

ОТНОСНО ИЗПОЗЛВАНЕТО НА СУПЕРКЛЪСТЕРИ В АВТОМАТИЗАЦИЯТА

Георги Михалев¹, Александър Любенов²

¹Технически университет - Габрово

²Технически университет - Габрово

ON THE USE OF SUPERCLUSTERS IN AUTOMATION

Georgi Mihalev¹, Aleksandar Lubenov²

¹Technical university of Gabrovo

²Technical university of Gabrovo

Abstract

The paper draws attention to the growing need of high performance computing devices. The use of more powerful computer systems is growing rapidly due to the appearance of new multidisciplinary scientific fields such as big data, computer simulations, artificial intelligence and others. An overview of the trends in the construction of superclusters is made. The main aspects of their software have been studied. Data on performance and consumed electricity are summarized according to the size of the computer cluster. In view of modern trends in automation, guidelines are given for the place of superclusters in the control of technological objects and processes.

Keywords: Superclusters; single board computers; automation.

ВЪВЕДЕНИЕ

До сега компютърните клъстери бяха прекалено скъпа и недостъпна технология за обикновения потребител. След популяризирането на единно бордовите компютри (SBC) се появяват и първите суперклъстери. Компютърният клъстер представлява съвкупност от взаимодействащи едноплатков компютри. Разбира се реалната основа на паралелните изчисления е дадена много по-рано с развитието на полупроводниковата технология.

В основата на много проблеми може да се открие необходимостта от вграден паралелизъм, който и поставя въпроса за компютърното обезпечаване за решаването на тези проблеми. Именно на това се дължи факта, че паралелните компютри имат значително място в науката и техниката [1].

Обикновено при използването на хардуерна система с възможност за паралелни изчисления се смята, че разработването на софтуера отнема много време и коства ин-

тензивни усилия. Това може да се отдаде основно на сложността на определянето и координирането на едновременните задачи, липса на алгоритми и стандартизирана среда и инструменти за разработка. Ако се разглежда в контекста на бързия темп на развитие на микропроцесорите и усилията в разработването на софтуер, по време на който основния хардуер е възможно вече да е остарял, то разработчика е склонен да преосмисли необходимостта от отдаване на значителни усилия за използване на паралелизъм като средство за ускоряване на приложенията [2].

От друга страна все по-вече от споменатите по-горе предпоставки започват да изчезват или вече са изчезнали. Налице са някои недвусмислени тенденции в хардуерния дизайн, които показват че използването на архитектура с един процесор няма да е в състояние да обезпечи необходимостта от реализиране на увеличение на производителността в бъдеще. Основно това е резултат

тат от липсата на неявен паралелизъм, и други тесни места, като пътят на данните и паметта. В същото време стандартизираните хардуерни интерфейси и разработваните вградени системи способстват за намаляване на времето за разработване на паралелна система базирана на микропроцесори. Освен това бе постигнат значителен напредък в стандартизацията на програмните среди, за да се осигури по-дълъг жизнен цикъл за паралелните приложения. Всичко това дава убедителни аргументи в ползата и необходимостта от паралелните изчислителни платформи и системи [3].

Целта на настоящата статия е да се проучи и обобщи информацията относно възможността за изграждане на суперкълъстер базиран на вградени микропроцесорни системи и използването му в автоматизацията.

ИЗЛОЖЕНИЕ

С развитието на системите за управление терминът система, често остава пренебрегван. Скрит остава и фактът, че всъщност паралелизацията на изчислителните процеси обезпечаващи дадено управление се използва в практиката от много рано. Въпреки това, системи за управление използващи набор от процесори и изпълняващи паралелни изчисления се използват в краен случай, само за изключително сложни задачи свързани с симулация, моделиране и анализиране в области като енергетика [4], комуникации [5], образование [6], медицина [7], икономика [8], електроника [9], биология [10], химия [11], физика [12] и други.

Използването на суперкълъстери в областта на автоматиката не се изчерпва само по отношение на управлението на дадени процеси [13,14,15]. Може да се направи сравнителен анализ и съпоставка с други често използвани процесорни технологии като: програмируемите логически контролери (PLC), микроконтролери (MC), DSP процесори, FPGA програмируеми матрици и др. Суперкълъстера не отстъпва по отношение на предимствата на описаните по-горе технологии, които са:

- Възможност за имплементация на достатъчно сложни закони за управление;
- Възможност за разширение;
- Висока надеждност;

- Реализиране на сложни алгоритми за филтрация, предварителна обработка, операции с матрици и други в реално време;
- Осигуряване на периферни модули и др. [16]

В повечето случаи един суперкълъстер превъзхожда споменатите хардуерни системи и то в пъти, като се има предвид производителността, надеждността, възможността за разширение, добавяне на допълнителни входове и изходи, ресурси на системата и т.н.

Използването на суперкълъстери в задачи за управление не присъства явно в разработките на автори използващи паралелни изчисления. Поради тази причина могат да се начертаят само определени тенденции и необходимост от използването на суперкълъстери при управлението на технологични обекти.

Основно паралелните изчисления са необходими в системи за управление на обекти, използващи сложни методи и алгоритми, които често се състоят от комбинация между интелигентни методи за управление и класически закони за регулиране. В такива системи най-често се използват изкуствени невронни мрежи [17,18], размита логика [19,20], адаптивни регулатори [21], робастни регулатори [22,23], оптимално управление [24], моделно предсказващо управление [25,26] и др.

Освен високата производителност необходима при реализирането на тези видове управляващи устройства, използването на суперкълъстери в автоматизацията ще даде и големи предимства по отношение на визуализацията [27], кибер-сигурността [28,29], използване и управление на допълнителни системи [30], реализиране на централизирано управление и др.

По отношение на изграждането на суперкълъстерите, основен приоритет остава най-вече цената. Нарастващата популярност обаче на единно бордовите компютри, като Raspberry Pi определено е решение на проблема, тъй като те дори се приемат за причина за създаването на суперкълъстери, базирани на вградени системи.

Развитието на Raspberry Pi е толкова бурно, че до 2018 са произведени около 18

милиона броя, а само за 2019 – 30 милиона броя [31].

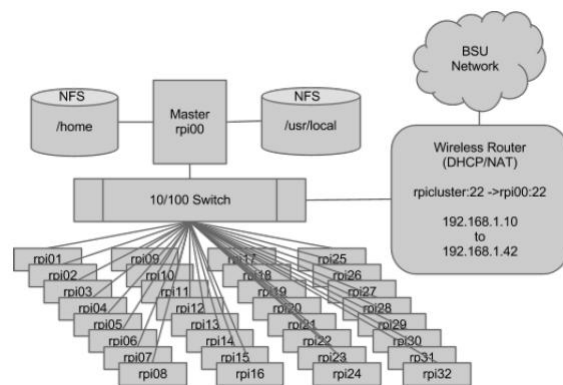
Този тип едноплаткови компютри се използват широко за редица инженерни решения в областта на автоматизацията [32,33,34,35,36]. Явяват се и предпочитано звено в изграждането на суперкълъстери за различни приложения по цял свят [37,38,39,40,41,42,43], тъй като притежават конвенционална Ethernet връзка, необходима за ефективно създаване на кълъстер на Beowulf.

ИЗГРАЖДАНЕ НА СУПЕРКЪЛЪСТЕР БАЗИРАН НА RASPBERRY PI

Най-често реализирания суперкълъстер е т.н. суперкълъстер на Beowulf. Този тип суперкълъстери най-общо представлява набор от еднакви компютърни хардуерни системи, свързани в мрежа и изпълняващи определен софтуер за паралелна обработка, които позволява на всеки възел в кълъстера да споделя данни и изчисления.

В кълъстер на Beowulf приложните програми никога не виждат изчислителните възли (наричани още подчинени компютри), а взаимодействат само с "Master" възела, който е специфичен компютър, обработващ планирането и управлението на подчинените. В типичното изпълнение Master възела има два мрежови интерфейса, единият, който комуникира с частната мрежа на Beowulf за подчинените възли, а другият е за мрежата с общо предназначение. Подчинените компютри обикновено имат собствена версия на същата операционна система, както и локална памет и дисково пространство. Частната подчинена мрежа обаче може да има и голям и споделен файлов сървър, който съхранява глобални постоянни данни, достъпни за подчинените при необходимост [44].

Необходими са само пет основни компонента за изграждане на суперкълъстер: компютърен хардуер, Linux ОС, MPI библиотека, Ethernet switch и рутер. Ако избраният хардуер е Raspberry Pi, то един 32 ядрен суперкълъстер би имал архитектурата показана на фиг. 1[45].



Фиг. 1. Архитектура на суперкълъстер базиран на Raspberry Pi

Самото изграждане на суперкълъстера е сложен и продължителен процес, но може да се опише с няколко основни, но задължителни стъпки, които са:

- Инсталиране на Lixix базирана ОС на хардуерните възли;
- Инсталиране на софтуер за паралелно програмиране (MPI);
- Проверка за работоспособност;
- Създава се копие на Master възела;
- Добавят се подчинените Salve възли;
- Системата се конфигурира и разширява с допълнителни възли един по един.

Суперкълъстерите на Beowulf имат три основни характеристики, които са:

- Единствената цел на възлите е да обслужва Beowulf кълъстера в мрежата;
- Всички възли работят с Open source софтуер;
- Beowulf суперкълъстера се определя, като изчислителна техника с висока производителност [46].

Един от проблемите при проектирането на кълъстера е колко плътно са свързани отделните възли. Например, една компютърна задача може да изисква честа комуникация между възлите: това означава, че кълъстерът споделя специална мрежа, възлите са гъсто разположени и вероятно са хомогенни възли. Другата крайност е, когато компютърната работа използва един или няколко възела и се нуждае от малко или никаква комуникация между възлите, прилагайки принципа на елетронно-изчислителната мрежа.

По отношение на програмното осигуряване, особено значение има софтуера за паралелно програмиране. MPI (Message-Passing Interface) е преобладаващата спецификация, модел и библиотека за паралелно програмиране за научни изчисления днес, превръщайки я в ключова технология [47].

Всички елементи на MPI са от значение, но основно може да се каже, че MPI обръща внимание предимно на модела за паралелно програмиране, чрез предаване на съобщения, посредством които данните се преместват от адресното пространство на един процес в това на друг процес, използвайки съвместни операции за всеки от процесите. Реализираните разширения на класическия модел за предаване на съобщения се изразяват в колективните операции, операции за достъп до отдалечена памет, създаване на динамични процеси и паралелни I/O операции. Тъй като MPI е спецификация, а не изпълнение то съществуват множество реализации. Спецификацията е за библиотечен интерфейс, и MPI не се счита за език, а всички операции са представени като функции, инструкции или методи, съгласно съответните езикови свързвания, като C/C++ и Python са част от стандарта.

Основните предимства на MPI са преносимостта и лекотата на използване. В среда за комуникация с разпределена памет, например в която рутинните програми на високо ниво и абстракциите са изградени върху рутинни програми за предаване на съобщения от по-ниско ниво предимствата на този стандарт са особено очевидни. Подобрява се и мащабируемостта, чрез предоставянето на ясно дефиниран набор от процедури, които могат да се прилагат ефективно за случаи при които има хардуерна поддръжка. Като цяло общата цел на MPI е да се установи широко използван, практичен, преносим ефективен и гъвкав стандарт за писане на програми за предаване на съобщения.

Производителността, цената и консумираната електроенергия са ключови характеристики на почти всеки новосъздаден суперкълъстер. Редица автори представят изследванията си за тези характеристики, но без да се уточнява конкретната изчислителна задача или използват готови тестови

приложения. Това поставя тези изследвания в областта на компютърните системи и технологии, но не намалява значимостта им и по отношение на евентуални приложения на дадената система при управление на технологични обекти.

Основно се дават зависимостите на производителността от размера на кълъстера, използваната памет, сложност на задачата и др. Представят се и определени функционални зависимости за необходимия брой възли, спрямо други параметри, които зависят от конкретната задача или от енергийната ефективност на системата. Почти навсякъде фигурират данни за себестойността на цялата система [29,37,38,39,48,49,50,51,52]. Обобщаващите резултати от тези изследвания показват, че суперкълъстер реализиран на хардуерна основа с Raspberry Pi е 10 пъти по-производителен в сравнение с реализации с други SBC, цената за 1 Gflops е най-ниската (само \$21,43), но е и енергийно най-неефективен.

ЗАКЛЮЧЕНИЕ

В статията е представен преглед на научните изследвания в областта на разработването и анализа на суперкълъстери, изградени от едноплаткови вградени хардуерни системи с оглед на приложението им в областта на автоматизацията и управлението на технологични процеси и обекти. Представени и обобщени са най-често използваните елементи на суперкълъстерите, като основно е наблегнато на хардуерните възли и софтуера за паралелно програмиране. Обобщени са и резултати от изследвания за производителността, консумираната електроенергия и цената на подобен тип системи.

REFERENCE

- [1] Z. J. Czech, "Introduction to parallel computing," 1st Edition, Cambridge University Press, 2017.
- [2] Giles, M. B., and I. Reguly. "Trends in high-performance computing for engineering calculations." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 372.2022, 2014.
- [3] C. S. Yeo, et al., "Cluster computing: High – performance, high – availability and high – throughput processing on a network of

- computers", In Zomaya, A. Y. (eds) Handbook of Nature – Inspired and Innovative Computing, Springer, Boston, MA, 2006, pp. 521 – 551.
- [4] Gagnon, Richard, et al. "Large-scale real-time simulation of wind power plants into Hydro-Québec power system." Ninth International Workshop on Large-scale Integration of Wind Power into Power Systems as well as on Transmission Networks for Offshore Wind Plants, Quebec City, Canada, 2010.
- [5] Woh, Mark, et al. "Mobile supercomputers for the next-generation cell phone." Computer 43.1, 2010, pp.81-85.
- [6] Fernández, Álvaro, et al. "Supercomputers to improve the performance in higher education: A review of the literature." Computers & Education 128 (2019): 353-364.
- [7] Miyano, Satoru. "Revolutionizing cancer genomic medicine by AI and supercomputer with big data." 2018 IEEE Symposium on VLSI Technology. IEEE, 2018.
- [8] Okrepilov, Vladimir V., et al. "Application of supercomputer technologies for simulation of socio-economic systems." R-Economy. 2015. Vol. 1. Iss. 2 1.2 ,2015, pp.340-350.
- [9] Shibut, I. "APPLICATION OF SUPERCOMPUTER TECHNOLOGIES FOR THE THERMAL ANALYSIS OF RADIO-ELECTRONIC DEVICES." Nauchno-Tekhnicheskii Vestnik Informatsionnykh Tekhnologii, Mekhaniki i Optiki 10.2, 2010.
- [10] Zhang, Peng, et al. "Scalability Test of multiscale fluid–platelet model for three top supercomputers." Computer physics communications 204, 2016, pp.132-140.
- [11] Sulimov, Alexey V., et al. "New generation of docking programs: Supercomputer validation of force fields and quantum-chemical methods for docking." Journal of Molecular Graphics and Modelling 78, 2017, pp.139-147.
- [12] Goncharsky, A. V., S. Yu Romanov, and S. Yu Seryozhnikov. "Comparison of the capabilities of GPU clusters and general-purpose supercomputers for solving 3D inverse problems of ultrasound tomography." Journal of Parallel and Distributed Computing 133, 2019, pp.77-92.
- [13] Penaflor, B. G., et al. "Real-time control of DIII–D plasma discharges using a Linux alpha computing cluster." Fusion engineering and design 56, 2001, pp.739-742.
- [14] Jablkowski, Boguslaw, Ulrich Thomas Gabor, and Olaf Spinczyk. "Evolutionary planning of virtualized cyber-physical compute and control clusters." Journal of Systems Architecture 73, 2017, pp. 17-27.
- [15] Baker, M., et al. "Cluster Computing and Applications," in the Encyclopedia of Computer Science and Technology.", 2001.
- [16] Petkov, Petko Hristov, Jordan Konstantinov Kralev, and Tsonyo Nikolaev Slavov. Design of embedded robust control systems using Matlab/Simulink. Institution of Engineering & Technology, 2018.
- [17] Dagli, Cihan H., ed. Artificial neural networks for intelligent manufacturing. Springer Science & Business Media, 2012.
- [18] Ormandzhiev K., S. Yordanov, "Neuro-Fuzzy Control Synthesis for Electro-Hydraulic Follow-up System," 2019 16th Conference on Electrical Machines, Drives and Power Systems (ELMA), Varna, Bulgaria, 2019, pp. 1-6.
- [19] Jager, R., Fuzzy logic in control. 1995.
- [20] Ormandzhiev K., St. Yordanov , S. Stoyanov, Synthesis of fuzzy controller for cross-flow water turbine, March 2017, Information Technologies and Control 15(1): 9-16, Print ISSN: 1312-2622; Online ISSN: 2367-5357, DOI: 10.1515/itc-201
- [21] Åström, Karl J., and Björn Wittenmark. Adaptive control. Courier Corporation, 2013.
- [22] Gu, Da-Wei, Petko Petkov, and Mihail M. Konstantinov. Robust control design with MATLAB®. Springer Science & Business Media, 2005.
- [23] Yordanov St., K. Ormandjiev, G. Mihalev, „Robust control of cross-flow water turbine“, Automatics and informatics, Sofia 3-5 October 2019, pp. 49-52
- [24] Liu, Derong, and Qinglai Wei. "Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems." IEEE Transactions on Cybernetics 43.2, 2013, pp. 779-789.
- [25] Wang, Liuping, Model predictive control system design and implementation using MATLAB®. Springer Science & Business Media, 2009.
- [26] Mihalev G., St. Yordanov, “Model predictive control of vibratory bowl feeder“, UNITEX 2019, 15-16 November, 2019, pp.I-269-275
- [27] Stefanov, Konstantin, et al. "Dynamically reconfigurable distributed modular monitoring system for supercomputers (DiMMon)." Procedia Computer Science 66, 2015, pp. 625-634.
- [28] Mincho Hadjiiski, Cybersecurity of control and e-leadership systems, Journal of informatics and innovative technologies (JIIT), Plovdiv, 2019.
- [29] Matthews, Suzanne J., Raymond W. Blaine, and Aaron F. Brantly. "Evaluating single board

- computer clusters for cyber operations." 2016 International Conference on Cyber Conflict (CyCon US). IEEE, 2016.
- [30] Marković, Dušan, et al. "Image Processing on Raspberry Pi Cluster." *International Journal of Electrical Engineering and Computing* 2.2, 2018, pp. 83-90.
- [31] <https://www.zdnet.com/article/raspberry-pi-now-weve-sold-30-million/>
- [32] Kazala, R., and P. Straczynski. "The Most Important Open Technologies for Design of Cost Efficient Automation Systems." *IFAC-PapersOnLine* 52.25, 2019, pp. 391-396.
- [33] Dorzhigulov, Anuar, et al. "ANFIS based quadrotor drone altitude control implementation on Raspberry Pi platform." *Analog Integrated Circuits and Signal Processing* 95.3, 2018, pp. 435-445.
- [34] Sobota, Jaroslav, et al. "Raspberry Pi-based HIL simulators for control education." *IFAC-PapersOnLine* 52.9, 2019, pp.68-73.
- [35] Yordanov S., V. Mitev, T. Kovachev, 3-degree manipulator control with Raspberry Pi and CodeSys, *UNITEX 2018*, 16-17 November, Gabrovo, 2018, pp. I-285-292
- [36] Mitev V., S. Yordanov, G. Mihalev, Process Controller with Integrated Autotunig Module, *International Scientific Conference UNITEX 2019*, 15 – 16 November 2019, Gabrovo, ISSN 1313-230X
- [37] Baun, Christian. "Mobile clusters of single board computers: an option for providing resources to student projects and researchers." *SpringerPlus* 5.1, 2016.
- [38] Myint, Khin Nyein, Myo Hein Zaw, and Win Thanda Aung. "Parallel and Distributed Computing Using MPI on Raspberry Pi Cluster." *International Journal of Future Computer and Communication* 9.1, 2020.
- [39] Qureshi, Basit, and Anis Koubâa. "On energy efficiency and performance evaluation of single board computer based clusters: A hadoop case study." *Electronics* 8.2, 2019.
- [40] Bleeker, Ellen-Louise, and Magnus Reinholdsson. "Creating a Raspberry Pi-Based Beowulf Cluster.", 2017.
- [41] Doucet, Kevin, and Jian Zhang. "Learning cluster computing by creating a Raspberry Pi cluster." *Proceedings of the SouthEast Conference*, 2017.
- [42] Wazir, Saad, et al. "Performance Comparison of MPICH and MPI4py on Raspberry Pi-3B Beowulf Cluster." *arXiv preprint arXiv:1911.03709*, 2019.
- [43] Ağbahca, Erdem, and Adem Alpaslan Altun. "Performance Test of MPI on Raspberry Pi 2 Beowulf Cluster." *International Journal of Computing* 5.4, 2016, pp. 233-238.
- [44] Daydé, Michel, E. Pacitti, and J. Lopes. "High performance computing for computational science." *VECPAR*, Berlin Heidelberg, Germany, 2007.
- [45] Kiepert, Joshua. "Creating a raspberry pi-based beowulf cluster." *Boise State*, 2013.
- [46] Robert G Brown. *Engineering a beowulf-style compute cluster*. Duke University Physics Department, 2004.
- [47] Chunduri, Sudheer, et al. "Characterization of MPI usage on a production supercomputer." *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2018.
- [48] Basford, Philip J., et al. "Performance analysis of single board computer clusters." *Future Generation Computer Systems* 102, 2020, pp. 278-291.
- [49] Orhean, Alexandru Iulian, Florin Pop, and Ioan Raicu. "New scheduling approach using reinforcement learning for heterogeneous distributed systems." *Journal of Parallel and Distributed Computing* 117, 2018, pp. 292-302.
- [50] Simon J. Cox, James T. Cox, Richard P. Boardman, Steven J. Johnston, Mark Scott, Neil S. O'Brien, *Iridis-pi: a low-cost, compact demonstration cluster*, *Cluster Computing*, June 2013.
- [51] Papakyriakou, Dimitrios, Dimitra Kottou, and Ioannis Kostouros. "Benchmarking raspberry Pi 2 beowulf cluster." *International Journal of Computer Applications* 975, 2018.
- [52] Priyambodo, T. K., A. W. Lisan, and M. Riassetiawan. "Inexpensive green mini supercomputer based on single board computer cluster." *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 10.1-6, 2018, pp. 141-145.