27.05.2017

**Knots concept – yet another computing solution**

1) knots computer is an internal network-based fat client computer architecture for GPGPU computing with practical aspects in mind. It is designed for small Research & Development up to 10 people teams. It consists of a collection of fully operational user workstations and a background distributed computing architecture,

2) it is based on Linux operating system and NVidia CUDA C GPU programming language provided independently of single user Personal Computer workloads,

3) single knot ( from overal recommended 10 knots per switch ) proposition:

MOTHERBOARD: FM2A88X Extreme6 ( 1xPCIe 3.0 x16; 1xPCIe 3.0 x4; ATX format ),
CPU: AMD A10 7850k ( Accelerated Processing Unit – CPU with GPU ),
RAM: 2x8GB 1333MHz ( majority of users works seldom exceeds 6GB of used RAM ),
HDD: 4x120GB SSD Goodram CX 300 in RAID 0 configuration,
NET: 1port 10GbE PCIe adapter ( LR SFP+ GBIC – up to 10km; AOC SFP+ up to 300m ) at PCIE5 slot,
GPGPU: user-independent **GTX 1070 8GB** at PCIE2 slot, ( theoreticall ~3.4TFLOPS ),
MONITOR: 2x23.8'' HP 24er connected to motherboard,
HEADSET: Sennheiser HD280, keyboard: Lenovo Combo,
STUFF: cool-looking case with at least 500W power suply; pendrive;

4) internal 10GbE network + disk array knot:

-10GbE stacking switch ( Ubiquiti ES-16-XG: 4x10GbE Cat7 uplinks + 12x 10GbE SFP+ ),
-ordinary knot without GPU + network drives on two ancient Dell MD1000 with **new** disks ( 30TB on 2TB SATA 7.2krpm; RAID 5 ) accessible via 8Gigabit Fibre Channel. Disk arrays are in RAID 0 configuration in total RAID 50 ( hardware RAID's 5 + software RAID 0 ),
-such slow disk array should do the job, during current workloads,

5) internet connection with at least 300Mbps bitrate on 24ports GbE switch via 2 firewalls cascade ( bought from different manufacturers ). Redundant access to knot is quite useful in author opinion,

6) each knot is internally interconnected via 10GbE switch. 10-knots set is easily extendable with switches, only disk array knot should be made on server with more PCIE slots ( f.e. Dell r910 ),

7) distibuted – computations program development:

-knots could be efficiently programmed with OpenCL with AMD products only ( R9 Nano ),
-single GPGPU productive kernel program development at first step,
-data distribution from network disks to local drives via Samba ( local drives capacity is much bigger than memories ). Then data partially is consecutively loaded to RAM shared memory ( #shmget ),
-GPGPU works on shared memory on its host ( some problems can be solved with direct reads from network to GPU memory, but author does not recommend such solution ),
-background Operating System service for shared memory handling,
-GNU Parallel ( please cite ) package functionality for work deployment,
-background service for turning off unused knots and Wake on LAN magic packets,

8) 10GbE is less problematic, much more popular and poses higher capabilities than some exotics like Infiniband and Fibre Channel network devices - but it is slower with higher connection latency,

9) data distribution between knots via network is ~8 times slower than RAM shared memory access. Computational problems should be assymetric in read-write operations to computation. Please note quite acceptable connection between knots,

10) there are 4ports GbE PCIe adaptors which could be connected to switch ( for example 52 ports GbE ) via knot 4xGbE aggregated link in more economical designs,

11) please note some custom cases filled with vaseline oil providing significant heat distribution efficiency increase. Long term high - load computations ( f.e. GPU's database ) should be provided on the basis of two **GTX 1080ti**'s,

12) knots concept does not require devices airconditioning resulting in lower overall system power,

13) above mentioned design of 10 knots might provide practical peak of ~60TFLOPS computational capabilities for 40k$. Annually it could consume 18k$ of current at full load.

14) one can take into consideration architecture of non-workstation knots for computation purposes. It is a simiral approach instead of few theoretical tricks:

- there is no need for monitors, headsets, and HDD's - use only pendrive USB3.0 for operating system deployment, knots could be cheaply attached via passive SFP+ DAC's, additionally one can implement Multiple General Purpose Graphics Processing Units,
- all data is read from internal network to GPGPU memory – it's integration must be checked for each transfer, there is a redundant need of signing files and fingerprints calculations,
- use some supervisor server for example Dell R720xd 24SFF; R920, populated with 2x ( 12xSSD's in RAID0 ). One can tests some other solutions f.e. Fujitsu RX2520 M1 SFF server,
- additionaly use disk array Dell MD1220 24SFF, populated with 2x ( 12x SSD's in RAID 0 at each controller ). Connect to Dell R920 with two PCIe Dell PERC H810/ PERC H710P ( controllers PCIe2.0x8 bandwidth bottleneck ) controller from each EMM.
- connect server to switch with acceptable capabilites f.e. Dell 8164f; Cisco Nexus N3K-C3064PQ-10GE, via two or four separate 40GbE PCIe QSFP+ adaptors ( currently in 2017, as far as author know, there is no link aggregation for 40GE – knots must be divided on supervisor server addresses ). Additionally pleaso note, that QSFP+ adaptors requiers PCIe3.0 standard slot for full 40GE network support,
- mainly applicable for specific distributed computations: fast data accesses are economically costly in this architecture ( better suitable for more symmetric problems ),

15) as far as author, know best disk array performance results are for two-stages tree RAID 00 ( software RAID0 on hardware RAID's 0 made on SSD's ) in mdadm. There is some exotic solution like making a ramdisk on older servers with big RAM memories ( Dell R910, R810 ), directly connected via 10GbE's, but it is only up to ~0.256TB of file system space at reasonable costs,

16) during RAID configuration author cuts each drive to constant size ( each manufacturer makes products with some approximation in size f.e. 120GB SSD could be: 111.7GB, 111,71GB, 111,81GB, et cetera... ) at beggining of disk array configuration,

17) generally DAC's have lower practical latencies,

18) cheapest SSD's ( 120GB CX300 Goodram ) have greatest capabilities: IOps / GB of data ( it is not the same coefficient as Read/Write speed approximately ~400MBps ),

19) There is a possibility for higher SSD efficiency in RAID 0 – additional use of USB3.0 ports with SATA-USB3.0 adaptors ( it could be usable in knot, at which one will develop code for other workstations ). There are some PCIe to USB3.0 adaptors which could extend it even more ( PCIe2.0 x1 has only 1GBps throughput resulting in only 2 disks per such adaptor ). Disk array could be easily made with RAID controllers for example Dell PERC H200E; H710p etc. ( SATAI 3Gbps is only ~15% slower than real-long term SSD read/write capabilities ), with two miniSAS SFF8088 connectors populated with inline adapter 2xSFF8088 to 8x SATA. With aesthetics in mind one could use 2x( 4x2.5'' ) SATA backplanes. Please note, that such system populated with SSD disks could be more ecological than obtaining single Dell MD1220 device without disks,

Post Scriptum: additionally each knot CPU via AMD APU provides ~0.8TFLOPS computations capabilities in heterogenous programming model,

Post Post Scriptum: please note, that knots capabilities could be easily extended with Volta microarchitecure devices in future,

Post Post Post Scrpitum if one is searching for some reasonable decryptor, one would design well-tested Application-Specific Integrated Circuit and would buy enough amount of such devices. There is a similar approach platform made by Adapteva.