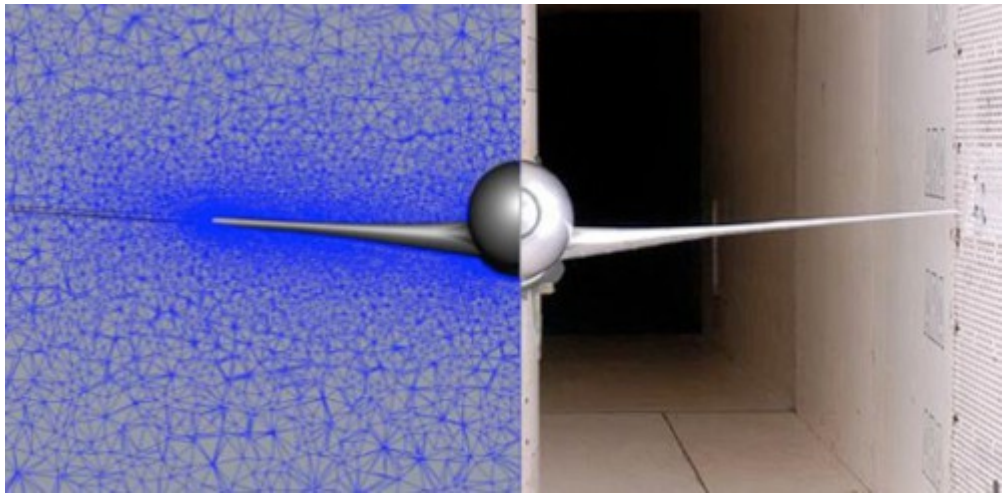
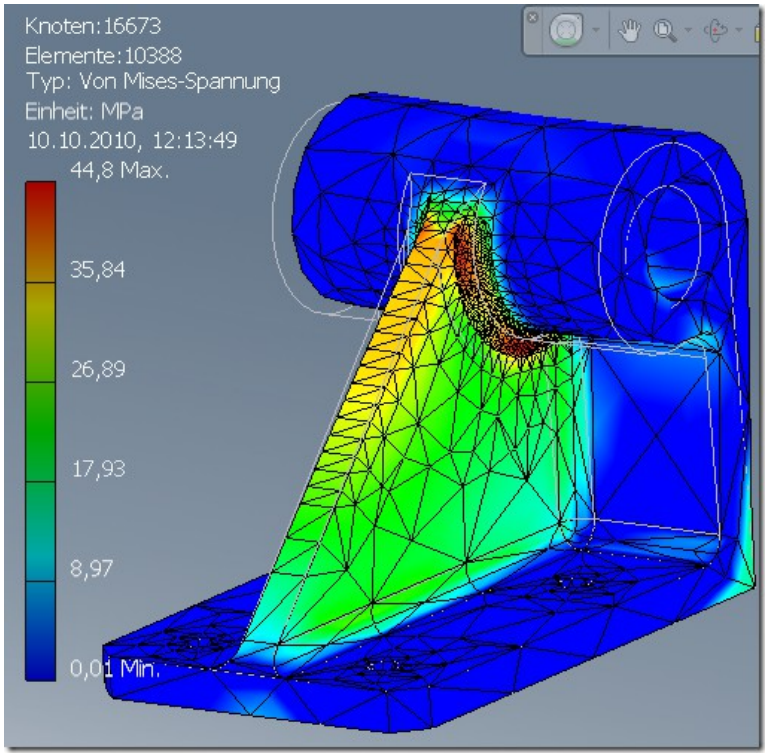
HPC PUPPET
&
EasyBuild

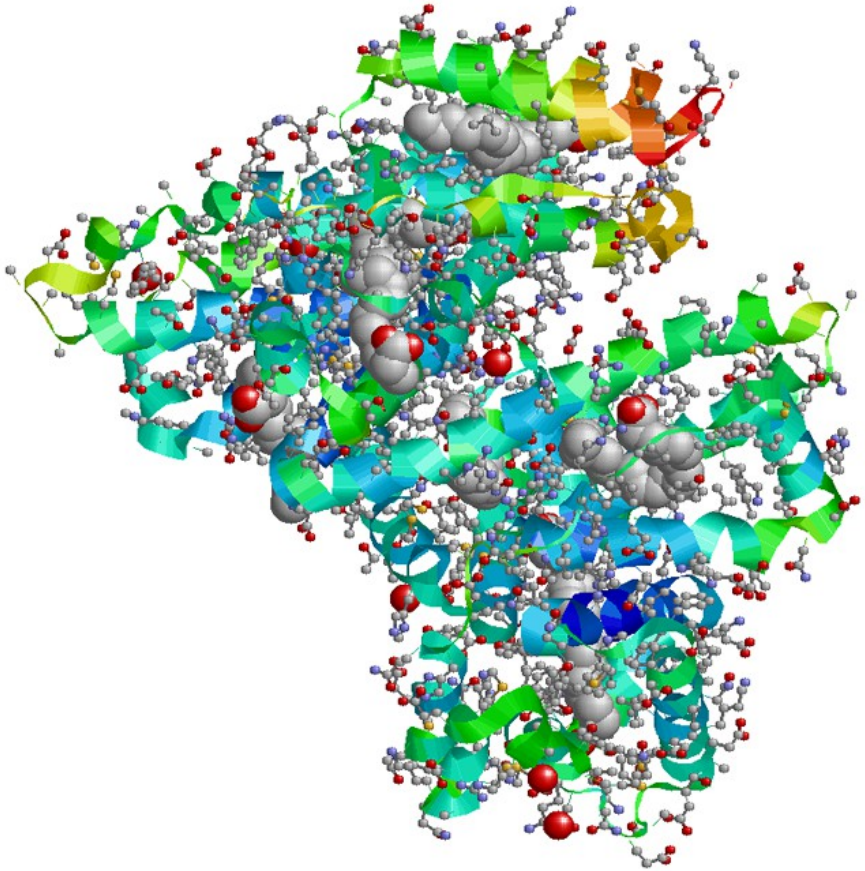
jure.pecar@arctur.si

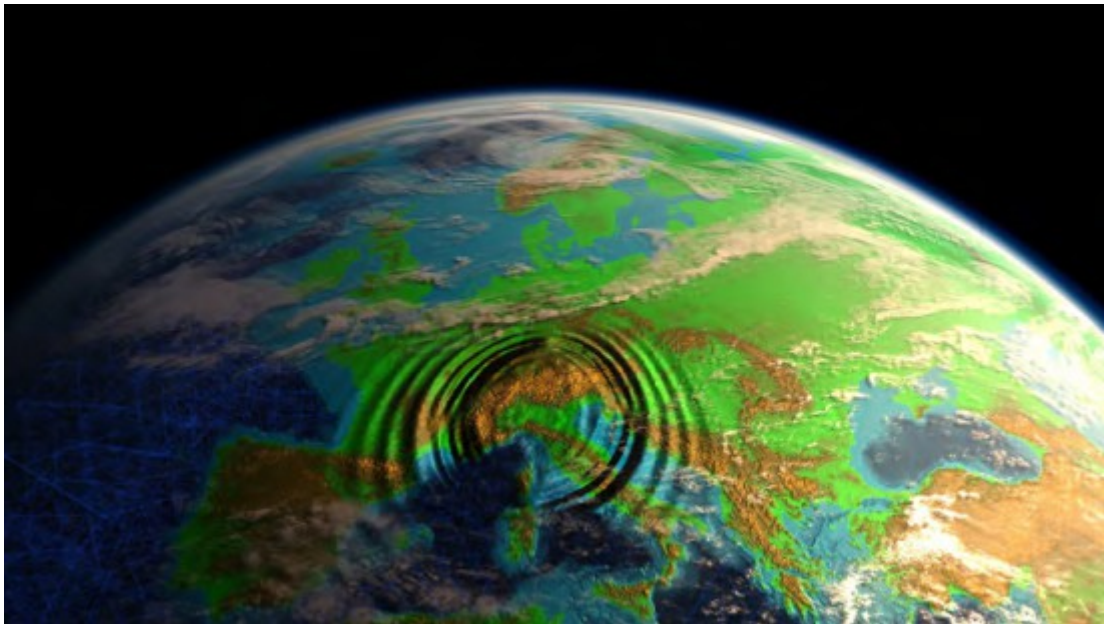
Ljubljana, 11.7.2014

HPC == orodje

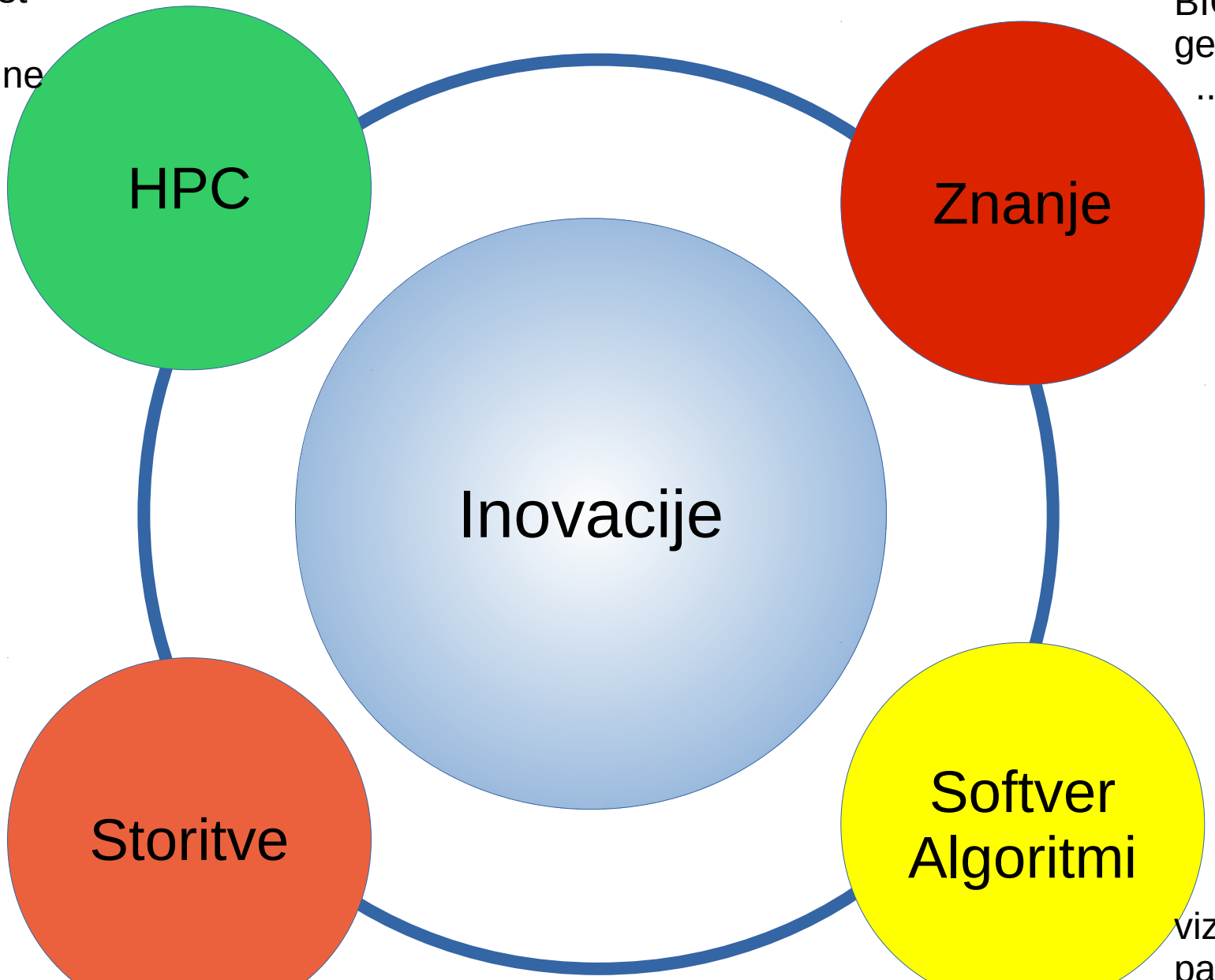



Knoten:16673
Elemente:10388
Typ: Von Mises-Spannung
Einheit: MPa
10.10.2010, 12:13:49
44,8 Max.
35,84
26,89
17,93
8,97
0,01 Min.





Sodelovanje



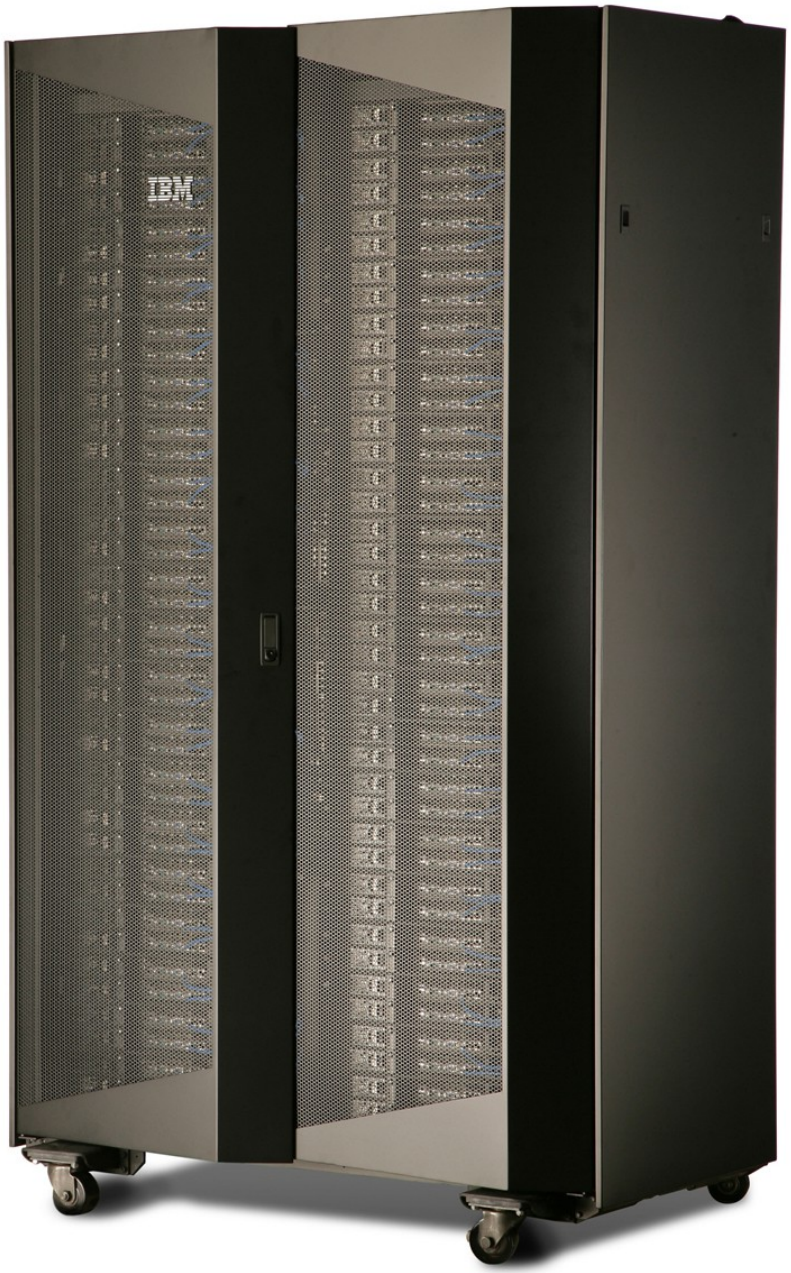
zmogljivost
skaliranje
komercialne
kode
...

FEM CFD
BIO GEO
genomika
...

svetovanje
analiza
strategija
partnerstva
dodana
vrednost
...

vizualizacija
paralelizacija
ozka grla
big data
...

HPC kot storitev



Kaj je HPC

- zelo tradicionalna (*staromodna*) oblika računalništva
  - batch processing
  - zelo malo interaktivnega dela
- kombinirana z najnovejšim hardverom
  - najnovejši CPUji
  - pospeševalniki (GPU, Phi)
  - komunikacije (infiniband, …)
- obsežna, hitra diskovna polja
- *več kot delovna postaja*

Kaj potrebuje uporabnik

- dostop
  - tradicionalno ssh in ukazna vrstica
  - vedno bolj pogost tudi grafičen dostop (vnc, nx)
- približno standardno okolje
  - shell, prevajalnik, mpi, batch sistem
- prostor za podatke
- softver
- zmožnost poganjanja jobov

Kaj pričakuje uporabnik

- pomoč pri uvajanju v delo
  - vzpostavitev okolja
  - izbira primernega middleware softvera
- pomoč pri prenosu večje količine podatkov
- vizualizacijo rezultatov
- pomoč pri prilagoditvi in optimizaciji lastne kode
- pomoč pri pogajanjih za sw licence

Ugotovitve:

- HPC je danes množica precej standardnih strežnikov
- vlogo “HPC” jim omogoči nameščen softver
- strežnike, ki niso zaposleni, lahko uporabimo v drugih vlogah

Motivacija za razvoj lastnega sistema za upravljanje strežnikov

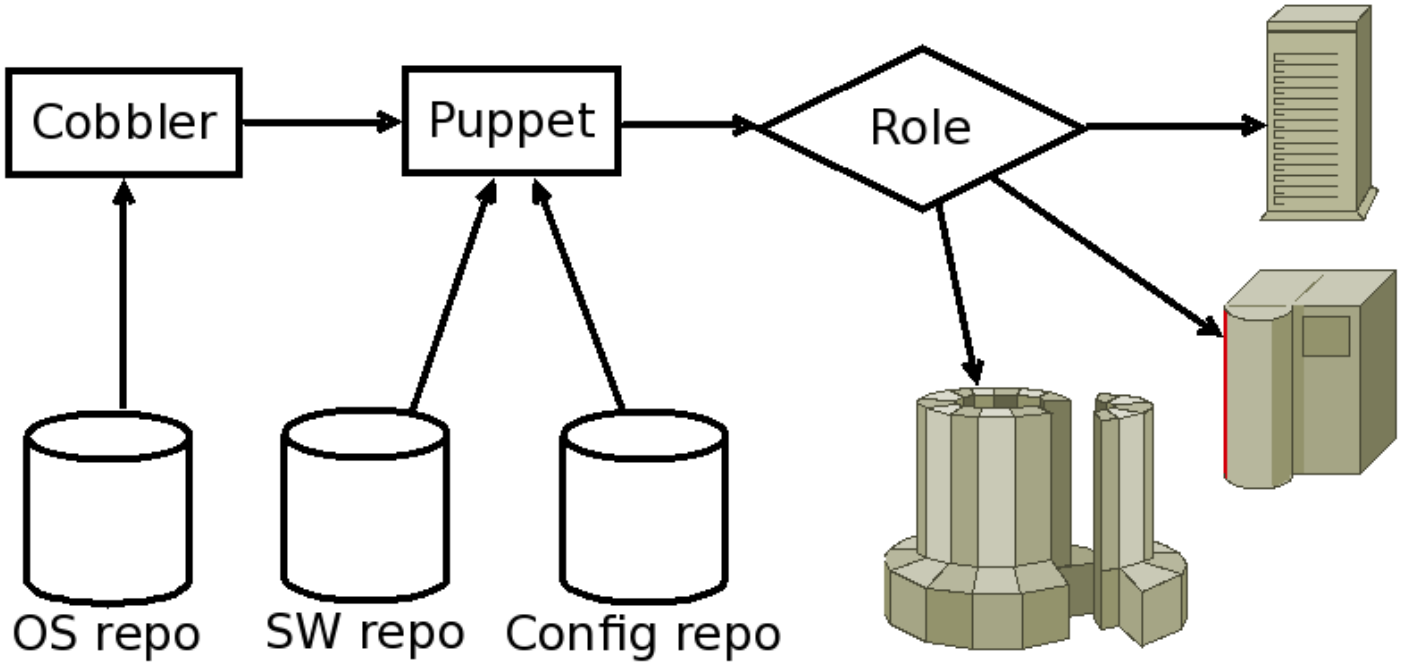
Puppet

- orodje za avtomatizacijo
  - eno izmed mnogih
- configuration management
  - v človeku in stroju(!) berljivi obliki opišeš stanje sistema
- DevOps
  - release early, release often ;)

Proces

- ugotovimo potrebne vloge v okolju
- opišemo te vloge
- dodelimo vloge posameznim strežnikom
  - vloga je vezana na hostname
- namestimo minimalni OS na strežnik
- poženemo puppet, ki poskrbi za vse ostalo

Puppet je ključni del celotne slike



Kaj pridobimo

- Z operativnega vidika
  - uniformnost
  - ponovljivost
  - interna standardizacija
- S poslovnega vidika
  - prilagodljivost
  - agilnost, fleksibilnost
  - čas za ukvarjanje z novostmi

HPC Puppet: cilji

- delujoča HPC gruča, sestavljena iz paketov v javno dostopnih repozitorijih
- okolje, skladno z IntelClusterReady zahtevami
- posvojitev sw best practices z implementacijo orodja EasyBuild
- vse potrebno za sodelovanje v Sling in EGI grid
- hiter čas namestitve posameznega strežnika

HPC Puppet

- implementira vse ključne elemente gruče:
  - login node
  - resource manager
  - compute nodes
  - /home nfs strežnik
  - /lustre medatada in object strežnike
  - podporne strežnike (ldap, sw build, …)
  - grid gateway
- objavljen pod GPL v2
- https://github.com/Arctur/hpc-puppet

Alternative

- Oscar – http://oscar.openclustergroup.org/
- Rocks - http://www.rocksclusters.org/
- Qlustar - https://www.qlustar.com/

- kar nekaj komercialnih

easybuild

- izdelek UGent
- motivacija: znanstveniki pišejo softver, ki rešuje njihove probleme in ne softver, ki je prijazen uporabniku
  - nestandardne build procedure
  - hardcoded parametri
  - neobstoječa dokumentacija
- nameščanje je časovno potratno, prihaja do napak, težave pri optimizaciji

Obstoječa orodja

easybuild

- Niso primerna (deb, rpm)
  - HPC ljudje prevajajo izvorno kodo
  - skripte za prevajanje je naporno vzdrževait
- ogromno podvajanja dela med HPC sistemi
- neprimerne rešitve
  - OS based (portage, homebrew)
  - App based (Dorsal – DOLFIN, gmkpack - Aladin)

Želje

easybuild

- fleksibilno ogrodje za znanstvene kode
- avtomatizirano prevajanje
- ponovljivost
- podpora za hkratnih več verzij istega softvera
- deljenje znanja znotraj HPC skupnosti
- avtomagično reševanje odvisnosti

Stanje

easybuild

- http://hpcugent.github.io/easybuild/
- python
- projekt začet 2009, inhouse skoraj 3 leta
- GPLv2 od 2012
- stable API z verzijo 1.0 (november 2012)
- zadnja verzija 1.13
- community-driven razvoj
  - več hackatonov letno
- nova verzija vsakih 4-6 tednov

Kaj zna

easybuild

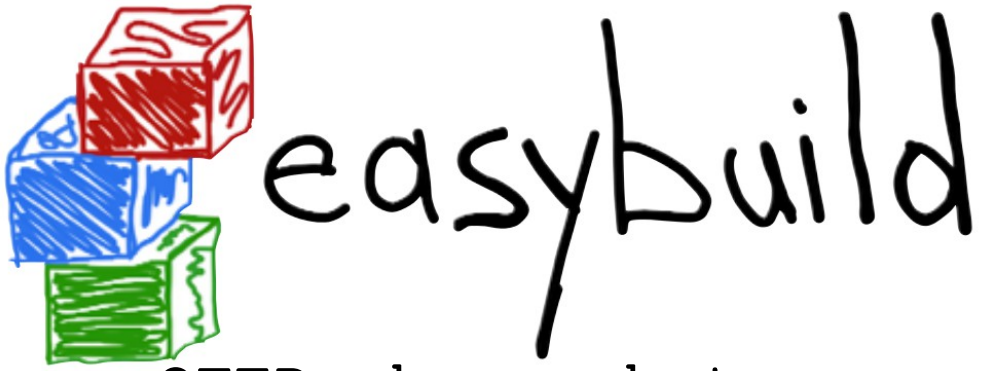
- samostojno prevesti softver
  - optimiziran za vaš hw
- zapisati dogajanje med prevajanjem in ga shraniti
- razrešiti odvisnosti
- prevajati vzporedno, tudi kot batch job
- stestirati namestitev
- v repozitorijih že podpora za 483 paketov s prek 2500 variacijami

Terminologija

easybuild

- ogrodje (framework)
  - jedro easybuilda, python moduli, eb wrapper script
  - ponuja podporne funkcije
- easyblock
  - implementira build proces za konkreten softver
  - lahko je generičen (configure && make && make install) ali specifičen
  - govori framework API
- easyconfig
  - tekstovna datoteka z navodili
  - podana kot parameter ukazu eb
- compiler toolchain
  - zbirka prevajalnika in MPI, BLAS, LAPACK, FFT knjižnic
  - C/C++/Fortran: GNU gcc, Intel icc

Primer




```
eb HPL-2.0-goolf-1.4.10-no-OFED.eb --robot
```

- z interneta prenese vse potrebne pakete
- prevede in namesti “goolf” toolchain: GCC, OpenMPI, LAPACK, OpenBLAS, FFTW, ScaLAPACK
- prevede HPL z goolf toolchainom
- sestavi modulefile za vsak kos softvera
- za uporabo: `module load ...`

Zaključek

- vzpostavitev gruče je kompleksno opravilo
  - za nekoga z izkušnjami na velikih strežniških okoljih precej domače
- s stališča uporabne vrednosti je čas, porabljen za vzpostavljane gruče, *jalovo delo*
- kot končni uporabnik se s tem nočete ukvarjati
- kot končni uporabnik hočete reševati probleme
- HPC mora biti obravnavan kot infrastruktura državnega pomena

PGAS

Partitioned Global Address Space

PGAS

- še en način programiranja vzporednih algoritmov
- začetki segajo v osemdeseta (Cray)
- v devetdesetih ga zasenči MPI
- danes spet aktualen zaradi vzpona NUMA in manycore arhitektur

	memory model	programmin model	execution model	data structures	communication
MPI	distributed memory	cooperating executables (often SPMD in practice)		manually fragmented	APIs
OpenMP	shared memory	global-view parallelism	shared memory, multithreaded	shared memory arrays	/
Global Arrays	PGAS	Single Program, Multiple Data (SPMD)		global-view distributed arrays	implicit via APIs
CoArray Fortran, UPC, Titanium				co-arrays 1D block-cyc arrays/distributed pointers, class-based arrays/distributed pointers	co-array refs implicit method-based
Chapel	Asynchronous PGAS	global-view parallelism	distributed memory multithreaded	global-view distributed arrays	implicit
X10					visible in syntax

PGAS - motivacija

- poenostaviti programiranje
  - OpenMP je enostavnejši kot MPI
  - kako OpenMP enostavnost razširiti čez celo gručo?
- implementacije prinašajo zanimivo dodano vrednost
  - enostavnejšo implementacijo odpornosti na hw napake

PGAS - aktivnosti

- OpenSHMEM
- UPC, UPC++
  - gcc in clang
- Chapel, X10, resilient X10
- XcalableMP
- Phalanx (by Nvidia)
- Grappa
- TAU
- GPI-2
- GASNet
- ...

PGAS – današnje omejitve

- sw stack vnaša latence
- želja po več memory iops
- relativna visoka cena remote operacij
- zato relativno drago skaliranje na velikih sistemih

PGAS – namesto zaključka

- še eno žarišče akademskih aktivnosti
- ideje počasi prehajajo tudi v MPI
- prvi poskusi z realnimi aplikacijami – GROMACS zmanjšal obseg komunikacije za 50%

Hvala za pozornost

jure.pecar@arctur.si

Ljubljana, 11.7.2014