



**Grid aprēķinu vide**  
teorija • metodes • aprēķini



# NORDUGRID ARC STARPPROGRAMMATŪRA

PROGRAMMATŪRAS UZSTĀDĪŠANA UN UZSKAŅOŠANA

---

Dokumenta faila vārds:	<b>LG-ARC_v1</b>
Aktivitāte:	<b>1.aktivitāte "Grid klāsteru apvienošana vienotā un optimizētā tīklā"</b>
Projekta numurs:	<b>VPD1/ERAF/CFLA/05/APK/2.5.1./000055/027</b>
Organizācija:	Latvijas Universitātes aģentūra "Latvijas Universitātes Matemātikas un informātikas institūts"
Autori:	Baiba Kaškina, Kaspars Krampis, Bruno Martuzāns, Leo Trukšāns, Katrīna Sataki

---

## Anotācija:

Šajā dokumentā ir aprakstīti pētījumi, kas veikti, lai LU Linux centrā uzstādītu un uzskatītu Arc starpprogrammatūru. Darba gaitā izpētīta NorduGrid starpprogrammatūras darbība un komponentes, uzstādīta Arc starpprogrammatūra un Condor pakešdatu apstrādes sistēma. Apskatīta arī Condor praktiskā darbība un savietojamības iespējas ar citiem Grid klāstriem Latvijā.

---

## Saturs

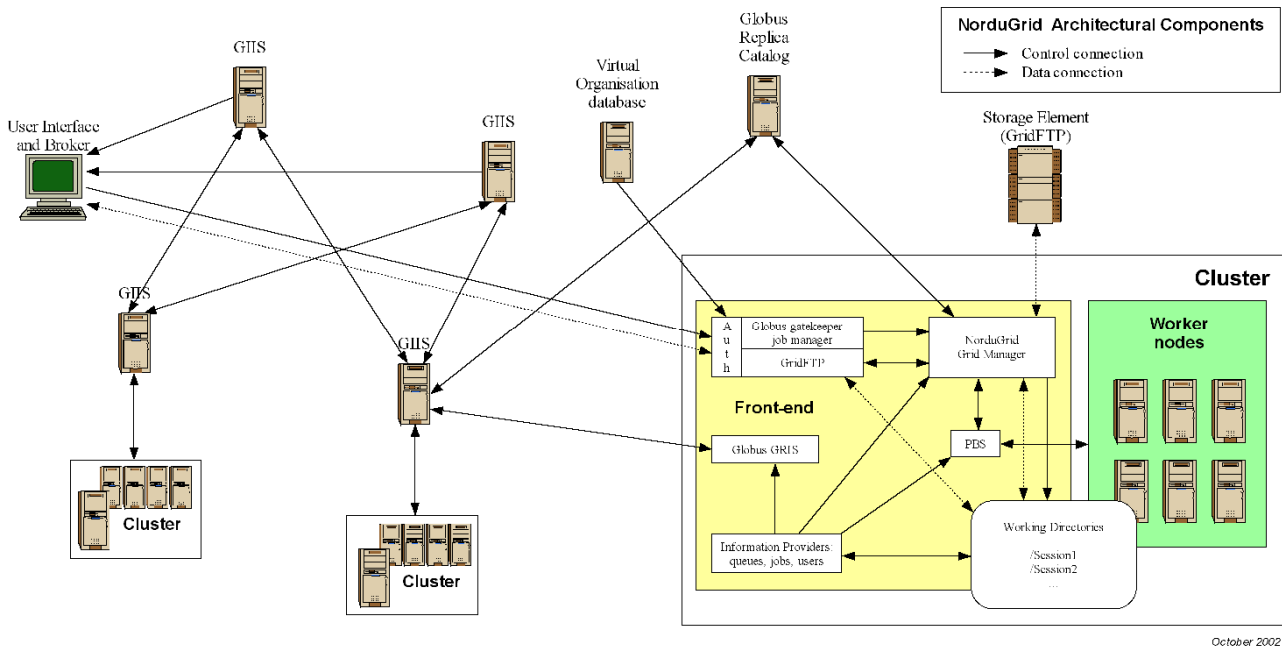
1. IEVADS .....	3
2. ARC UZSTĀDĪŠANA (TEHNISKAIS PROCESS) .....	6
3. ARC UZSKAŅOŠANA .....	9
4. CONDOR PAKEŠDATU APSTRĀDES SISTĒMA .....	11
5. CONDOR DARBĪBĀ .....	14
6. NOBEIGUMS .....	16
7. AVOTI .....	17

## 1. IEVADS

Lai spētu apstrādāt lielus apjomus ar datiem, kas rodas zinātnieku pētījumu rezultātā tika radīts *NorduGrid* projekts, saukts arī par *Nordic Testbed for Wide Area Computing*. Projekta mērķis bija radīt Grid infrastruktūru ziemeļvalstīs (Dānijā, Norvēģijā, Zviedrijā un Somijā), kas kalpotu kā platforma zinātniekiem apjomīgu skaitļošanas problēmu risināšanai un Grid rīku novērtēšanai [1]. *NorduGrid* filozofija ir pēc iespējas izmantot esošus rīkus un nepieciešamības gadījumā izveidot papildinājumus, kuri nodrošinātu trūkstošo funkcionalitāti.

*NorduGrid* sistēma sastāv no vairākām komponentēm, kas sadarbojas savā starpā:

- **User interface** – viena no galvenajām komponentēm, kas nodrošina pieejamo resursu apzināšanu, uzdevumu iesniegšanu un to statusa pārbaudi;
- **Information system** – kalpo kā informācijas sniedzējs citām komponentēm. IS sastāv no atsevišķām datubāzēm, kas ir savienotas gan ar skaitļošanas, gan datu glabāšanas resursiem, nodrošinot informāciju par to statusu;
- **Computing cluster** – tā ir skaitļošanas vienība, kas sastāv no priekšgala (*front-end*) mezgla un aizmugures (*back-end*) mezgliem, kas parasti savienoti slēgtā iekšējā tīklā. Programmatūras līmenī šī komponente sastāv no pakešdatu apstrādes sistēmas (*batch system*) un papildu slāņa saskarnei ar Grid. *NorduGrid* starpprogrammatūru nepieciešams uzstādīt tikai uz priekšgala datora;
- **“Smart” Storage element (SSE)** – tiek plānota kā autonomas serviss, kas nodrošinātu pamatfunkcionalitāti darbam ar datiem bez lietotāja iejaukšanās. Galvenais SSE mērķis ir kalpot kā daļai no pilnas datu glabāšanas sistēmas [2];
- **Replica catalog** – komponente tiek izmantota datu avotu reģistrēšanai un apzināšanai.



### 1. att. Nordugrid komponentes

Tā kā pārējās komponentes vairāki skaitļošanas resursi var izmantot kopīgi, galvenā uzmanība tika veltīta *Computing Cluster* jeb klastera uzstādīšanai. Nordugrid projektā kā starpprogrammatūra, kas veido komunikāciju starp tīkla komponentēm un arī lietotāju, tiek izmantots produkts ar nosaukumu ARC (*Advanced Resource Connector*).

Sākotnējais uzdevums tika formulēts šādi - uzstādīt *Nordugrid* ARC klasteri LU *Linux* Centrā (turpmāk LC).

Ikdienā LC datorklase tiek izmantota akadēmiskā darba vajadzībām, proti, tajā tiek vadītas lekcijas, praktiskie darbi, tā ir brīvi pieejama studentiem mājasdarbu pildīšanai vai personīgām vajadzībām. Klastera izveidošanas priekšnosacījums – saglabāt klases esošo funkcionalitāti un neizmantotos resursus novirzīt Grid rēķiniem.

LC datorklases datoriem tiek izmantota *Ubuntu Linux* operētājsistēma. Tā kā Ubuntu reizi 6 mēnešos iznāk jaunāka versija, klastera izveidei nevajadzētu ierobežot sistēmas atjaunināšanas iespējas, t.i., Grid vajadzībām nepieciešamai programmatūrai jābūt neatkarīgai no konkrētas operētājsistēmas (*Linux* distributīva).

Klastera infrastruktūra iedalāma divās galvenajās vienībās:

- tā saucamais priekšgals (*front-end*), kurš sastāv no viena datora – tas sazināsies gan ar iekšējā tīklā esošajiem datoriem, gan ar ārējiem resursiem;
- uz šī datora jāuzstāda ARC starpprogrammatūra;

- 
- tā saucamais aizmugures mezgls, kurš sastāv no skaitļošanas vai datu resursiem – šie datori sazināsies tikai savā starpā vai ar galveno datoru;
    - uz katra no šajā grupā esošajiem datoriem jāuzstāda pakešdatu apstrādes sistēmas (*batch system*).

Sākotnēji ARC programmatūru tika mēģināts uzstādīt uz *Ubuntu Linux 6.06*, tomēr, tā kā ARC tiek izstrādāts uz RPM pakotņu bāzētiem distributīviem (*Fedora Linux, RedHat Linux, SUSE Linux*, u.c.), radās nesavietojamības problēmas. Turklāt, ARC tiek izplatīts, vai nu izmantojot pirmkodu vai .rpm pakotņu veidā. Lai tās uzstādītu uz Ubuntu Linux (kurš ir bāzēts uz Debian Linux), rpm pakotnes jāpārveido par .deb pakotnēm. Tomēr ne vienmēr visa .rpm pakotnēs iestrādātā funkcionalitāte tiek korekti pārveidota uz .deb pakotni. Pēc vairākiem neveiksmīgiem mēģinājumiem tika secināts, ka *Ubuntu Linux* nav piemērotākais distributīvs ARC uzstādīšanai. Tā kā priekšgala datora pamatfunkcija paredzēta Grid klastera pārvaldīšana, tad šis dators netiks izmantots citiem mērķiem, tādējādi tam iespējams izvēlēties ARC starpprogrammatūrai piemērotāku *Linux* distributīvu.

ARC starpprogrammatūrai tika izvēlēts *Fedora Core 5 Linux* distributīvs:

- tas ir .rpm pakotņu bāzēts;
- tas ir bez maksas (pretstatā *SUSE Linux* un *RedHat Linux*);
- tas ir atzīts kā savietojams ar ARC starpprogrammatūru.

---

## 2. ARC UZSTĀDĪŠANA (TEHNISKAIS PROCESS)

Kā ceļvedis ARC uzstādīšanai tika izmantota *NorduGrid* mājaslapā pieejamā ARC servera uzstādīšanas pamācība [3]. Tomēr tajā minētas tikai galvenās darbības, kuras parasti ir vienādas uz dažādiem *Linux* distributīviem, bet specifiskās lietas, piemēram, programmatūras atkarības, šajā pamācībā nav atrodamas.

Pirms ARC uzstādīšanas jāpārlicinās par dažu priekšnosacījumu spēkā esamību:

- lai servera autentifikācija pret ārējiem un iekšējiem servisiem būtu iespējama, jāstrādā atgriezeniskajam DNS (*reverse DNS*), kuram jānorāda uz servera IP adresi. Atgriezeniskajam DNS jāstrādā gan no iekšējā tīkla, gan no ārpusaules. To var veikt, izmantojot DNS serveri vai arī `/etc/hosts` failu. Ja tiek izmantots `/etc/hosts` fails, tad pilnajam saimniekdatora nosaukumam (*fully qualified hostname*) jābūt rakstītam pirms īsā nosaukuma, piemēram:

```
1.2.3.4      nosaukums.domain.com nosaukums
```

- svarīgi, lai visiem datoriem klasterī pulkstenis rādītu vienu un to pašu laiku. Nepieciešams uzstādīt regulāru laika sinhronizāciju ar tuvāko NTP (*network time protocol*) serveri. Latvijā var izmantot `ntp.latnet.lv`. Var izveidot arī savu NTP serveri un klastera datoru laiku sinhronizēt ar to. Sīkāku informāciju par publiskajiem NTP serveriem vai sava servera izveidi var uzzināt [www.ntp.org](http://www.ntp.org);
- Grid vajadzībām jāizveido vismaz viens lietotājs (ja ir vēlme, var veidot vairākus, dalot tos pa grupām un definējot dažādas pieejas atļaujas), piemēram, ar nosaukumu 'grid';
- uz priekšgala datora jāizveido direktorijas Grid vajadzībām (`/scratch/grid`, `/scratch/cache`, `/SOFTWARE/runtime`, `/var/spool/nordugrid/jobstatus`). Galvenās direktorijas ir `grid` un `jobstatus` (tās ir obligātas), pārējās ir vēlamas. Plašāka informācija par nepieciešamajām direktorijām un to funkcijām pieejama ARC servera uzstādīšanas pamācībā [3] un *Grid Manager* dokumentācijā [4].
- tā kā priekšgala datoram jābūt pieejamam no ārējā tīkla, attiecīgajiem servisiem nepieciešams atvērt portus:
  - ienākošai un izejošai datu plūsmai:
    - MDS (noklusēti 2135);

- GridFTP (noklusēti 2811 un 9000-9300 datu pārraidei);
- HTTP un HTTPS (noklusēti 80 un 443);
- tikai izejošai datu plūsmai:
  - HTTPg (noklusēti 8443);
  - SMTP (noklusēti 25)
  - NTP (noklusēti 123) – šis nepieciešams ja tiek izmantots ārējais NTP serveris laika sinhronizācijai.

Ja visi iepriekšminētie nosacījumi izpildās, tad var sākt nepieciešamās programmatūras uzstādīšanu. ARC starpprogrammatūra sastāv no vairākām komponentēm. Tās jāuzstāda noteiktā secībā. Visa nepieciešamā programmatūra pieejama *NorduGrid* FTP serverī [5]. Papildus programmatūra (piemēram, kompilators) pieejama izmantojot *Fedora Core 5 Linux iebūvēto pakotņu pārvaldnieku Yum*.

Pirms ARC uzstādīšanas nepieciešams uzstādīt papildu programmatūru:

*Piezīme: ja nav minēts citādāk, tad tiek izmantota jaunākā versija no latest direktorijas. Uzstādīšana tiek veikta izmantojot komandrindas saskarni. Lai pakotņu pārvaldnieks uzstādītu nepieciešamās atkarības (ja tādas ir pieejamas repozitorijos), visu pakotņu uzstādīšana tiek veikta sekojoši (ar root tiesībām):*

```
# yum localinstall paketne.rpm
```

*Tā kā NorduGrid veidotās .rpm pakotnes nav digitāli parakstītas, tad Yum pēc noklusējuma atsakās tās uzstādīt, jo nevar garantēt to izcelsmi. Lai paraksta pārbaudīšanu atceltu, nepieciešams izmainīt Yum konfigurāciju - /etc/yum.conf failā gpgcheck=1 jānomaina pret gpgcheck=0. Pēc ARC uzstādīšanas ieteicams šo parametru iestatīt noklusētajā stāvoklī.*

- Grid Packaging tools (pieejama FTP serverī [5]);
  - lai uzstādītu GPT, pirms tam jāuzstāda gsoap un gsoap-devel
- Globus Toolkit 4 (pieejama FTP serverī [5]);
  - Globus Toolkit 4 sastāv no vairākiem desmitiem pakotņu (iespējams, ka pakotnes, kuru nosaukuma beigās ir devel, jāuzstāda pirms tām, kurām tas nav). Tās visas jālejuplādē un jāuzstāda. Ja tās saglabātas vienā direktorijā un tajā citu failu nav, tad visas pakotnes var uzstādīt ar vienu komandu – paketne.rpm vietā rakstot  
\*;

- 
- Pēc noklusējuma GPT un Globus tiek uzstādīti /opt/gpt un /opt/globus direktorijās. Lai vai kā, vēlams uzstādīt sistēmas mainīgos GLOBUS\_LOCATION un GPT\_LOCATION, lai tie norāda uz attiecīgajām direktorijām. Ja uzstādīšana bijusi veiksmīga tad šie mainīgie jau ir iestatīti, par to var pārliecināties ar komandu printenv. Savukārt, ja tie nav iestatīti, tad nepieciešams ierakstīt sekojošās rindas /etc/bashrc failā:

```
export GLOBUS_LOCATION="/opt/globus"
```

```
export GPT_LOCATION="/opt/gpt"
```

- Tā kā jaunākā ARC versija (0.6) *Fedora Core 5 Linux* nav pieejama binārā formātā, tā jākompilē no pirmkoda. Tam nepieciešams uzstādīt kompilatorus. Fedora distributīvā ir izveidota pakotņu grupa, kura satur vajadzīgos kompilatorus:

```
yum groupinstall "Development Tools"
```

- Viss nepieciešamais ARC uzkompilēšanai ir paveikts:

```
rpmbuild --rebuild nordugrid-arc-<x.y.z>.src.rpm
```

,kur x.y.z ir jaunākās versijas numurs.

Kad ARC starpprogrammatūras komponentes uzkompilētas, tās jāuzstāda. ARC uzkompilētā versija sastāv no vairāk par 10 pakotnēm. Visas nav obligāti jāuzstāda. Minimālai ARC instalācijai nepieciešamas sekojošas pakotnes: nordugrid-arc-server, nordugrid-arc-ca-utils, nordugrid-arc-gridmap-utils un nordugrid-arc-libs. Savukārt nordugrid-arc-client, nordugrid-arc-compatible, nordugrid-arc-libs-devel un nordugrid-arc-doc pakotnes ir vēlamas.

Uzstādīšanas process ir tieši tāds pats kā iepriekš:

```
yum localinstall pakotnes_nosaukums
```



---

### 3. ARC UZSKAIŠANA

Visa ARC konfigurācija (neatkarīgi no servera funkcijām) glabājas vienā failā - `/etc/arc.conf`. ARC konfigurācijas faila paraugs glabājas noklusētajā ARC direktoriņā (ja uzstādīšanas gaitā tā nav mainīta): `/opt/nordugrid/share/doc/arc.conf.template`. Sākot darbu, tas jāpārkopē uz `/etc` direktoriju. `arc.conf` satur gan visu ARC konfigurāciju gan plašu dokumentāciju, tādējādi sākotnējais šablona fails satur teju vai 1500 rindiņas, no kurām lielāka daļa ir komentāri. Pieejamais paraugfails bez labošanas nav derīgs kā ARC konfigurācijas fails, tas ir jāpielāgo.

ARC konfigurācijas fails ir sadalīts pa blokiem, kur katrs no tiem atbilst kādam no ARC servisiem:

- `common`, `grid-manager`, `gridftpd`, `httpsd`, `group`, `vo`, `infosys`, `registration`, `cluster`, `queue`, `se` (visi tiek rakstīti [kvadrātiekvās]).

Īsumā par galvenajiem `arc.conf` blokiem:

- **common** – satur pamatinformāciju par serveri, definē konfigurācijas faila atrašanās vietu, kāda pakešdatu apstrādes sistēma tiek izmantota, kāds ir pilnais saimniekdatora nosaukums, kur glabājas sertifikāti u.c. iestatījumi. *Common* blokā esošo informāciju izmanto citi bloki;
- **group** – šajā blokā iespējams definēt lietotāju grupas, to pieejas tiesības izpildot uzdevumus un kuri lietotāji šajā klasterī drīkstēs izpildīt uzdevumus. Informāciju par grupām izmanto `gridftpd` un `httpsd` bloki;
- **vo** – VO jeb *NorduGrid* virtuālā organizācija ir lietotāju un servisu pārvaldības sistēma. VO sastāv no lietotāju un servisu datubāzes, kuras tiek glabātas LDAP serverī. *NorduGrid* VO datubāze satur lietotāju sarakstu, definējot to piekļuves tiesības noteiktiem *NorduGrid* resursiem [6]. Šajā `arc.conf` sadaļā var definēt, kādas VO datubāzes tiks izmantotas, to adreses un piekļuves vietas;
- **grid-manager** – šī servisa uzdevums ir apstrādāt iesniegtos uzdevumus, padot tos pakešdatu apstrādes sistēmai, apstrādāt uzdevuma ievad- un izvaddatu failus. Šajā blokā tiek definēti `grid-manager` uzstādījumi, piemēram, direktorijas, kurās glabāt informāciju par šobrīd izpildāmajiem uzdevumiem, kur glabāt žurnālfailus. Plašs

---

grid-manager uzskaitošanas apraksts pieejams *The NorduGrid Grid Manager And GridFTP Server* [4].

- **gridftpd** – visa datu pārraide starp klienta datoru un grid-manager (priekšgalu) notiek, izmantojot GridFTP protokolu. Šajā blokā var iestatīt žurnālfailu atrašanās vietu, pieslēgumvietu, cik vienlaicīgi savienojumi drīkst būt. Plašs gridftpd uzskaitošanas apraksts pieejams *The NorduGrid Grid Manager And GridFTP Server* [4].
- **cluster** – šajā blokā tiek definēta informācija par klasteri. Tā būs pieejama citiem NorduGrid tīklā esošajiem resursiem. Šī sadaļa ir informatīva, tomēr vēlams to aizpildīt korekti, piemēram, operētājsistēmas nosaukumu, procesoru jaudu, operatīvās atmiņas apjomu u.c. parametrus;
- **queue** – šajā blokā tiek definēts, kādu pakešdatu apstrādes sistēmu dotais klasteris izmanto un kā ARC starpprogrammatūrai ar to sazināties un ar tās starpniecību izpildīt uzdevumus.
- **se** – SE jeb *storage element* ir atsevišķa vienība, kura nodrošina datu resursus. Šajā blokā tiek definēti SE parametri, tā nosaukums, atrašanās vieta, kādas virtuālās organizācijas drīkst to izmantot.

ARC darbību var pārbaudīt ar vienkāršas 'Hello World' programmas palīdzību:

```
ngsub -e '&(executable="/bin/echo")(arguments="Hello World")(stdout="stdout")' -c arc.servera.adrese
```

ARC tik iesūtīts uzdevums, kurš ar /bin/echo ieraksta Hello World failā stdout.

Plašāka informācija par ARC uzskaitošānu ir pieejama `arc.conf` failā un papildu dokumentācijā.

---

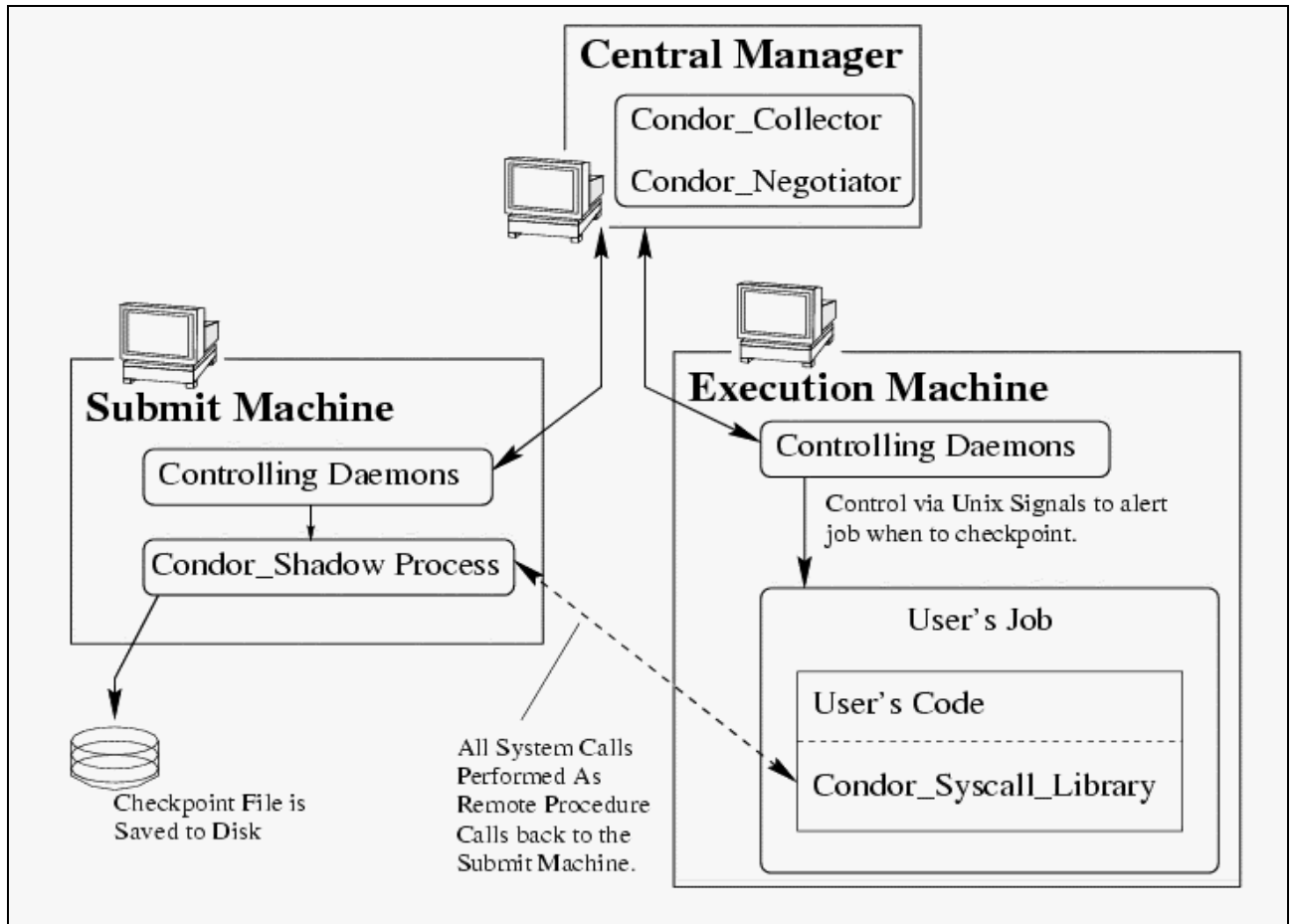
#### 4. CONDOR PAKEŠDATU APSTRĀDES SISTĒMA

ARC starpprogrammatūra darbojas Grid līmenī. Tā savieno dažādus klasterus vienotā tīklā. Zemākajā, klastera līmenī darbojas pakešdatu apstrādes sistēma, kura veic uzdevumu dalīšanu, savieno atsevišķos datorus vienotā klasterī. ARC atbalsta vairākas pakešdatu apstrādes sistēmas (*PBS, Torque, Condor* u.c.), no kurām piemērotākā tieši *Linux* Centra datorklasei ir *Condor* [7]. Tā ir paredzēta tieši tādai darba videi, kāda ir LC, kur paralēli klastera uzdevumiem datori tiek izmantoti kā darbstacijas.

*Condor* sistēmā var definēt, kādai jābūt datora noslodzei, lai tam drīkstētu dot pildīt uzdevumu. Kad pēdējo reizi bijusi kaut kāda aktivitāte – tā varēt būt gan peles kustināšana vai klaviatūras izmantošana vai procesora noslodze. Ja esošie rādītāji apmierina uzstādītos, tad uzdevums tiek dots konkrētajam datoram, ja nē, tad netiek.

*Condor* sistēmā katrs dators var pildīt kādu no funkcijām:

- *Central manager* jeb serveris – šo funkciju pilda tikai viens dators;
- *Execute* jeb uzdevumu izpildītājs – šo funkciju var pildīt ikviens klasterī esošs dators (tai skaitā serveris);
- *Submit* jeb darbu iesniedzējs - šo funkciju var pildīt ikviens klasterī esošs dators (tai skaitā serveris).



## 2. att. Condor arhitektūra

*Condor* uzstādīšana salīdzinājumā ar ARC ir samērā ērta un vienkārša. Tam pieejamas binārās pakotnes (gan ar dinamiskajām atkarībām, gan statiskajām) dažādiem *Linux* distributīviem un *Unix* paveidiem, tai skaitā *Debian Linux*, *Fedora Core*, *RedHat Enterprise Linux*, *SuSE Linux*, *Mac OS X*, *FreeBSD* un pat *Windows* operētājsistēmai. Ērtākais ir izmantot failu ar statiskajām atkarībām. Tā kā *Condor* nepieciešams uzstādīt uz visiem klasterī esošajiem datoriem, tad statisko atkarību versija nodrošina ērtu pārnēsamību uz citu sistēmu ar ļoti minimālu uzskaitošānu. Tā kā esošajā infrastruktūrā *Condor* servera funkciju pildīs tas pats dators, uz kura uzstādīta ARC starpprogrammatūra, tad tam var izmantot pieejamo dinamisko pakotni *Fedora Core 5 Linux* distributīvam, starp citu, šī pakotne ir paredzēta arī *RedHat Enterprise Linux 3* distributīvam un nosaukumā minēts tieši tas.

*Condor* programmatūra pēc noklusējuma meklē sistēmā lietotāju ar vārdu `condor`. Ērtākais variants ir visos klastera datoros izveidot lietotāju `condor`, turklāt, vēlams, lai visos datoros savstarpēji sakristu lietotāja identifikators (UID) un grupas identifikators (GID). Piemēram, uz visiem datoriem lietotājam `condor` UID = 1002, GID = 1002 (UID un GID nav obligāti jābūt vienādiem, bet parasti tā būs). Tādējādi konfigurācijas failā šis parametrs nebūs jāmaina, kopējot *Condor* direktoriju no viena datora uz citu.

---

*Condor* uzstādīšana no rpm pakotnes ir tieši tāda pati, kā iepriekš aprakstīts ARC gadījumā:

```
yum localinstall condor-xyz-linux-x86-rhel3-dynamic-1.i386.rpm
```

kur xyz ir jaunākā *Condor* versija (šobrīd 6.8.5). Pārējām darbstacijām var izmantot statisko *RedHat Enterprise Linux 3 Condor* versiju un no servera nokopēt konfigurācijas failus. Noklusētā *Condor* direktorijs ir `/opt/condor-xyz`. Tajā glabājas visi izpildāmie faili un konfigurācija. Globālais konfigurācijas fails ir `/opt/condor-xyz/etc/condor_config`, savukārt katrai darbstacijai pieejams lokālais konfigurācijas fails, kurā var definēt atribūtus, kuri attiecas tikai uz lokālo datoru, tas atrodas:

`/opt/condor-xyz/local.nosaukums/condor_config.local`, kur nosaukums ir datora īsais nosaukums, piemēram, *Linux* Centra laboratorijā visiem datoriem piešķirti nosaukumi: lab1, lab2, ..., lab20. Katra datora saimniekdatora nosaukumam jābūt unikālam iekšējā tīkla robežās. *Ubuntu Linux* saņemot IP adresi izmantojot DHCP klientu, nepiešķir datoram saimniekdatora nosaukumu automātiski, kā to dara, piemēram, *Fedora Core Linux*. Tas jāveic ar rokām izmainot `/etc/hostname` un `/etc/hosts` failus. Pārkopējot *Condor* no viena datora uz citu, nepieciešams pārsaukt `/opt/condor-xyz/local.nosaukums`, direktorijas nosaukumu, lai tā atbilstu dotā datora nosaukumam.

Lai *Condor* spētu strādāt, tam nepieciešams arī sistēmas mainīgais, kurā norādīta globālā konfigurācijas faila atrašanās vieta. Jāuzstāda `CONDOR_CONFIG` mainīgais, pēc noklusējuma `/opt/condor-xyz/etc/condor_config`.

*Condor* uzskaitošana “zemūdens akmeņi” nav novēroti un galvenais, kas jāpaveic, lai iegūtu strādājošu klasteri, ir `condor_config` failā norādīt, kur glabājas lokālās darbstacijas konfigurācijas fails (šis parametrs jāmaina, ja konfigurācijas faili tiek kopēti no vienas darbstacijas uz citu), klastera nosaukums, *Condor* atrašanās direktorijs, `condor` lietotāja mājas direktorijs. Ar to pietiek, tomēr vēlams arī definēt, kuri datori drīkstēs iesniegt uzdevumus, kuri nē. Plašs `condor_config` direktīvu apraksts pieejams *Condor* lietotāja pamācībā [8].

---

## 5. CONDOR DARBĪBĀ

*Condor* darbības pārbaudīšanai izmantota neliela programma, kas tiek izsaukta ar diviem argumentiem norādot skaitļu intervālu, kurā tiks meklēti pirmskaitļi. Visi pirmskaitļi, kuri atradīsies dotajā intervālā tiks izdrukāti. Programmas izsaukšana no komandrindas ir šāda:

```
primes 1 1000
```

Tiks izdrukāti visi pirmskaitļi intervālā no 1 līdz 1000.

Lai šo uzdevumu varētu izpildīt *Condor* klasterī, jāizveido uzdevuma iesniegšanas fails, kurš saturēs nepieciešamo informāciju par tā izpildi. Šim failam parasti izmanto .cmd paplašinājumu. `primes.cmd` piemērs:

```
executable      = primes
output          = prime.$(Process).out
error          = prime.$(Process).err
log            = prime.log
arguments      = 1 9999
queue
arguments      = 10000 19999
queue
arguments      = 20000 29999
queue
```

Izpildāmais fails (*executable*) ir `primes`, izejas dati tiks rakstīti failos ar nosaukumu `prime.uzdevuma_nr.out` (piemēram, `prime.1.out`, `prime.2.out`), kļūdas paziņojumi tiks rakstīti pēc tāda paša principa. Katram procesam nepieciešams savs fails, jo citādi katrs nākamais pārrakstīs iepriekšējā izvadam pāri, nevis papildinās to. `prime.log` tiks izmantots kā žurnālfails. Tālā seko 3 `primes` izsaukumi, kur katrs no tiem tiks izsaukts ar citiem argumentiem.

Uzdevuma iesniegšana tiek veikta ar `condor_submit` komandu:

```
condor_submit prime.cmd
```

*Condor* pats parūpēsies, uz kuriem klastera datoriem dotos 3 darbus izpildīt. Rezultātā tiks iegūti trīs faili, kuros būs ierakstīti visi pirmskaitļi intervālā no 1 līdz 29999.



---

Kad *Condor* saslēgts kopā ar ARC (`arc.conf` failā norādot `lrms = "condor"`), tad visi darbi, kuri tiks iesniegti ar ARC `ngsub` komandu, lokālā klastera līmenī automātiski tiks iesniegti *Condor* pakešdatu apstrādes sistēmai un pēc tam izvads padots atpakaļ uz ARC.



---

## 6. NOBEIGUMS

LU *Linux* Centrā ir realizēts klasteris, kurš izmanto *Condor* pakešdatu apstrādes sistēmu un *NorduGrid* ARC starpprogrammatūru. Uzstādīts ARC starpprogrammatūras serveris un veikta tā uzturēšana. ARC uzstādīšanas gaitā radušās problēmas palīdzēja risināt ARC lietotāju atbalsts [9], kurā darbojas zinoši un atsaucīgi cilvēki. Uzstādīta *Condor* pakešdatu apstrādes sistēma – serveris un klienti, kuri veic uzdevumu risināšanu. Izstrādātas procedūras, kā pievienot papildu datorus izveidotajam klasterim.

Citi Latvijas klasteri kā starpprogrammatūru izmanto EGEE (*Enabling Grids for E-sciencE*) izveidoto un lietoto *gLite*, kuras attīstībā aktīvi darbojas CERN laboratorija. Diemžēl šobrīd pastāv problēmas ar *gLite* un ARC starpprogrammatūru savietojamību, tādēļ *Linux* Centra klasteris nav iekļauts vienotajā Grid tīklā.





---

## 7. AVOTI

1. *"The NorduGrid architecture and tools"*. P.Eerola et al., in Proceedings of CHEP 2003,eConf C0303241:MOAT003,2003. - <http://www.nordugrid.org/documents/MOAT003.pdf>
2. *"The NorduGrid "Smart" Storage Element"*. A.Konstantinov [NORDUGrid-TECH-10] - <http://www.nordugrid.org/documents/SE.pdf>
3. *ARC 0.5.x/0.6.x Server Installation Instructions: Setting Up A Grid Resource* – <http://www.nordugrid.org/documents/ng-server-install.html>
4. *"The NorduGrid Grid Manager And GridFTP Server: Description And Administrator's Manual"*. A.Konstantinov [NORDUGrid-TECH-2] - <http://www.nordugrid.org/documents/GM.pdf>
5. NorduGrid projekta FTP serveris programmatūras lejupielādei – <ftp://ftp.nordugrid.org>
6. NorduGrid Virtual Organisation - <http://www.nordugrid.org/NorduGridVO/vo.php>
7. Condor Project Homepage - <http://www.cs.wisc.edu/condor/>
8. Condor Manual - <http://www.cs.wisc.edu/condor/manual/v6.8/>