

# Cactus Code

-Lefter Dragoș Constantin SRIC-

**Abstract** — În această lucrare este prezentat un framework pentru paralelizarea aplicațiilor utilizat în general în proiecte foarte complexe de cercetare sau inginerie. Va fi prezentată arhitectura Cactus și de asemenea câteva din cele mai importante proiecte.

**Termeni de bază** — paralelism, cactus, cod, thorn, Einstein Toolkit

## I. INTRODUCERE

Cactus este un mediu open source pentru rezolvarea unor probleme de inginerie sau cercetare. Structura sa modulară permite cu ușurință calculul paralel pe diferite arhitecturi și dezvoltarea de cod între diferite grupuri. Cactus își are originea în comunitatea de cercetare academică, unde a fost dezvoltat și folosit de foarte muți ani, bazat pe o colaborare internațională între fizicieni și oameni de știință.

Numele Cactus provine de la modelul unui nucleu central (flesh), care se conectează la modulele aplicației (thorn) printr-o interfață extensibilă. Spini pot implementa aplicații dezvoltate personalizat de oamenii de știință sau ingineri, cum ar fi dinamica computerizată a fluidelor. Alte ramuri dintr-un set de instrumente de calcul standard oferă o gamă de capabilități de calcul, cum ar fi sisteme de intrare/ieșire paralele, de distribuție de date, sau checkpointing.

Cactus rulează pe mai multe arhitecturi, aplicațiile dezvoltate pe stațiile de lucru standard sau laptop-uri, putând fi rulate fără probleme pe clustere sau supercomputere. Cactus oferă acces ușor la multe tehnologii de ultimă generație fiind dezvoltat în comunitatea de cercetare academică, incluzând Metacomputing Globus Toolkit, fișierul paralel de intrare/ieșire HDF5, biblioteca științifică PETSc, interfață web, și instrumente avansate de vizualizare.

Comunitatea utilizatorilor Cactus a creat și se ocupă de menținerea instrumentelor pentru mai multe domenii de cercetare. Ca exemplu, Einstein Toolkit se referă la calculul în astrofizică relativistă, sprijinirea simulării de găuri negre, stele neutronice, și a sistemelor aferente.

## II. LICENȚA

Codul Cactus este foarte ușor de portat existând totuși câteva cerințe de bază și fiind necesar utilizarea unor utilitare ca gmake sau Perl și a unui compilator ANSI C/C++ pentru construirea structurii. Ramurile de bază pentru calcul sunt, în principal scrise în C (cu câteva mici excepții în C++). Unele aplicații necesită, de asemenea, un compilator Fortran 90 deși F77 ar fi de ajuns pentru rulare. Driverile existente în pachetul actual folosesc Message Passing Interface (MPI) pentru paralelizarea codului pe mai multe procesoare sau mașini, ambele implementari nativ și free (MPICH, LAM) fiind acceptate.

Codul programului Cactus este protejat prin copyright de către autori individuali și distribuit sub GNU Lesser General Public License. Cele mai multe dintre pachetele de calcul care vin cu programul urmează această licență, dar autorii oricărei părți din program sunt liberi să urmeze orice tip de licențiere pe care o consideră adecvată, atât timp cât o prezintă în mod explicit în distribuția lor.

## III. PROIECTE DE BAZĂ

În figura 1 sunt prezentate o serie de proiecte construite cu ajutorul Cactus

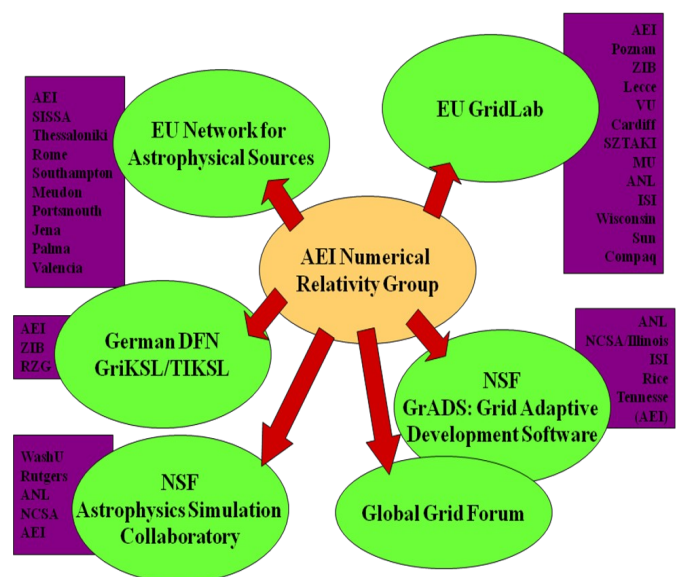


Figura 1. Proiecte dezvoltate pe baza Cactus

Pe lângă proiectele prezentate mai sunt există și o serie de proiecte ce sunt construite pe infrastructura oferită de Cactus și au la bază acest framework:

a) **Alpaca** dezvoltă instrumente Cactus pentru profiling la nivel de aplicație și analiza corectitudinii. Proiectul Alpaca dezvoltă instrumente de nivel înalt, care ajută oamenii de știință să creeze și să mențină aplicații mari, complexe. Instrumentele Alpaca permit dezvoltatorilor și utilizatorilor finali să examineze și să valideze corectitudinea unei aplicații, ajutându-i foarte mult la măsurarea și îmbunătățirea performanțelor sale în mediile de producție. Alpaca se bazează pe Cactus, iar instrumentele sale sunt componente ale framework-ului, sunt construite în aplicație și interacționează cu ea. Abordarea Alpaca include aplicații tolerante la defecte, ceea ce devine o nevoie necesară având în vedere miile de noduri existente în aplicații în zilele noastre.

b) **XiRel** dezvoltă o infrastructură de generație următoare pentru relativitatea numerică. Acest proiect creează un strat sub forma unei rețele foarte precise, eficiente și scalabile la nivelul framework-ului, bazată pe driverul Cactus, și optimizată pentru a suporta studierea relativității numerice, studierea găurilor negre, a stelelor neutronice și a undelor gravitaționale. Instrumente noi vor fi distribuite făcând parte din Einstein Toolkit, oferind un cod gratis pentru simulări de găuri negre și încorporând mecanisme pentru a încuraja partajarea de cod, de verificare, și validare a acestuia.

XiRel va crea baza pentru o cyberinfrastructură modernă, pentru comunitatea ce studiază relativitatea numerelor la nivel mondial, care să permită furnizarea de software pentru scrierea de cod scalabil. Dezvoltarea framework-ului Cactus va face aceste noi capacități disponibile pentru multe alte discipline, și va integra proiecte de aplicații în curs de desfășurare axate pe cercetarea informatică, rețele optice, și programarea datelor.

#### IV. ARHITECTURA

Cactus a fost construit pentru a fi orientat obiecte și foarte extensibil. Partea centrală a codului se numește *flesh*, iar extensiile sale se numesc thors (spini). Partea centrală este cea care controlează

execuția programului, userii fiind cei care specifică parametrii de rulare și fișierele de intrare. Acest proces este foarte dinamic, permițând userilor să schimbe unii parametri și părți din cond în timpul rulării codului.

În figura de mai jos este prezentată arhitectura generală a Cactus[2]

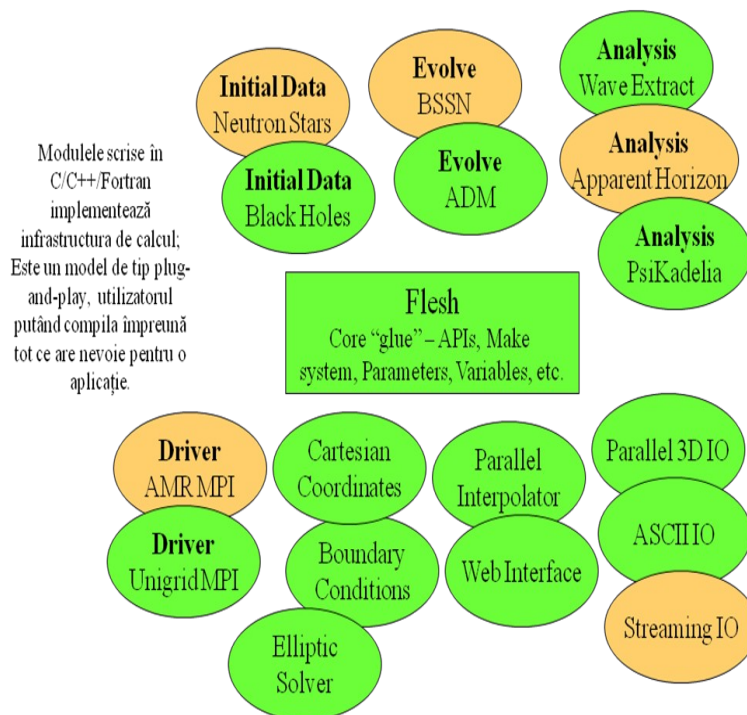


Figura 2. Arhitectura generală Cactus

O listă standard a modulelor toolkit este disponibilă pe site-ul celor de la cactus printre care putem enumera[1]:

- **Computational Toolkit:** module pentru capacități standard cum ar fi utilități pentru intrare/ieșire în formate și direcții de calcul diferite.
- **HDF5 Toolkit:** conține module necesare parametrilor de intrare/ieșire în formatul HDF5.
- **PETSc Toolkit:** module pentru intercațiunea cu bibliotecile PETSc pentru rezolvarea ecuațiilor de tip eliptic.
- **Benchmarking Toolkit:** oferă o aplicație pentru benchmarking (rezolvarea problemelor legate de undele gravitaționale).
- **Web Browser Toolkit:** oferă module ce se comportă ca un server web. Utilizatorii realizează calculele prin intermediul cererilor HTTP. Fluxurile de ieșire rezultate în urma rulării de aplicații pot fi afișate ca imagini în format JPEG.

- Einstein Toolkit: simulare găuri negre și stele neutronice.

În figura de mai jos este prezentată arhitectura pe nivele a Cactus[2]

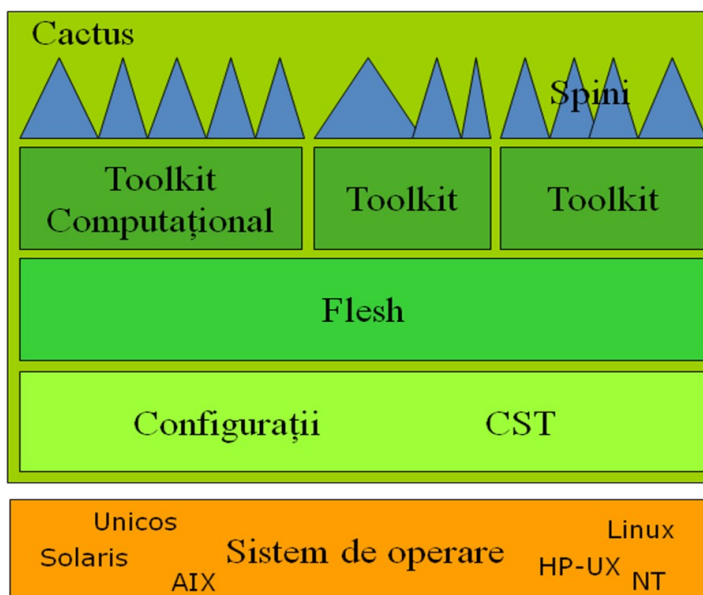


Figura 3. Arhitectura pe nivele a Cactus

Fișiere de configurare ale unui modul:

Fiecare modul conține trei fișiere de configurare (ccl) care descriu interfața modulului cu partea central (flesh) și alte module[3].

- interface.ccl – ce conține un header unde se găsesc detaliile despre relațiile între module, un bloc ce include fișierele folosite din alte module și cele oferite de modulul în cauză, detalierea funcțiilor alias furnizate sau utilizate de acest modul și o serie de blocuri ce conțin variabilele globale.
- param.ccl - constă într-o listă de itemi care specifică parametrii obiectelor separați de parametrii opționali ce detaliază accesul.
- schedule.ccl - instrucțiuni de atribuire și de stocare pentru variabilele de rețea pentru întreaga durată de execuție a programului; blocuri de programare pentru a programa o subrutină a unui modul să fie apelată la anumite ore în timpul programului
- configuration.ccl (optional)

Aceste fișiere sunt parsate (Perl) în timpul compilării de către CST (Cactus Specification Tool), și se generează liste de argumente (Fortran!), liste de parametrii și de rulare etc.

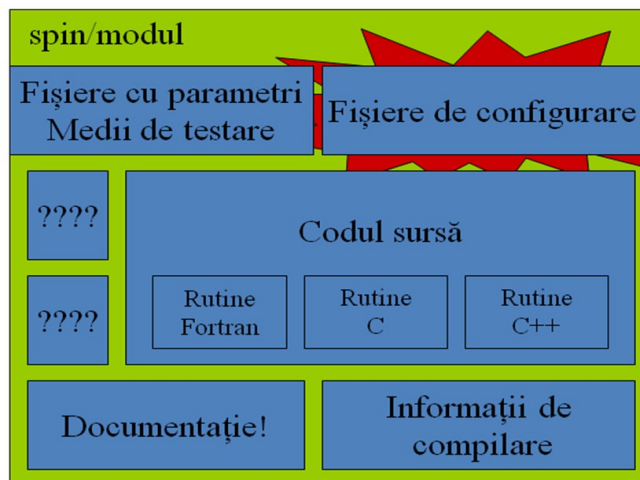


Figura 4. Arhitectura unui modul Cactus[2]

În ceea ce privește portaluri colaborative și de calcul, Cactus este o unealtă bazată pe cod, oferind o modalitate ce permite distribuirea de programe individuale printr-o metodă bazată pe web. Utilizatorii trebuie să aibă acces la codul sursă al aplicațiilor lor pentru a adăuga apeluri către metodele de Cactus.

Cactus oferă următoarele capacități de colaborare:

- O metodă care acționează ca un server web oferind acces la aplicațiile ce rulează.
- Multiplii participanți se pot conecta la acest server cu browserele lor. Sesiunea este protejată prin parolă.
  - Participanții pot controla aplicația prin schimbarea valorilor parametrilor de intrare. Acesta este un nivel ce lucrează cu cereri HTTP.
  - Cactus checkpointing și funcțiile intrare/ieșire permit codului să prezinte datele de ieșire ca un flux jpeg, astfel încât toți participanții la sesiune pot vizualiza datele de ieșire în browserele lor.

## V. RULAREA PROGRAMELOR

După scrierea programului urmează, pentru rulare se urmează pașii de mai jos:

1. Configure: se setează informațiile dependente de mașină și compilator.
2. Build: se crează codul din fișierul de configurare a modulului. Se trece prin fiecare modul și se crează biblioteci obiect apoi se leagă între ele.
3. Se crează o listă de module (ThornList) pentru executabile: isco.th
4. Se alege un nume de configuratie
5. Configure:
 

```
gmake isco-config THORNLIST=isco.th
```

```
gmake isco-config THORNLIST=isco.th
MPI=NATIVE
```

6. Make:
 

```
gmake isco
gmake isco WARN=yes TJOBS=4
```
7. Executabilul: exe/cactus\_isco

După obținerea executabilului rularea se face în felul următor:

- Pentru un singur procesor
 

```
./exe/cactus_isco MyParFiles/isco.par
```
- Pentru mai multe procesoare (compilat cu MPI)
 

```
mpirun -np 8 ./exe/cactus_isco
MyParFiles/isco.par
```
- Alte opțiuni de rulare
 

```
./exe/cactus_isco -help
```

## VI. CONCLUZII

Deși viteza și performanța calculatoarelor high-end au crescut dramatic în ultimii zece ani, ușurința de programare pe structuri paralele nu a progresat foarte mult. Este nevoie de timp și efort pentru a dezvolta și depăna software-ul necesar iar în domeniul științific și ingineriei timpul este unul din principalii factori pentru reușită. Dificultatea de a dezvolta software de înaltă calitate este recunoscută ca fiind una dintre provocările cele mai importante astăzi în utilizarea eficientă a calculatoarelor pe scară largă.

În aceasta lucrare am prezentat o parte din capacitățile pe care Cactus le are făcând din acest software unul din produsele care ușurează foarte mult programarea pe structuri paralele.

## REFERINȚE

- [1] Cactus Portal from Potsdam  
<http://aspen.ucs.indiana.edu/collabtools/extras/Cactus.doc>
- [2] Gabrielle Allen și Thomas Radke, Max Planck Institute for Gravitational Physics, (Albert Einstein Institute)  
[www.aei.mpg.de/~koppitz/proceedings/allen/alk.ppt](http://www.aei.mpg.de/~koppitz/proceedings/allen/talk.ppt)
- [3] Cactus User Guide  
<http://cactuscode.org/documentation/UsersGuide.pdf>
- [4] Alpaca project  
<http://cactuscode.org/community/projects/alpaca/doc/CactusCorrectness.pdf>
- [5] Reference Manual  
<http://cactuscode.org/documentation/ReferenceManual.pdf>

- [6] Gabrielle Allen, Ed Seidel, Peter Diener, Erik Schnetter, Christian Ott - The Cactus Framework & Numerical Relativity  
[http://www.cct.lsu.edu/~gallen/Presentations/ORNL\\_Nov06.pdf](http://www.cct.lsu.edu/~gallen/Presentations/ORNL_Nov06.pdf)