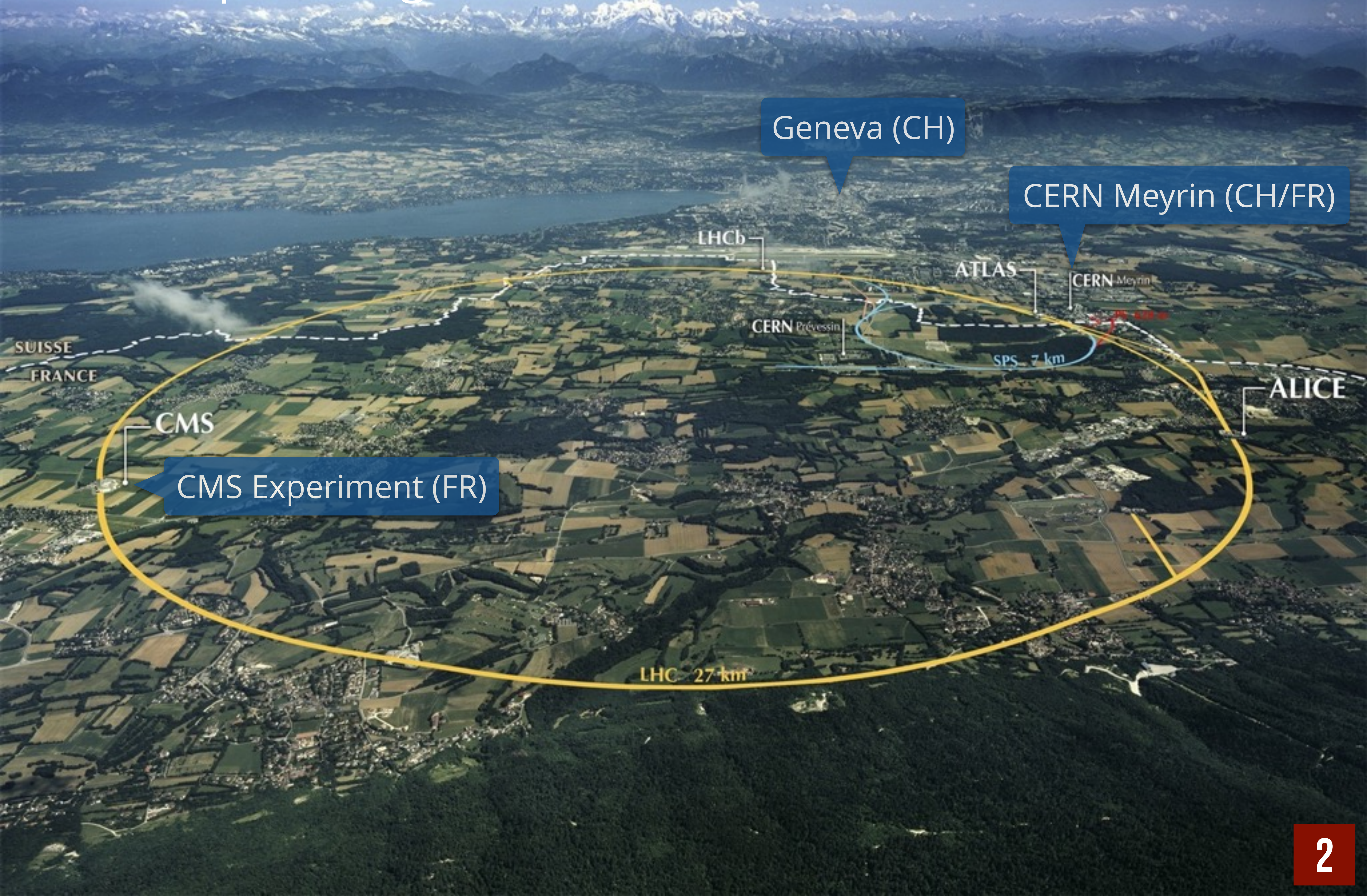
EXPLORATION OF FUTURE COMPUTING PLATFORMS AT CMS

COMPUTING ON LOW-POWER ARCHITECTURES (COLA), 25.02.2016

DAVID ABDURACHMANOV (FERMILAB)

CERN
The European Organization for Nuclear Research



Power In Data Centers

An Inconvenient Truth

- Energy-related costs account for approximately 12 percent of overall data center expenditure and are the **fastest-rising cost in the data center,** according to Gartner, Inc. (September 29, 2010)
- CMS for **2012** data used **~100K** x86_64 cores from **~350K** cores at Worldwide LHC Computing Grid (WLCG)
- Scaling up from the mix of machines at FNAL we estimate WLCG aggregate power consumption for machines at **10MW**
- **CMS expects 2 to 3 orders of magnitude increase in data produced in 15 years**

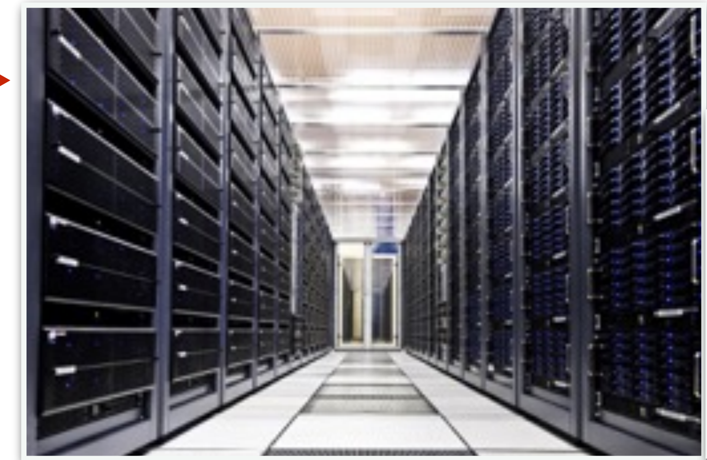
Think Green

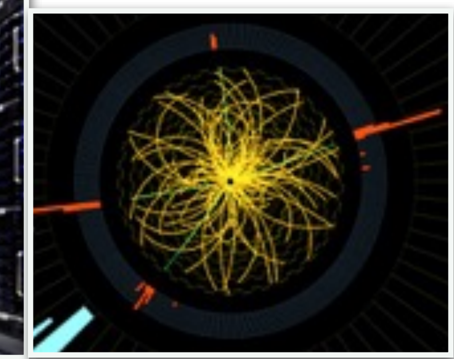
- Local green or/and cheaper power source, e.g., Princeton energy plant (**15MW**) combines electricity, heat and cooling. When electricity cost increased gas, diesel or/and bio-diesel fuel is used to power local generators. Hot water and steam is provided from waste energy.
- Low-power and / or highly efficient hardware, e.g., Intel Atom, X-Gene (ARMv8 64-bit), GPUs, Xeon Phi, FPGA, etc.

Computing & CMS

CMS Detector (HLT)

Worldwide LHC Computing Grid (WLCG)





CMS Detector (HLT)		WLCG
Full ownership ✓	→	**Partially owned**
Single "customer" ✓	→	**Multiple "customers"**
High-bandwidth interconnect ✓	→	**Bandwidth varies**

A virtual super computer (WLCG) is used to store, distribute and process LHC data

- Based on 170 computing centres in 42 countries
- Distribute and analyse ~30PB of data annually generated by LHC
- Experiments produce >15PB of new data annually

Why new architectures?

Distributed computing in **HEP before ~2000** had multiple vendors involved, and incl. special workstations and heterogeneous computing

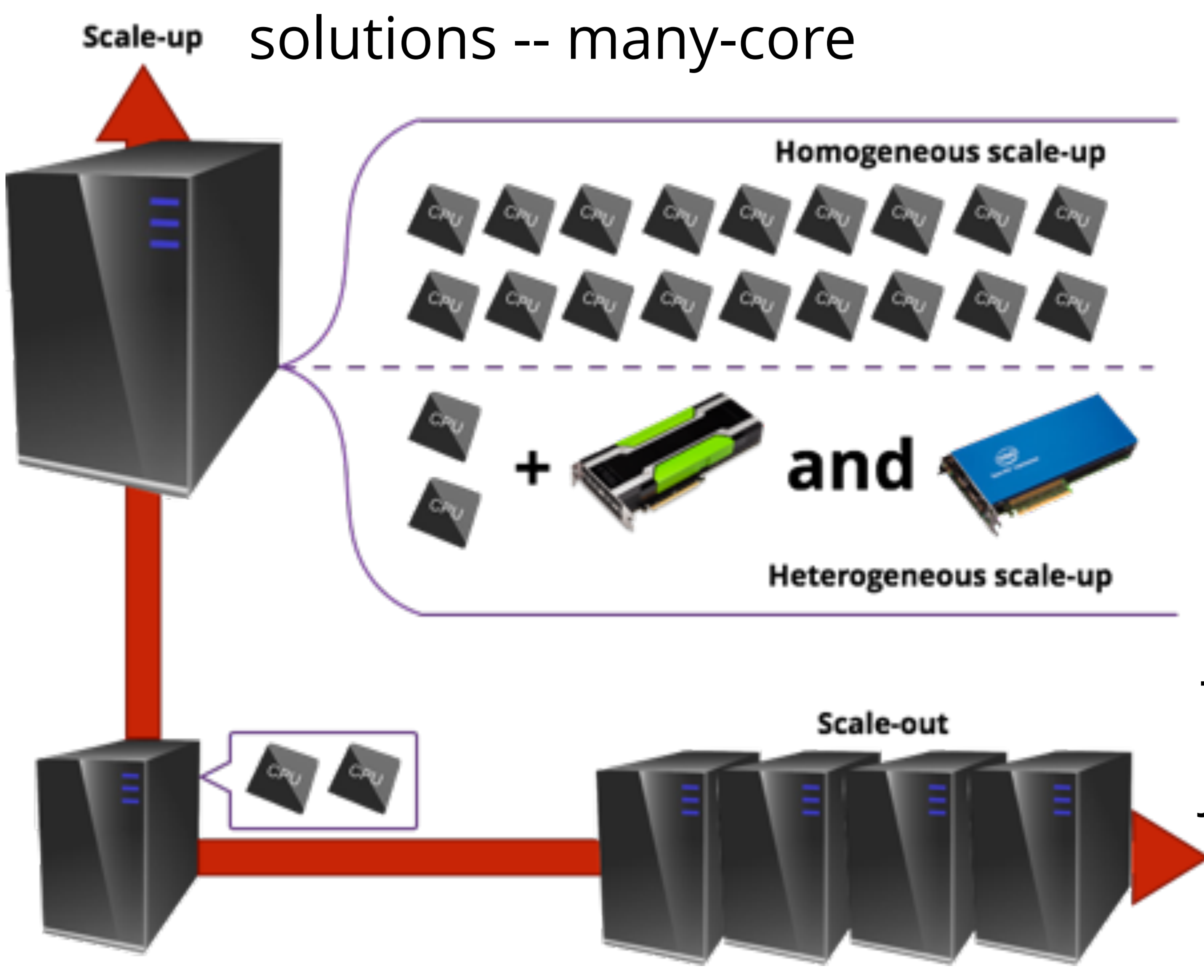
Hight Throughput Computing (HTC) converged on x86/Linux **at ~2000**

- Commodity hardware enabled the current model of WLCG:

  Build Once, Run Everywhere

- **Two vendors:** Intel (dominating) and AMD

The current commodity hardware itself is limited by power wall with stop-gap solutions -- many-core



Specialised processors and heterogeneous computing rise up

- Lightweight general-purpose low-power high-density, vector units, GPUs, Xeon Phis (highly-parallel long-vector), etc

The focus is shifting to **performance/watt**, not just **performance/price**

How we do it?

No single job batch submission system, incl. **LSF**, **HTCondor**, Slurm, SGE, Torque/Pbs

No single storage solution, incl. NFS, GlusterFS, **Hadoop** (popular in US)

Has 100+ different CPUs from the last **10 years**, most 4-5 years old

Common operating system: **RHEL/CentOS/Scientific Linux (SL)**

- Dominated by **SL 6** co-developed by CERN and Fermilab
- **CentOS 7 + CERN Special Interest Group** to follow **SL 6**

Software and essential precomputed data (e.g. LUT) distributed via **CernVM File System (CVMFS)**

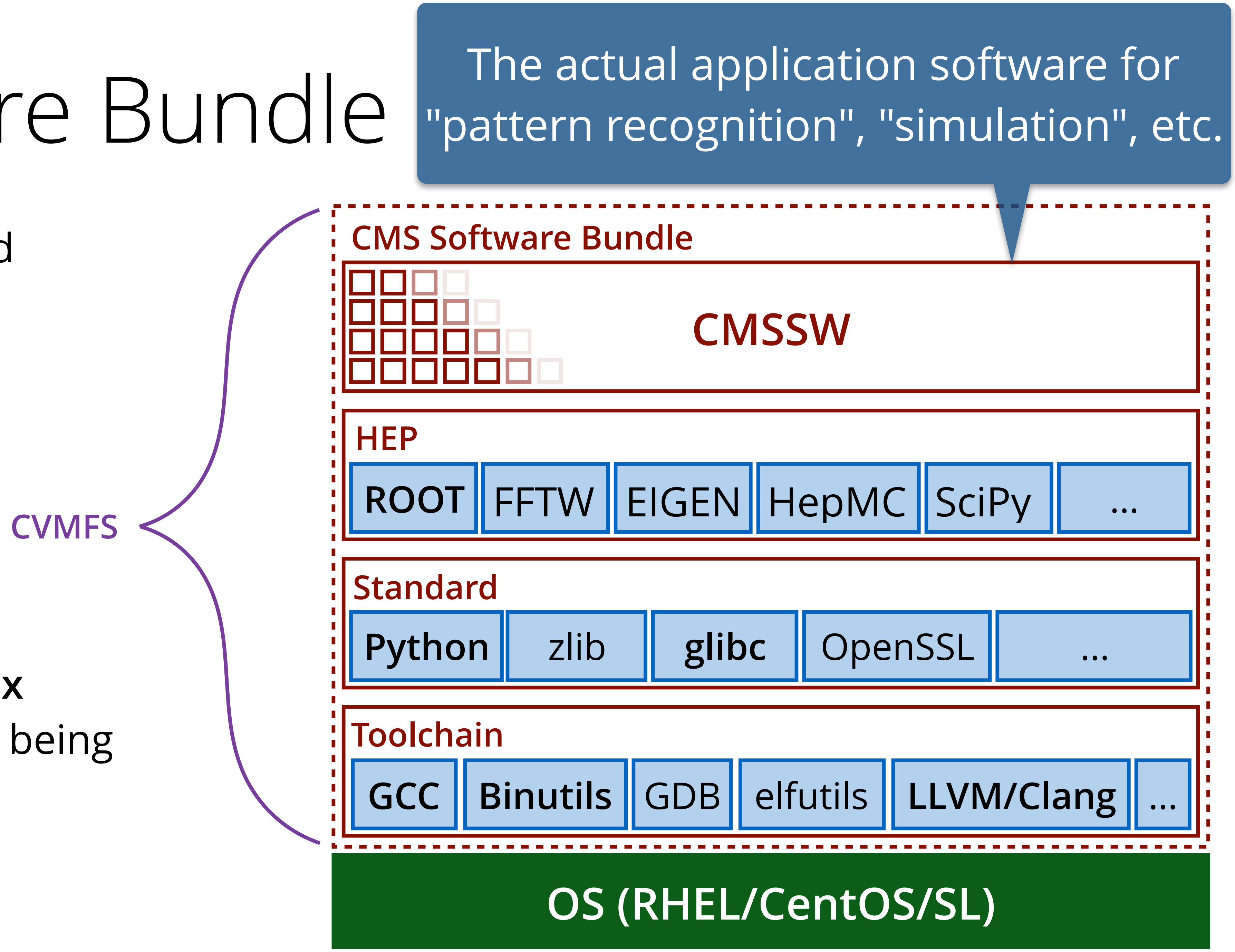
HEP SPEC '06 benchmark is used for accounting in WLCG and by experiments

- Designed to represent worker node activity under full load
- Based on CPU SPEC 2006 **all_cpp** benchmark set

CMS Software Bundle

CMSSW is **open-source** and available at GitHub

Mostly written in **C++14**, C, **Python** and **Fortran**



CMSSW is like **Software Collection** package or **Linux Container** without actually being any of them

Quick comparison:

	CMSSW	Firefox
SLOCs	6M	7M
Initial Release	2005	2002
Contributors	>1300	>1200
Memory Footprint	~2GB	~0.3GB

Other CERN developed software would increase SLOCs

ROOT6 w/o Clang: 1.7M

GEANT4: 1.1M

Porting to **ARMv8 (64-bit)**

CMSSW was originally ported to **ARMv7 (32-bit)** few years ago

- High-end mobile SoC based development boards were used
  - ODROID-U2 (Exynos 4412 Prime), ODROID-XU2 (Exynos 541), Arndale Octa (Exynos 5420), Jetson TK1 (Tegra K1)
- Resolved majority of porting issues and found numerous issues in CMSSW (even affecting x86_64)

CMSSW for **ARMv8 (64-bit)** port was started early

- **Step1**: ARM Foundation Model
- **Step2**: QEMU + binfmt_misc + user mode emulation
- **Step3**: APM Mustang
- **Step4**: HP Moonshot + m400

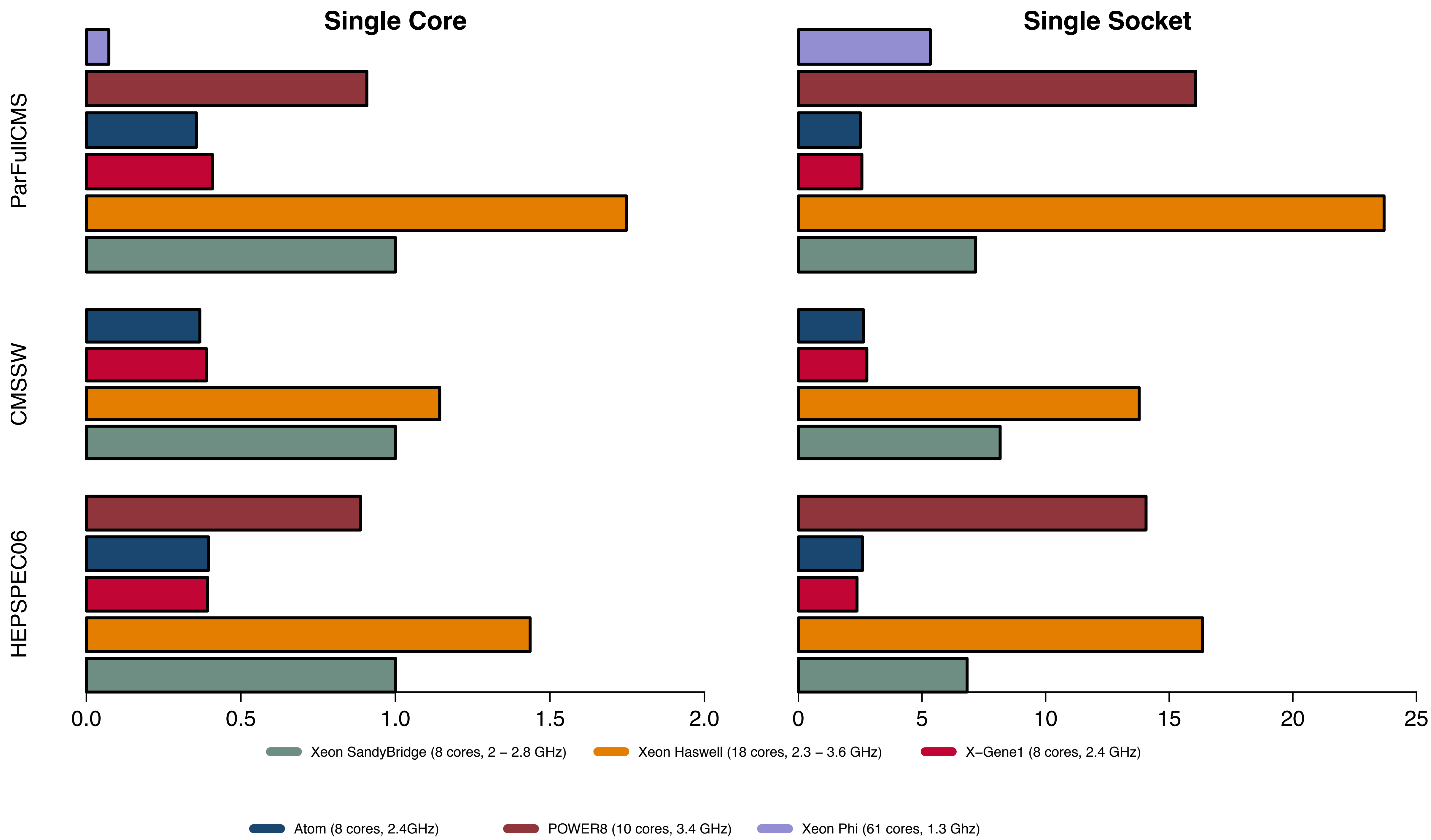
For ARMv8 we wanted to have CMSSW application software and GRID software (e.g., **HTCondor**) for software distribution, data transfers and job management

CPU Specifications

	Vendor	Model	Year	Fab	Process
SandyBridge	Intel	E5-2650	Q1/12	Intel	32nm
Haswell	Intel	E5-2699	Q3/14	Intel	22nm
Atom	Intel	C2750	Q3/13	Intel	22nm
X-Gene 1	APM	883408	Q3/13	TSMC	40nm
POWER8	IBM	8247-22L	Late 13	IBM	22nm
Xeon Phi	Intel	KNC7100	Q2/14	Intel	22nm

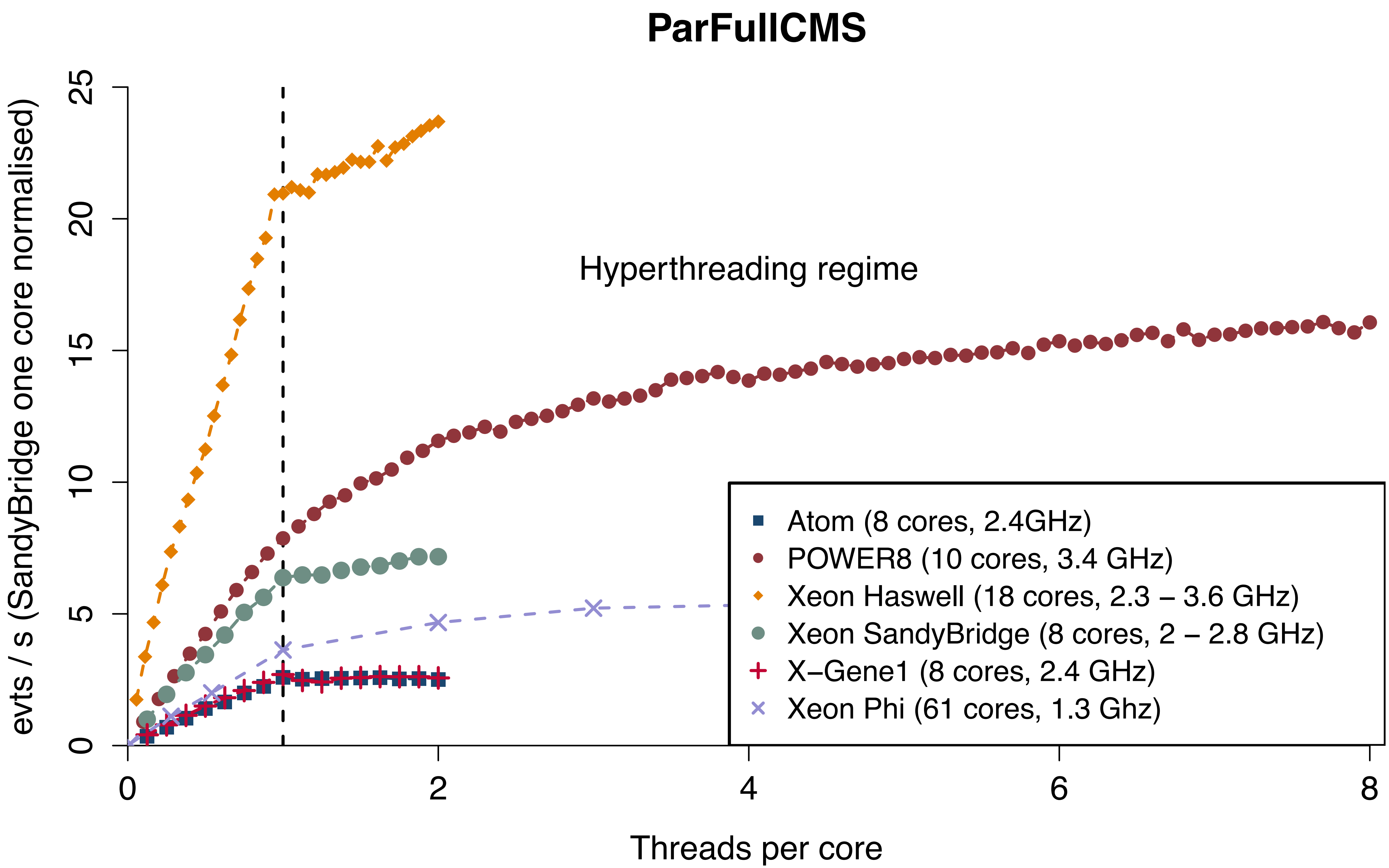
	Frequency (GHz)	Cores	Threads/Core
SandyBridge	2.0 (2.8)	8	2
Haswell	2.3 (3.6)	18	2
Atom	2.4	8	1
X-Gene 1	2.4	8	1
POWER8	3.45	10	8
Xeon Phi	1.23	61	4

Raw Performance



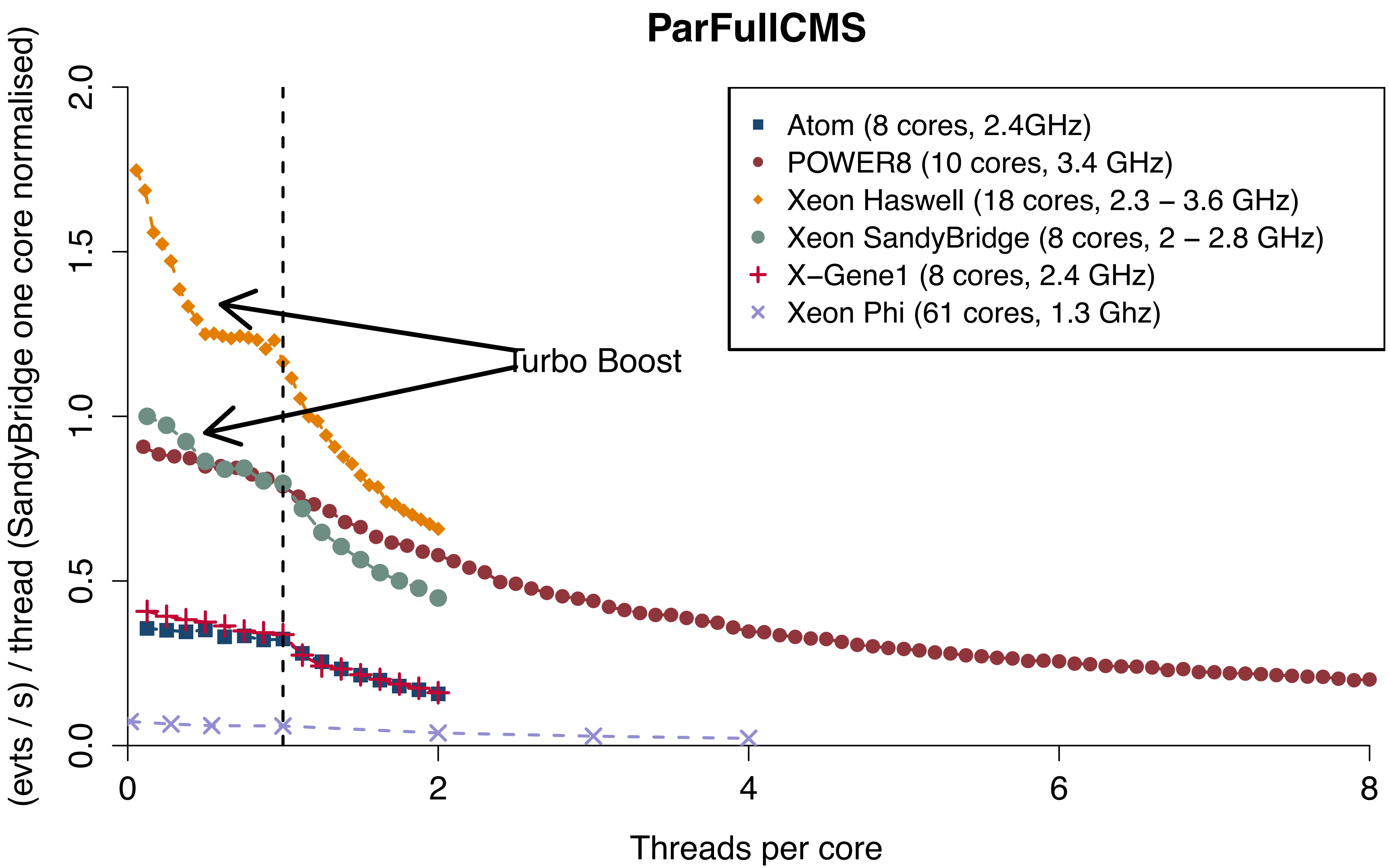
All numbers normalised to Xeon SandyBridge 1 core performance.

Scalability #1



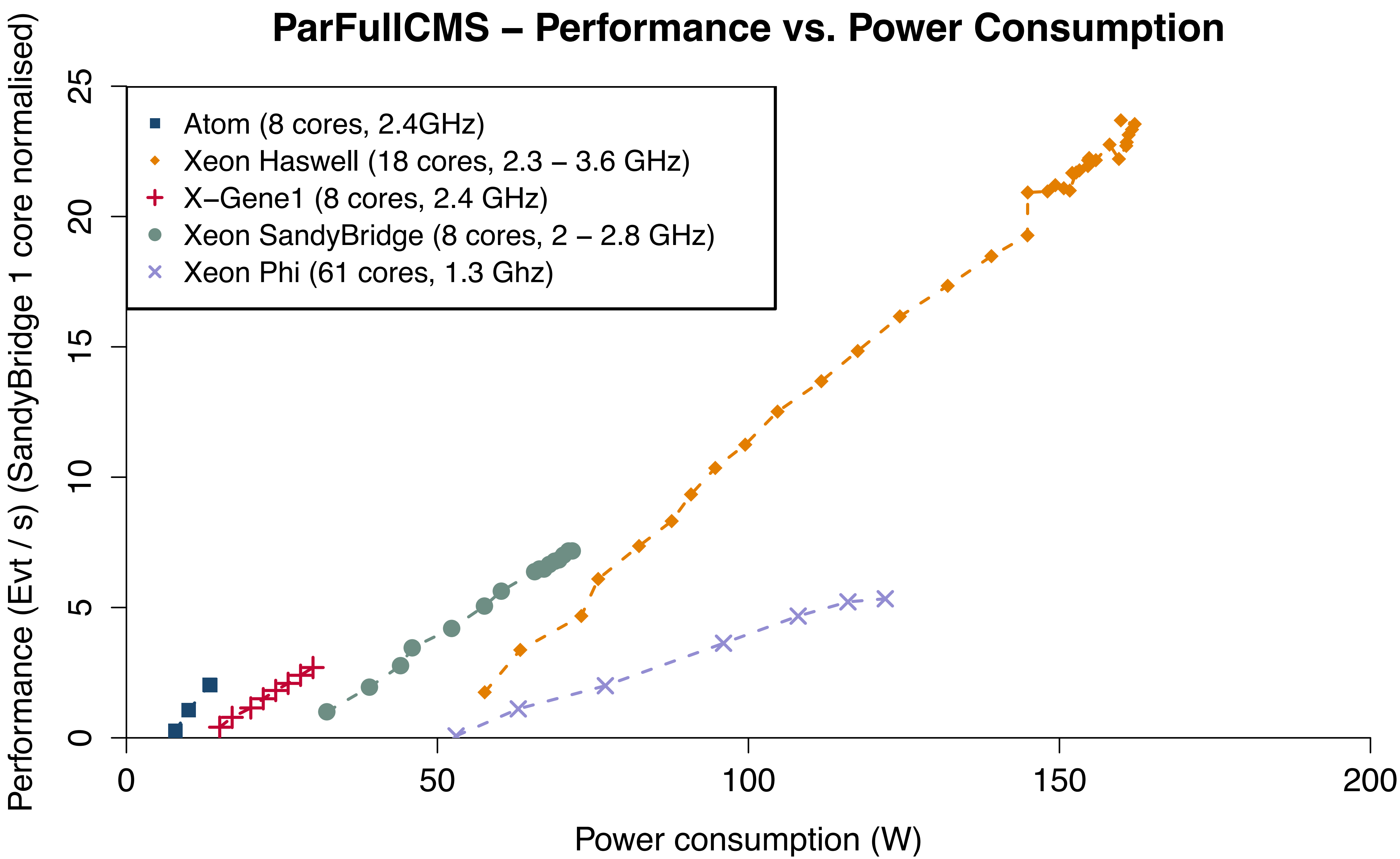
All numbers normalised to Xeon SandyBridge 1 core performance.

Scalability #2



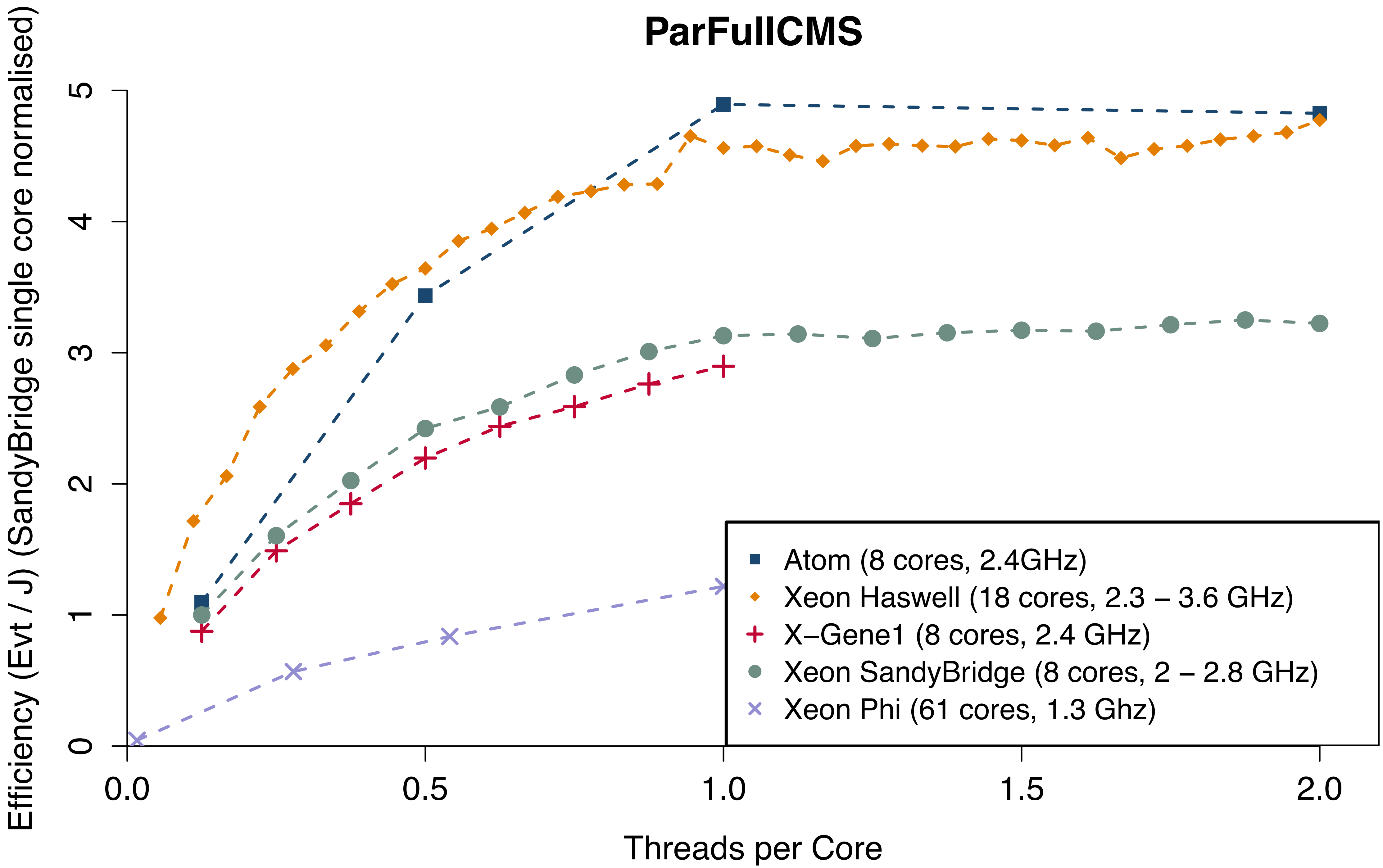
All numbers normalised to Xeon SandyBridge 1 core performance.

Power Efficiency (1S) #1



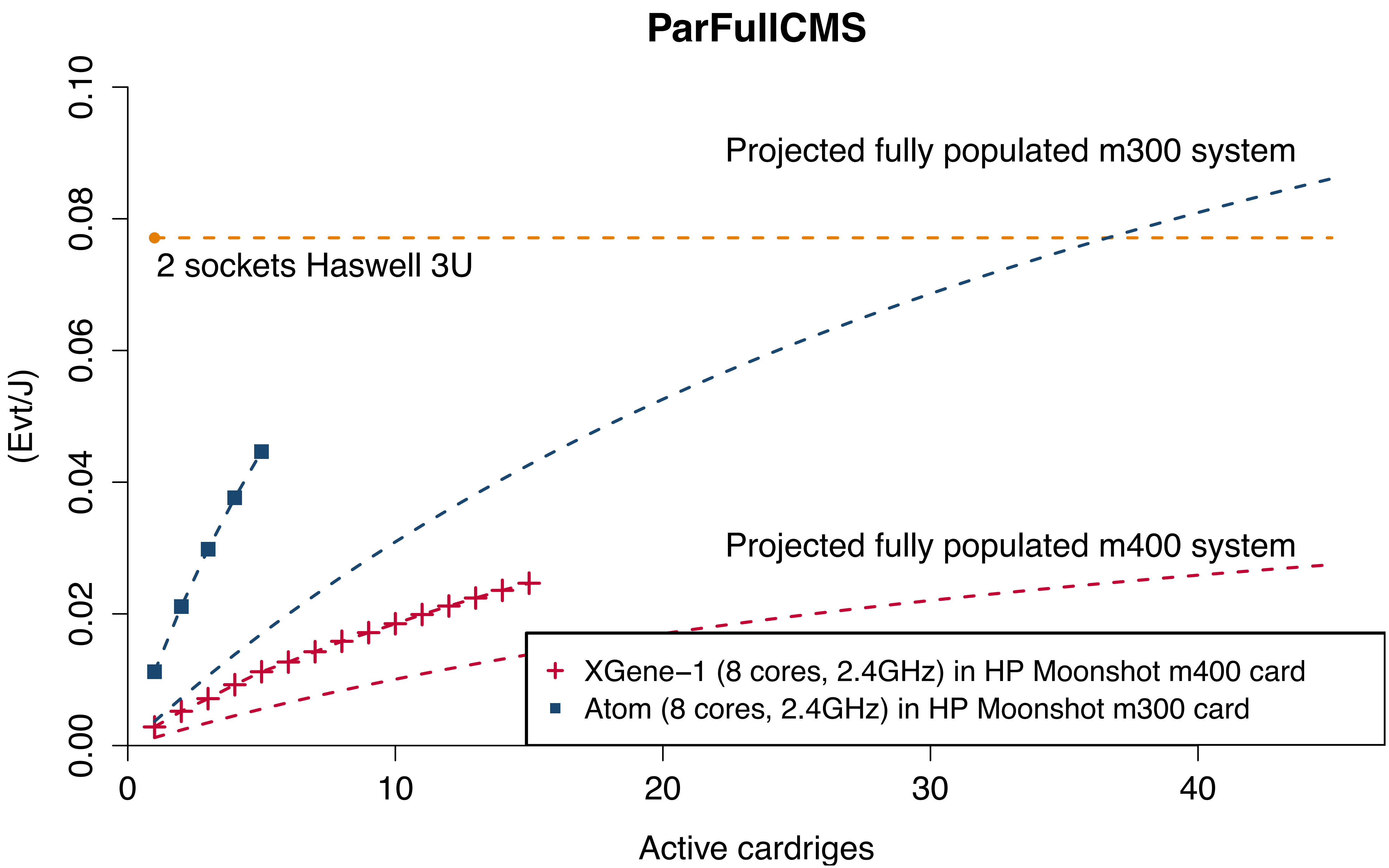
Atom power consumption is for node, not for a CPU.

Power Efficiency (1S) #2



All numbers normalised to Xeon SandyBridge 1 core performance.

Power Efficiency (box)



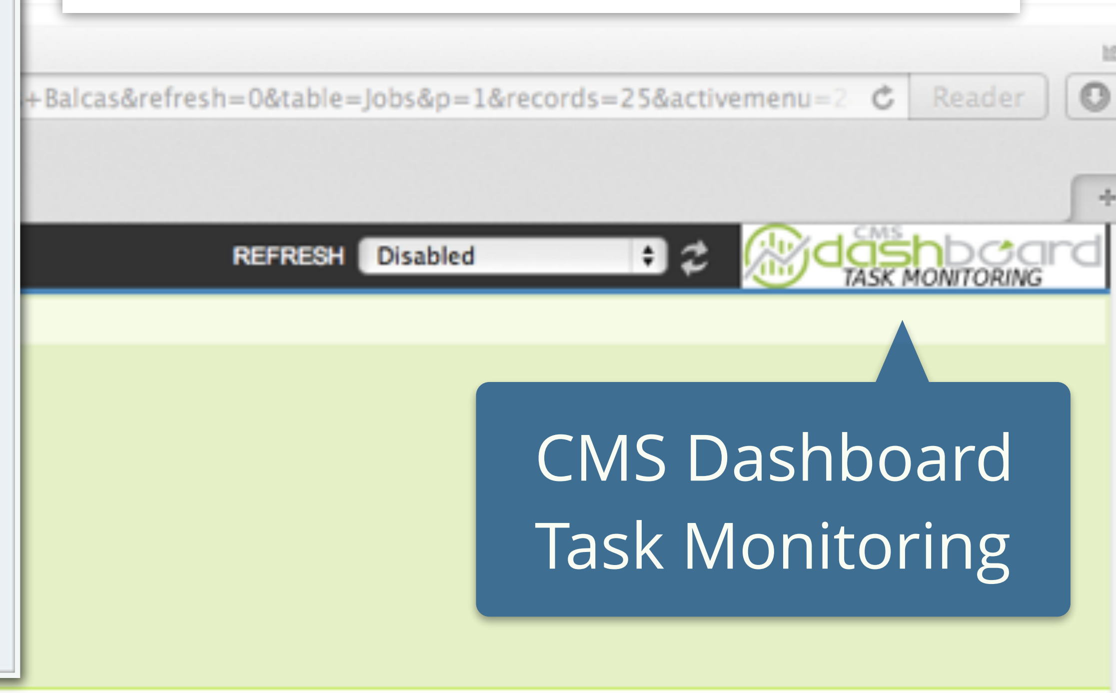
Heterogeneous Tier-3 Site

Goal 1: What is necessary for **AArch64-based** (or any other alternative architecture) production worker nodes to be a credible alternative to **x86_64-based** nodes for use in **WLCG** computing sites (given the availability of application level software like **CMSSW**)?

Goal 2: We wanted to demonstrate that such nodes can be added as a **"drop-in" replacement for x86_64** nodes in **WLCG** and even mixed **heterogeneously**.

With above goals in mind, we created **US_T3_Princeton_ARM** computing site using APM Mustang development board with Open Science Grid (OSG) infrastructure at Princeton University.

On mid-**2015** CMS **successfully** executed CMSSW based job on **AArch64 worker node** via standard job injection pipeline



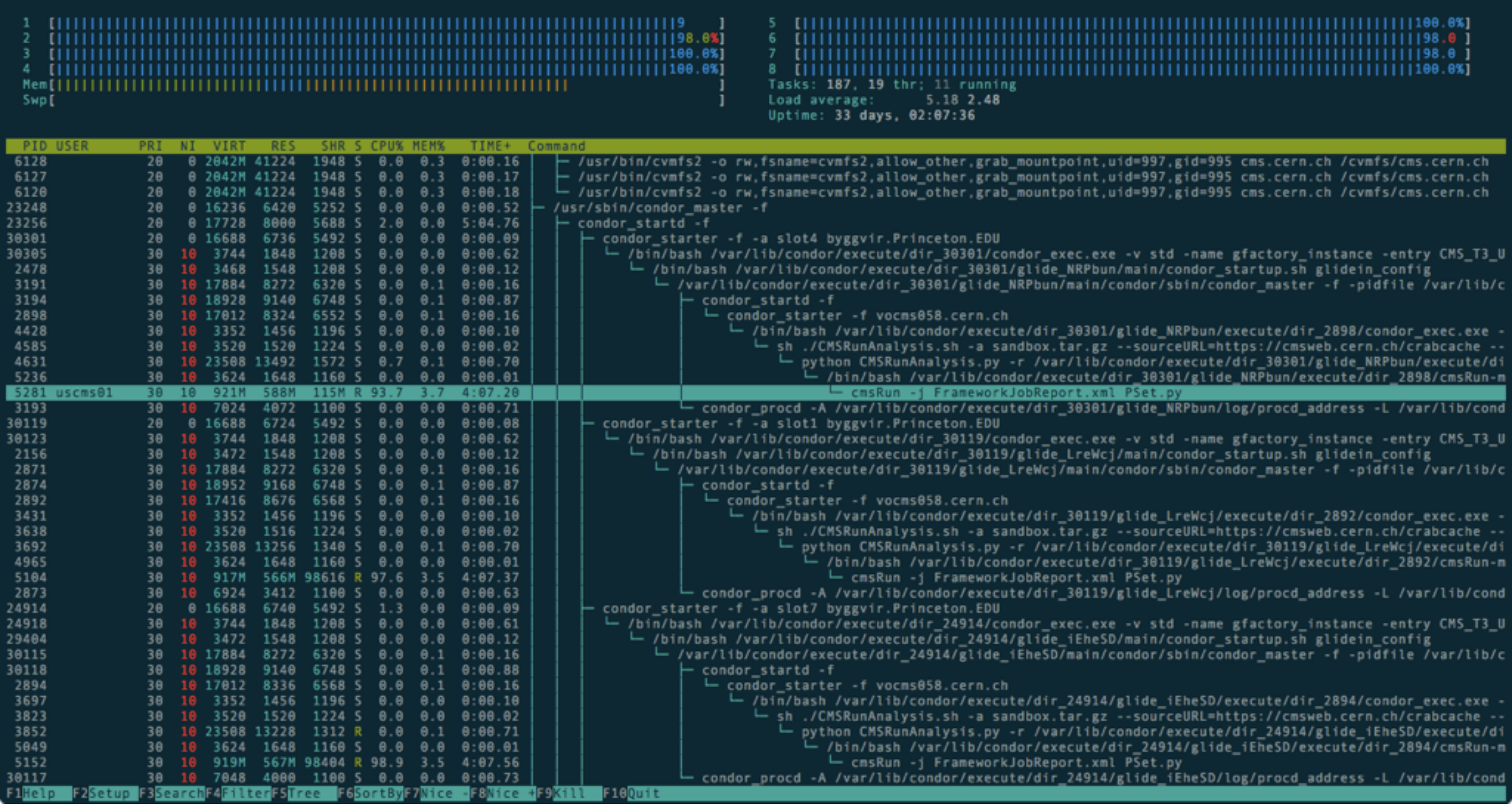
Start » [Justas Balcas] » Tasks » Jobs

Data | Charts | Show 25 entries

Task: **150608_200051:jbalcas_crab_ARM_TEST_2-output2** NJobTotal: **1000** Pending: **822** Running: **0** Unknown: **0** Cancelled: **0** Success: **168** Failed: **2** WNPostProc: **8** ToRetry: **0**

Id	Status	AppExitCode	Site	Retries	Submitted	Started	Finished	Wall Time	Job Log	File Access	FTS File Status
1	finished	0	T3_US_Princeton_ARM	1	2015-06-08T20:01:22	2015-06-08T20:05:35	2015-06-08T20:15:16	00:09:41	Job Log,Job Log JSON,Post Job Log	File Info	N/A
2	finished	0	T3_US_Princeton_ARM	[illegible]	2015-06-08T20:01:22	2015-06-08T20:05:3[illegible]	2015-06-08T20:15:1[illegible]	00:09:38	Job Log,Job Log JSON,Post Job Log	File Info	N/A
3	finished	0	T3_US_Princeton_ARM	1	[illegible]08T20:01:22	[illegible]08T20:05:37	[illegible]08T20:15:25	00:09:48	Job Log,Job Log JSON,Post Job Log	File Info	N/A
4	finished	0	T3_US_Princeton_ARM	[illegible]	[illegible]	[illegible]	[illegible]	00:09:55	Job Log,Job Log JSON,Post Job Log	File Info	N/A
5	finished	0	T3_US_Princeton_ARM	1	2015-06-08T20:01:22	2015-06-08T20:05:37	2015-06-08T20:15:34	00:09:57	Job Log,Job Log JSON,Post Job Log	File Info	N/A
6	finished	0	T3_US_Princeton_ARM	1	2015-06-08T20:01:22	2015-06-08T20:09:29	2015-06-08T20:16:00	00:06:31	Job Log,Job Log JSON,Post Job Log	File Info	N/A
7	finished	0	T3_US_Princeton_ARM	1	2015-06-08T20:01:22	2015-06-08T20:24:29	2015-06-08T20:28:52	00:04:23	Job Log,Job Log JSON,Post Job Log	File Info	N/A
8	finished	0	T3_US_Princeton_ARM	1	2015-06-08T20:01:22	2015-06-08T20:24:29	2015-06-08T20:29:03	00:04:34	Job Log,Job Log JSON,Post Job Log	File Info	N/A
9	finished	0	T3_US_Princeton_ARM	1	2015-06-08T20:01:22	2015-06-08T20:24:30	2015-06-08T20:29:32	00:05:02	Job Log,Job Log JSON,Post Job Log	File Info	N/A
10	finished	0	T3_US_Princeton_ARM	1	2015-06-08T20:01:22	2015-06-08T20:24:31	2015-06-08T20:28:10	00:03:39	Job Log,Job Log JSON,Post Job Log	File Info	N/A

The first AArch64 based WLCG site (demonstrator)



Heterogeneous computing: batch job submitted from **x86_64** machine at CERN to **AArch64** worker node at Princeton University

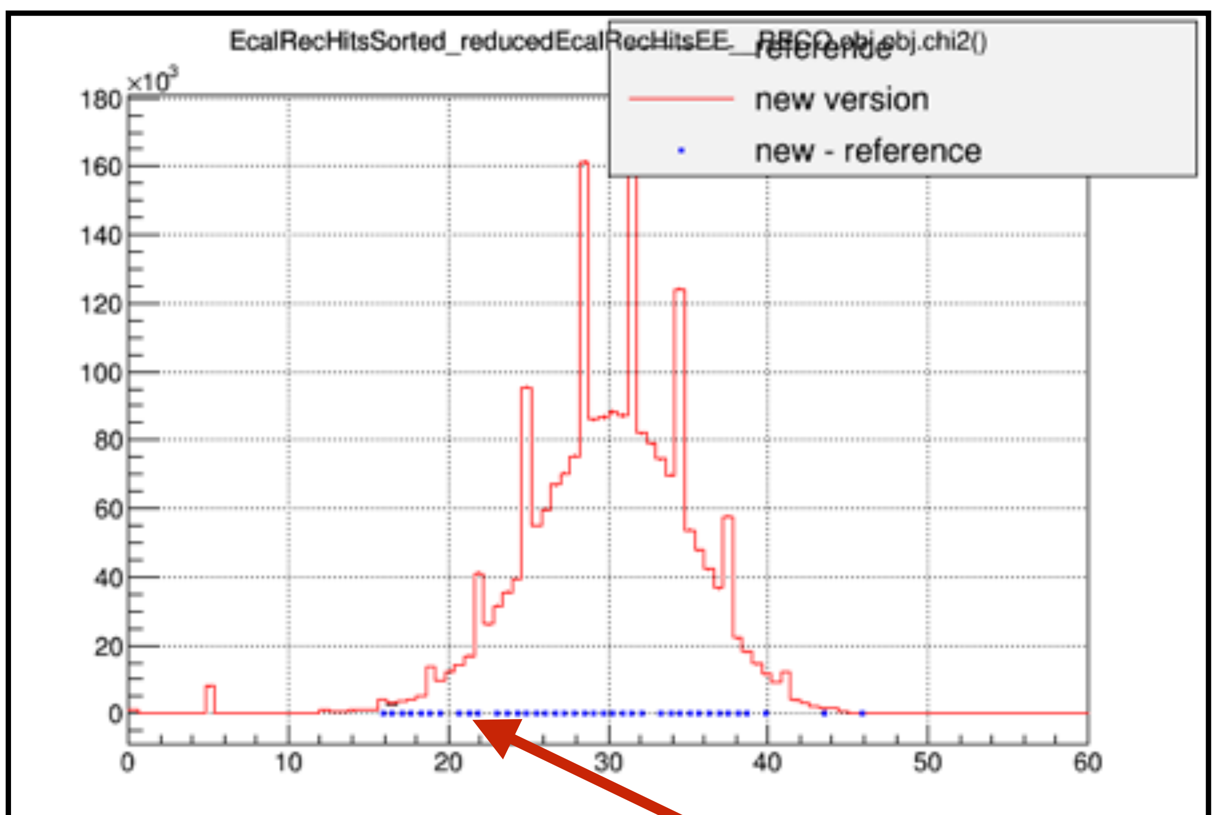
Showcased on **Fedora 19** on **APM Mustang** development board

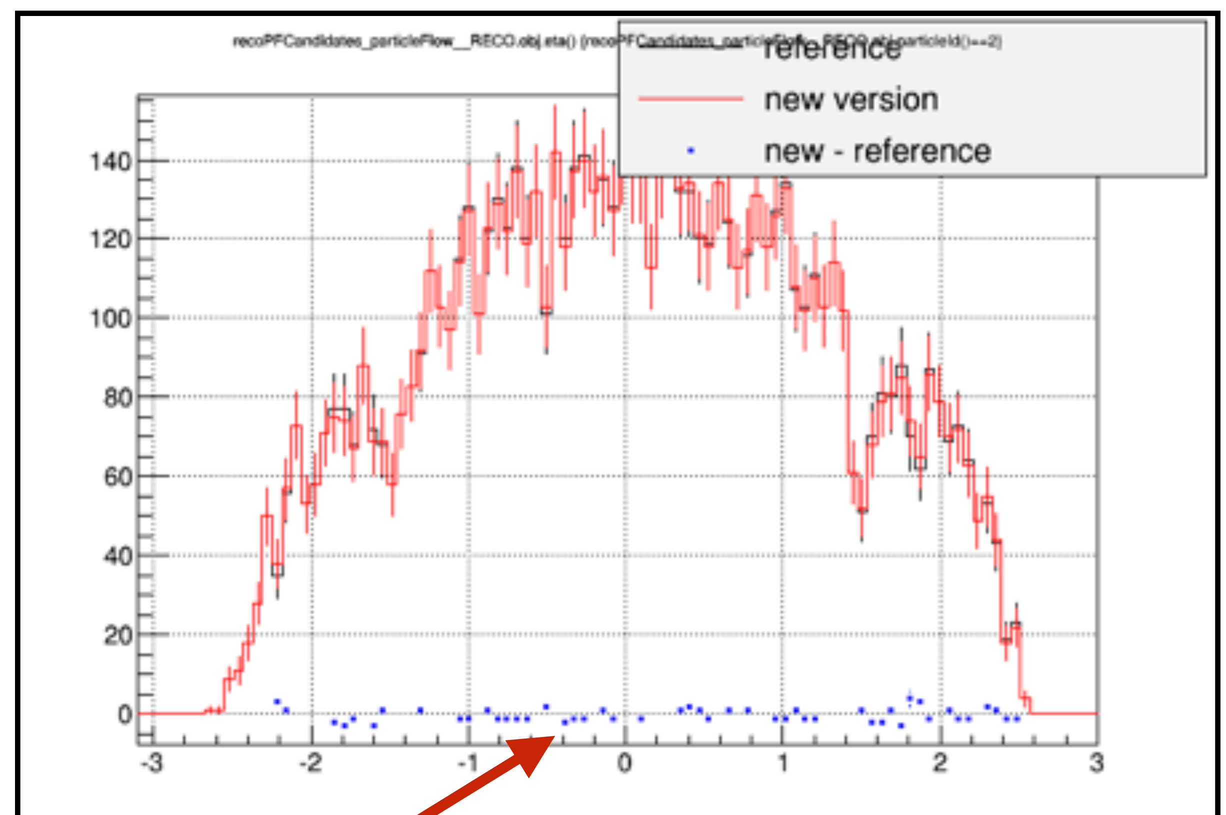
Moving to **CentOS 7.2** on **HP Moonshot** + 6 x **m400** (production-level system)

Numerical Validation (ARMv8)

We used CMSSW_7_2_0 pre-release for **x86_64** and **AArch64** reconstruction comparison, where input were generated on **x86_64.**

~950 differences were detected, but **majority were minimal**, i.e. non-significant; Examples:





Blue dots mark the difference

NEW!

What's new? #1

Official CMSSW Integration Builds (IBs) now include **aarch64** and **ppc64le**!

CMSSW_8_1_X_2016-02-21-0000

- IB Tag
- Static Analyzer
- Modules to thread unsafe statics
- Modules to thread unsafe EventSetup products
- HLT Validation
- Valgrind
- DQM Tests

Architectures	Builds	Unit Tests	RelVals	Other Tests	FWLite	Q/A
slc6_amd64_gcc493 Full Build	See Details	4 Tests Failing	Pass: 841 Fail: 12	See Details	See Details	
slc7_aarch64_gcc530 Full Build	26 Warnings	Unknown				
slc7_amd64_gcc530 Full Build	19 Warnings	5 Tests Failing	Pass: 1193 Fail: 18	See Details		
slc6_amd64_gcc530 Full Build	19 Warnings	3 Tests Failing	Pass: 1233 Fail: 3	See Details		
fc22_ppc64le_gcc530 Full Build	47 Warnings	32 Tests Failing	Pass: 1158 Fail: 30	See Details		

- No new pull requests since CMSSW_8_1_X_2016-02-19-2300

Missing tests!

Note: more work is needed to make everything stable

$ file /cvmfs/cms.cern.ch/{slc7_aarch64_gcc530,fc22_ppc64le_gcc530}/cms/cmssw/CMSSW_8_0_0
/cvmfs/cms.cern.ch/**slc7_aarch64_gcc530**/cms/cmssw/**CMSSW_8_0_0**: directory
/cvmfs/cms.cern.ch/**fc22_ppc64le_gcc530**/cms/cmssw/**CMSSW_8_0_0**: directory

Available for all sites now!

NEW!

What's new? #2

(1) CVMFS 2.3.0 (dev) is building on **CentOS 7.2/aarch64** since **Feb 14th** nightly build

- See "Technology Previews" under "CernVM-FS Downloads" page
- We are running CVMFS client under aarch64 since before ACAT '14 (September 2014) and have not observed issues
- Server to be tested once OverlayFS issues are solved (currently one needs aufs)

(2) Static **PRoot/QEMU** + CentOS/Fedora **rootfs** setup for doing non-native installations

(3) CMSSW port to **ppc64le (POWER8)** discovered 2 issues in LLVM (all resolved upstream)

(4) Attempt to for CMSSW **ppc64 (big-endian)** port revealed issues

- Bundled LLVM inside ROOT 6.06 is broken (waiting for move to 3.8.0)
- pyroot is not endian safe (patch WIP)

(5) Preparations for **Open Science Grid (OSG)** full (hopefully) repository rebuild for **aarch64** (will also require some EPEL packages to be built)

POWER8 **Very Early** Comparison

CMSSW reconstruction, Run II-like

No impact in performance

MT memory savings

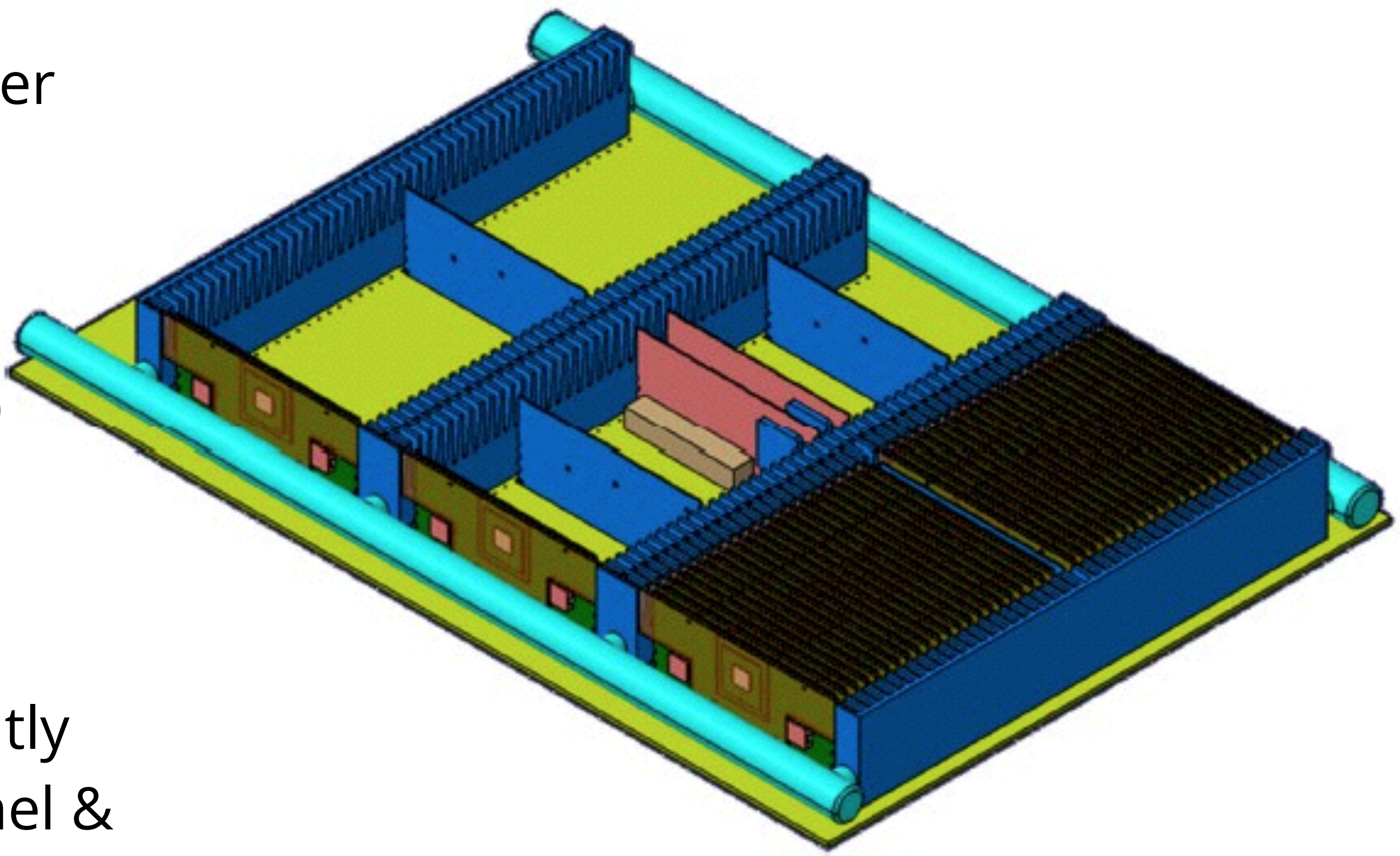
	2xIBM 8247-22L	2xHaswell E5-2699
# Physical core comparison (8 vs 2 threads/proc)		
Single thread (performance)	0.156907 ev/s	0.200261 ev/s
Multi threaded (performance)	0.155383 ev/s	0.198463 ev/s
Single thread (peak RSS)	15'190.5 MB	3'341.89 MB
Multi threaded (peak RSS)	3'145.62 MB	1'859.4 MB
# Full machine comparison (160 vs 72 threads) or (40 vs 18 4-threaded jobs)		
Multi threaded (performance)	2.78965 ev/s	3.65784 ev/s
Multi threaded (peak RSS)	97'844 MB	38'824.2 MB

Intel Xeon Haswell (E5-2699) provided **1.31x more events/s** compared to IBM POWER8 (8247-22L)

NEW!

Datacenter-in-a-box

IBM/ASTRON (in Zurich) DOME 64-bit μServer for SKA big data challenge

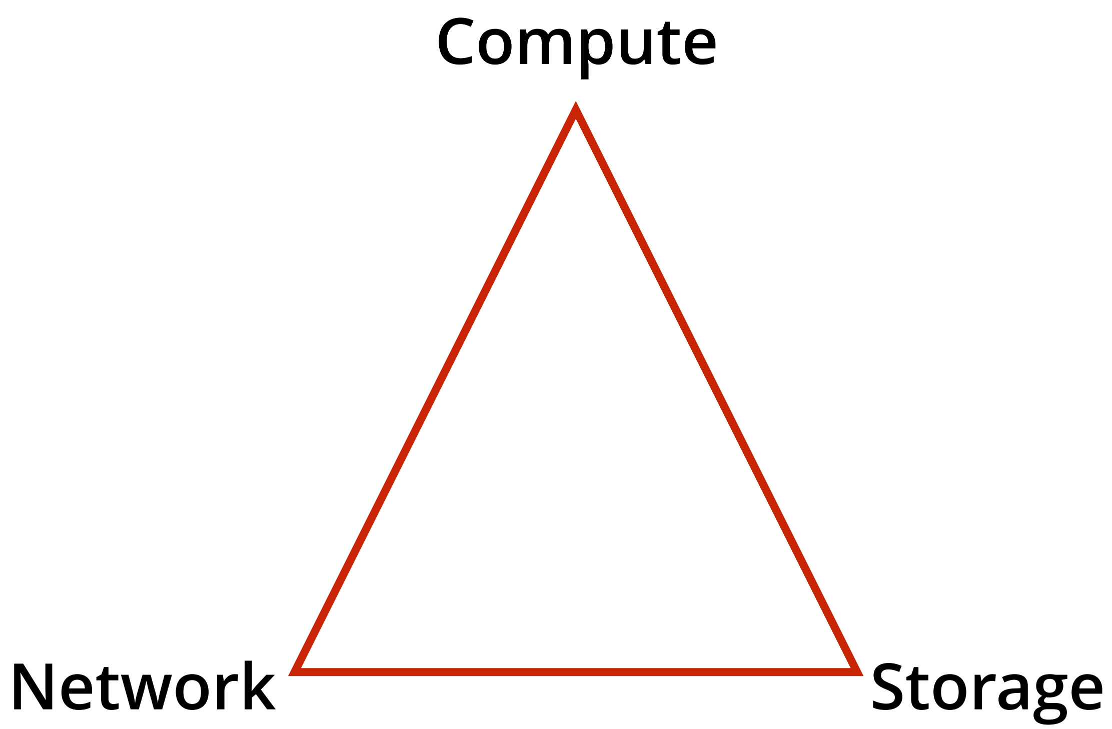


- 19" 2U w/ combined cooling & power
- 128 compute nodes
- **1536 ppc64** cores / **3074 threads**
- **6 TB DRAM**
- **1.28 Tbps** Ethernet (@40Gbps x 32)
- Expected total power is **~6kW**
- **Hot-water cooled for efficiency and density**
- Upstream support in future, currently runs Fedora rootfs + Freescale kernel & uboot
- **Memory bandwidth density:**
  DOME 128 nodes 2U: **159GB/s/Liter** (peak)
  POWER8 S822L (2S) **13.9GB/s/Liter** (peak)

It's all about SoC and packaging!
There will be **aarch64** version!

Motivation for porting **CMSSW** to **ppc64**

"Bring Balance to the Force"



Changing one (e.g. Compute) might disturb existing balance in the Force

Summary

- Power constraints and market evolution **may drive change in the kinds of processors we use**
- Application diversity could drive heterogeneity to aid in **{performance, power, cost}** optimizations
- The **race is heating up**, and **Intel/platform vendors are not sitting idle**
- We have been exploring **alternative general purpose architectures** to the current **x86_64** cores, incl. ARMv7 32-bit, ARMv8 64-bit, PowerPC (LE and BE), Xeon Phi
- We have demonstrated both application software (**CMSSW**) as well as job submission using **CRAB** (CMS Remote Analysis Builder) to **aarch64** nodes using a demonstrator cluster, and we will keep improving it
- We showed that **heterogeneity** by submitting jobs **from x86_64** machine and landing them **on aarch64** worker nodes
- We are involved with open source communities and industry partners
- We need to continue investigating new SoCs/CPUs and platforms (e.g. Xeon D, new ARMv8.{0,1} SoCs and platforms using better processes (i.e. 14/16nm FinFET)

Q&A

CONTACT

DAVIDLT AT CERN DOT CH

BACKUP

CPU Evolution

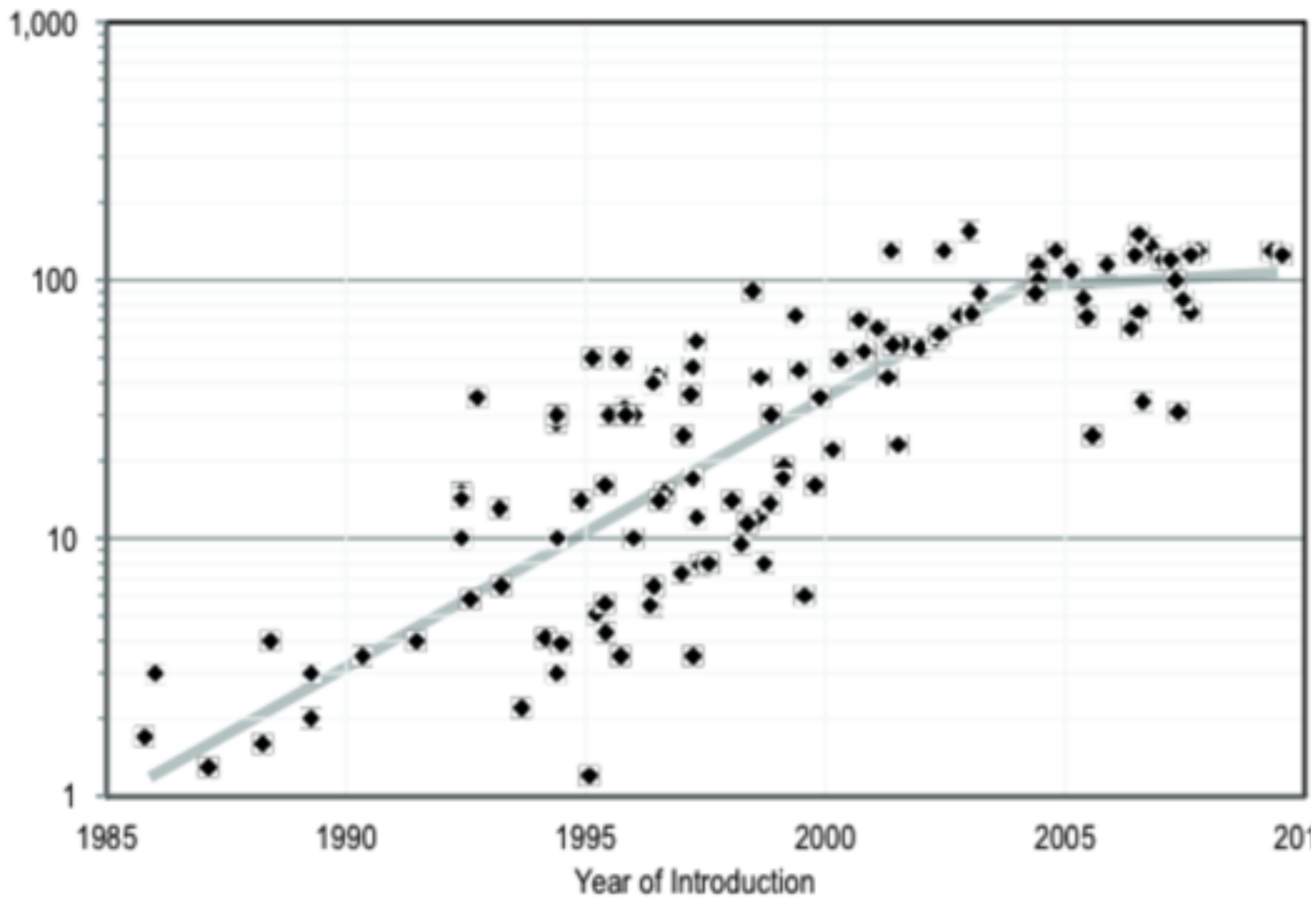
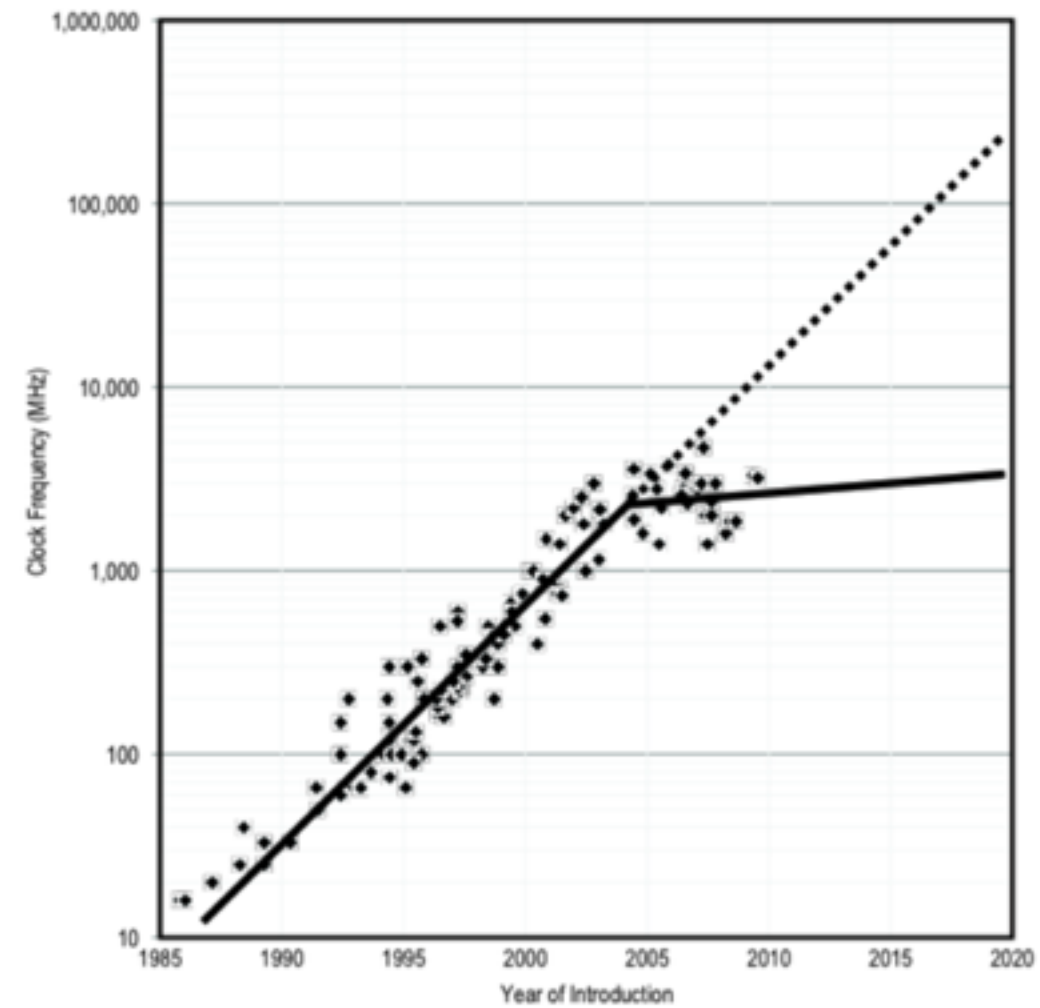


FIGURE A.4 Microprocessor power dissipation (watts) over time (1985-2010).



Source: "The Future of Computing Performance: Game Over or Next Level?"