C-DoTian team.

2. Performance Evaluation

The *performance* of CHiPPS-64 has been tested using a couple of test programs involving 1K x 1K Complex-2D-FFT and some pulsar programs.

The FFT program for a total of million points Complex-2D data with binary input from a file and binary output into a file took 57 seconds, whereas with in-core data generation, it took 45 seconds using 64 processors. The sequential version of the same program on SUN-SPARC-2 for same data size took 132 seconds in binary I/O case and 125 seconds in in-core data generation case.

The speed up ratio of CHiPPS-64 to SPARC-2 found to be 2.3 in case of I/O and 2.77 in case of in-core data generation. The *reasons* for such a small ratio (using 64

processors) are that (a) the application size was not large enough for 64 processors, (b) the computational part (on CHiPPS-64) took just 15 seconds out of 57 seconds, rest all was communication only. Thus the *compute to communication ratio* is just a fraction (0.365) which should be a large value for higher speed performance.
According to *Amdahl's law*, if f is the inherently sequential fraction of a computation to be solved by p processors, then the speed up s is limited according to the formula

$$s \leq 1/(f + (1 - f)/p)$$

This law assumes that the *parallel processors* have same speed and same architecture as the serial-machine processor.
Using this law and the timings with binary I/O on SPARC-2 and SUN4/490, the speed up s of a *parallel* system with respect to the *serial* systems for this particular application should be as given in the Table 1.

SN	System Name	Speed Up
1.	SUN-SPARC-2 (swati)	$s \leq 8.4155$ [With binary I/O]
2.	SUN4/490 (gmrt)	$s \leq 14.45$ [With binary I/O]

Table 1 : *Speed Up of Parallel System over Serial Computers*

The speed up of CHiPPS-64 using 64 *computing processors* for this application (one million points Complex-2D-FFT) is tabulated below.

SN	System Name	Speed Up	Speed Up
		With Binary I/O	Without I/O
0.	CHiPPS-64	1.0	1.0
1.	SUN-SPARC-2 (swati)	2.31	2.77
2.	SUN4/490 (gmrt)	2.77	3.51

Table 2 : *Speed Up of CHiPPS-64 over SUN-SPARC-2 and SUN4/490*

The results (with I/O) on SPARC-2 system and CHiPPS-64 reveal that the 118 seconds *computational* part out of total 132 seconds has been *parallelized* on CHiPPS-64 using 64 processors, which took 15 seconds for computation out of total 57 seconds. The same application took total 160 seconds on SUN4/490 out of which 150 seconds job was *parallelized*.
Hence if we see *computation* point of view, then CHiPPS-64 found to be *faster* about 8 *times* than SPARC-2 and 10 *times* than SUN4/490, even for such a small time computation application.
The *efficiency* (η) of a parallel system is defined as the ratio of speed up (s) and the number of processors (p) used.

$$\eta = s/p$$

For such a small application, the *efficiency* can be improved using less number of processors.
For a highly *compute intensive* problem which takes hours together on serial computers, this *parallel system* will show significant performance in terms of *speed up* and *efficiency*, because in that case also, the *communication part* will be almost same as in this case, but the large *computation time* will be divided into number of processors used and will reduce the over-all time tremendously.
The *front-end* of CHiPPS has got basic *limitations* in terms of both — *speed* and *memory*. The system clock at Main Controller (MC) is running at 16 MHz and memory size is 8 Mbytes.
The disk I/O (read/write) for total 8Mbytes of binary data takes **22** seconds on CHiPPS, whereas **4.5** seconds on SPARC-2 and **3.2** seconds on SUN4/490. For a total of one million points complex 2D matrix, CHiPPS-64 takes **1.65** seconds for parallel transpose at MDAM level. SPARC-2 and SUN4/490 take **2.64** and **2.1** seconds respectively.
The speed of broadcast link for *ppdbdt* call found to be **240** *Kbytes/second* when 512 x 512 size complex data was broadcasted. The speed of MDAM to PE (for call *ppmpfr*) and reverse (for call *ppmprv*) links found to be **105** *Kbytes/second* and **310** *Kbytes/second* respectively, when 1K x 1K complex data was sent and received through MDAM paths. There is a *difference* in speed of *forward* and *reverse* paths, which has been noticed and reported to Peri for modifications, ie. to speed up the *ppmpfr* call.
The *comparative study* in terms of *speed* is given in *Appendix* A and B.

3. System Software Update

The full use of *local RAM* at PE level and *Cache* at MDAM level have been implemented on CHiPPS-64 too. Few more HSRs are added, namely *ppswpg, ppswtr, ppswdn, ppmpfr, ppmprv, ppmdxp* and *ppexec*. The description of these HSRs is given in *User Document*. The HSRs *mdput* and *mdget* are updated by making *start-elem* as *offset* to *Md-ary* only. The communication calls through MDAM path are optimized. The matrix transpose at MDAM - *ppmdxp* is now available. Using this call, one can do a *transpose* of a square matrix of maximum size N x N, where N is 512 on CHiPPS-16 and 1024 on CHiPPS-64. Now all *data types* for communication are available. Cross switching between MDAM and PEs is provided using which one-one and one-many connections are possible.
Few more things are suggested to update in next version of system software. These include — software reset for PEs and MDAMs, introduction of all data types in *ppswtr* – call, speed up of *mdput* and *ppmpfr* calls, speed up of broadcast mode calls *ppdbdt* and *ppdbcd*, introduction of AWAIT kind of concept in all communication calls to overlap the operations and process scheduling. One of the HSRs *ppexec* was not working properly, which was brought to the notice.
From the discussion with Shri. M. Periasamy, it was clear that for the system release, they will install the latest version of MDOS (ver. 5.2), but the PEOS version will be 5.1. He added that for any small change, they can not PROM 128 chips all the time and that is obviously justified. He assured that whatever the changes will be made in system software will be communicated to us.
The development of PARAS compiler is in progress, Peri added and will be given with

CHiPPS-64 (hopefully) when the system will be shifted to Pune.

4. Comments

The CHiPPS-64 was used rigorously for a week to check its stability and performance. The benchmark programs which involve the use of all communication paths and all computation processors were run successfully every day 3 to 4 hours. Besides this, these programs were run for consecutive two over-nights in supervision. In each of the two over-nights, test program failed once and the problems were brought to the notice of Shri. Periasamy, which were then solved. The overall system performance found to be stable during this period.

The performance analysis of the system has been reported in detail above. The front-end of the system is too slow which increases the communication time. The parallel portion of the system is all wright and will show better performance for very high compute-intensive applications which take hours together on serial computers.

The system performance depends upon various factors —

(a) how much part of sequential application is parallelized (in terms of time),

(b) how much time the total application takes (whether it is large time consuming for computations),

(c) programming style (whether the communication is properly overlapped and the data transfer is through faster links, etc.),

(d) the programming languages used (OCCAM, C, Fortran),

(e) application code optimization,

(f) optimization of compilers used.

Finally, a new list of *requirements* has been given to Peri. This list includes the *documents* to be received, things to be updated in system software in next version, some hardware problems related to *real time* clock and Cartridge tape drive and some Uni-Plus+ problems. See *Appendix* C.

Shri. Periasamy assured to fulfil all these requirements before shifting the parallel system (CHiPPS-64) to TIFR, Pune.

*** APPENDIX A ***

** TIMINGS FOR COMPLEX-2D-FFT ON DIFFERENT SYSTEMS ***

* In Core Data Generation
* No Read/Write statements
* No Comparision of results

{ 1K x 1K Complex-2D-FFT }

System Name	Timex (real,user,sys)	(User+Sys)
0. CHiPPS-64 (64 PEs)	1:04.36,11.18,34.11	(45.24 Secs)
1. swati (SPARC-2)	2:12.4,2:02.4,2.5	(124.9 Secs)
2. gmrt (Sun4/490)	2:58.3,2:36.0,1.8	(157.8 Secs)
3. rohini (SPARC-1)	7:57.2,4:41.2,16.0	(297.2 Secs)

{ 512 x 512 Complex-2D-FFT }

1. swati	29.0,28.4,0.5	(28.9 Secs)
2. gmrt	47.0,37.0,0.7	(37.7 Secs)
3. ashwini(SPARC-IPC)	53.5,51.3,1.0	(52.3 Secs)
4. rohini	1:10.0,1:05.8,1.3	(67.1 Secs)
5. tifr (MC68020)	24:54.8,20:45.7,13.2	(1258.9 sec)

{ 2K x 2K Complex-2D-FFT }

1. gmrt	18:01.7,11:32.6,17.0	(709.6 Secs)

********** General Details about Systems **********

* Node Name	System Name	Clock Freq.	Memory Size *
1. swati	SUN-SPARC-2	40 MHz	32 MBytes
2. gmrt	SUN-4/490	33 MHz	64 MBytes
3. ashwini	SPARC-IPC	25 MHz	8 MBytes
4. rohini	SUN-SPARC-1	20 MHz	8 MBytes
5. tifr	MC68020	25 MHz	16 MBytes

*** APPENDIX B ***

** TIMINGS FOR COMPLEX-2D-FFT ON DIFFERENT SYSTEMS ***
* Binary I/O (Read/Write) from files
* No Comparision of results

{ 1K x 1K Complex-2D-FFT }

System Name	Timex (real,user,sys)	(User+Sys)
0. CHiPPS-64 (64 PEs)	2:23.95,5.36,51.61	(56.97 Secs)
1. swati (SPARC-2)	2:49.2,2:06.3,5.7	(132.0 Secs)
2. gmrt (Sun4/490)	5:54.0,2:42.9,4.6	(167.5 Secs)
3. rohini (SPARC-1)	12:09.6,4:58.1,20.3	(318.4 Secs)

{ 512 x 512 Complex-2D-FFT }

1. swati	34.3,27.9,1.3	(29.2 Secs)
2. gmrt	1:18.7,35.9,1.4	(37.3 Secs)
3. ashwini(SPARC-IPC)	1:48.1,50.9,2.2	(53.1 Secs)
4. rohini	2:24.5,1:05.3,3.2	(68.5 Secs)

{ 2K x 2K Complex-2D-FFT }

1. gmrt	25:57.2,11:07.4,27.5	(694.9 Secs)

********** General Details about Systems **********

* Node Name	System Name	Clock Freq.	Memory Size *
1. swati	SUN-SPARC-2	40 MHz	32 MBytes
2. gmrt	SUN-4/490	33 MHz	64 MBytes
3. ashwini	SPARC-IPC	25 MHz	8 MBytes
4. rohini	SUN-SPARC-1	20 MHz	8 MBytes
5. tifr	MC68020	25 MHz	16 MBytes

*** APPENDIX C ***

********* Documents to get from C-DoT *********

* Pegasus Debugger - Documents
* Green Hill's Compiler - manuals
* Monitor Menu - Document
* System Diagnostics - Documents
* PPSOFT Document
* List of all LSRs and Action Codes and How to write HSRs using LSRs.
* Drivers - Documents
* List of Utilities and Application Programs
* User Document - updated version

---

* PARAS Compilers

********** Things to be done **********

* Software Reset
* Introduction of all Data Types in "ppswtr" call
* Make sure the working of "ppexec" call
* Speed up of "mdput" and "ppmpfr" calls
* Speed up of Broadcast mode - "ppdbdt", "ppdbcd"
* Introduction of AWAIT kind of Concept in all Communication paths
* Process Scheduling

---

* Real Time Clock - to be made working
* Cartridge Tape Drive - not working

********** UniPlus+ Problems **********

* lpr spooler - Not working on Network
  (Error - socket protocol : not supported)
* mt - magnetic tape manipulating program
  Not working
  (Error - mt : unknown error)

++++++++ Shyam W. Khobragade (TIFR, Pune) +++++++++