

Sistemes Distribuïts en Xarxa (SDX)  
Facultat d'Informàtica de Barcelona  
Examen Final (2<sup>a</sup> part). 18 de Juny 2015

**Contesteu a les preguntes de manera concisa i precisa**  
**Contesteu al mateix full**  
**No es poden consultar apunts**  
**(només els VOSTRES informes de les lectures)**  
**Durada: 75 minuts**  
**Publicació de notes: abans del 24 de Juny**  
**Revisió de notes: 25 de Juny a les 12h**

---

**Nom i Cognoms:**

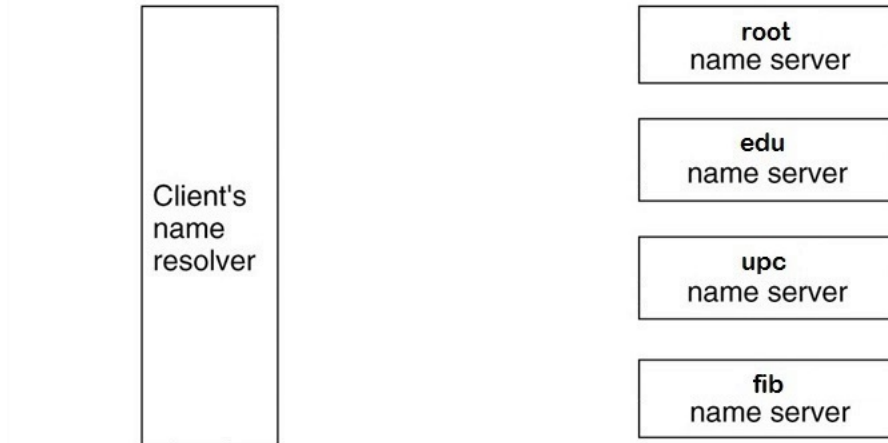
---

1. (5 punts) Seleccioneu la resposta (una només) que considereu correcta en cadascun dels apartats. Cada resposta correcta val 1/2 punts. Cada resposta incorrecta resta 1/6.
  - (a) Quin dels següents components de LDAP és comparable en funcionalitat als *Name Resolvers* de DNS?
    - Directory User Agents (DUA)
    - Directory Service Agents (DSA)
    - Directory Information Tree (DIT)
    - Directory Information Base (DIB)
  - (b) Quina semàntica utilitza NFS a l'hora de compartir fitxers?
    - UNIX semantics
    - Immutable files
    - Session semantics
    - Transactional semantics
  - (c) Quina de les següents afirmacions referents al sistema de fitxers distribuït Coda és falsa?
    - Els fitxers s'agrupen en volums
    - Un fitxer pot formar part de més d'un volum
    - El volum és la unitat de muntatge als clients
    - El volum és la unitat de replicació als servidors
  - (d) Quin dels següents paradigmes permet generar codi per sistemes distribuïts basats en Web que no s'executa a la part del servidor?
    - CGI
    - JSP
    - Java Servlet
    - JavaScript

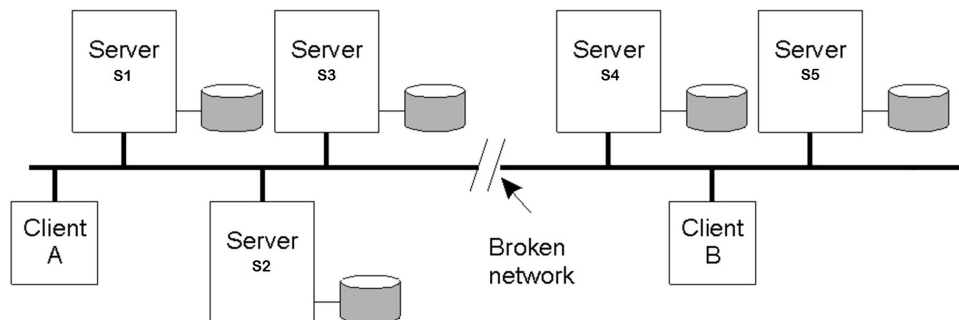
- (e) Quina de les següents alternatives faries servir per registrar i cercar Web Services?
- UDDI
  - SOAP
  - XML
  - WSDL
- (f) Què indica el paràmetre *time-to-live (TTL)* que assigna Gnutella a cada missatge de cerca?
- El temps de validesa del missatge
  - El nombre de nodes veïns als quals es reenviarà el missatge
  - El nombre màxim de nodes que poden rebre el missatge
  - El nombre màxim de vegades que el missatge pot ser reenviat
- (g) Quina de les següents polítiques utilitzades per *BitTorrent* s'aplica només al principi d'una descàrrega?
- Rarest Piece First
  - Strict Priority
  - Random First Piece
  - End Game Mode
- (h) Quin dels següents tipus de Cloud consisteix en disposar d'una infraestructura operada segons els principis del Cloud per l'ús exclusiu d'una única organització?
- Public Cloud
  - Private Cloud
  - Community Cloud
  - Hybrid Cloud
- (i) Quina de les següents formes de volatilitat exhibeixen els sistemes de computació mòbil i ubiqua?
- Fallades en els dispositius i en els canals de comunicació
  - Canvis en les característiques de la comunicació (p.ex. en l'ample de banda)
  - Volatilitat en la creació i destrucció d'associacions entre components
  - Totes les anteriors
- (j) Quina de les següents afirmacions referents a les *wireless sensor networks* és falsa?
- Són xarxes descentralitzades sense cap control global
  - Estan formades per una gran quantitat de dispositius petits, amb capacitat sensorial, de computació i de comunicació
  - Cada node té comunicació directa amb tots els altres
  - Usen tècniques diverses amb l'objectiu de conservar l'energia i operar contínuament a pesar de la volatilitat

Nom i Cognoms:

2. (1,25 punts) Donat el següent servei de resolució de noms, numera ordenadament els missatges que s'envien entre el resolver i els diferents servidors de noms per resoldre el domini  $[www, fib, upc, edu]$  i tot seguit el domini  $[ftp, fib, upc, edu]$  utilitzant resolució de noms **recursiva** i tenint en compte que el resolver no fa caching però els servidors de noms sí, indicant per cada missatge el seu contingut: quina part del domini es demana resoldre i quins resultats s'obtenen. Indica també la llista de subdominis i el contingut de la cache del servidor de noms 'root' un cop finalitzades les dues resolucions.



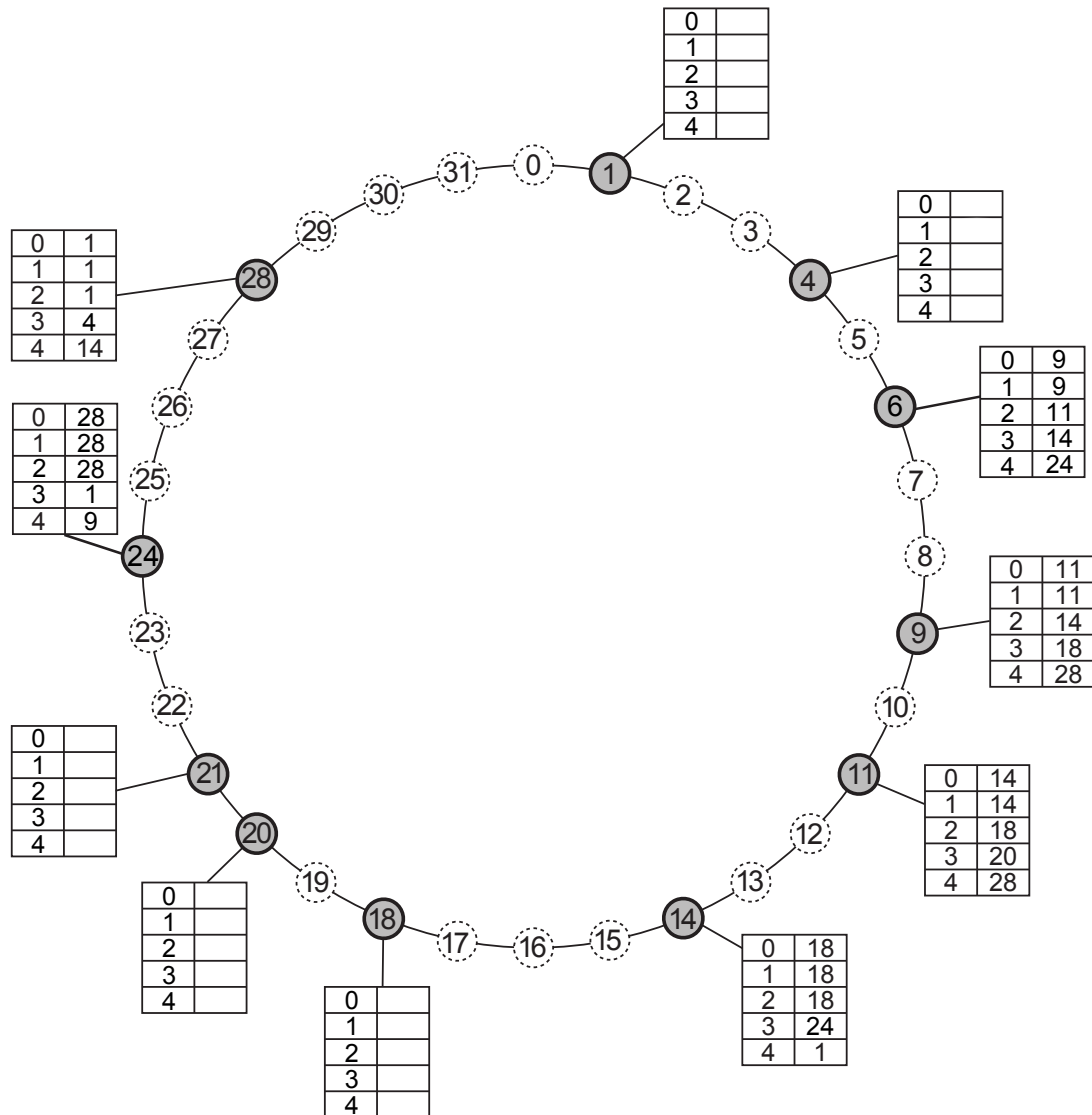
3. (1,25 punts) Donat el següent sistema de fitxers distribuït Coda, on el volum que conté el fitxer 'a' està replicat als servidors  $\{S1, S2, S3, S4\}$ , i el volum que conté el fitxer 'b' està replicat als servidors  $\{S2, S4, S5\}$ :



- a) Indica el contingut del Accessible Volume Storage Group (AVSG) dels clients A i B per cadascun dels dos fitxers.
- b) Indica quins servidors accedirà cada client si cadascun executa la seqüència d'operacions detallada a continuació mentre la xarxa està particionada. Per simplificar, pots assumir que les operacions es fan directament als servidors, sense utilitzar la cache local dels clients.
- Client A:  $\begin{cases} read('a'): \\ read('b'): \\ write('a'): \end{cases}$
  - Client B:  $\begin{cases} read('b'): \\ write('a'): \\ write('b'): \end{cases}$
- c) Explica com detectaria Coda el conflicte d'escriptura concurrent en el fitxer 'a' que es genera, un cop reparada la partició de xarxa, per l'execució de les seqüències anteriors.

4. (1,5 punts) Donat el següent sistema P2P Chord, on els nodes ombrejats són aquells que formen part del sistema en aquest moment:

- Completa les finger tables dels nodes 1, 4, 18, 20 i 21.
- Indica la interacció de missatges quan i) el node 6 fa un lookup de la key 22, ii) el node 18 fa un lookup de la key 3.
- Si el node 15 fa join al sistema, indica el contingut de la seva finger table, les modificacions que s'haurien de fer a les finger tables dels altres nodes, i quines keys li correspondrien al node 15.



Nom i Cognoms:

5. (1 punts) Assenyala quins sistemes de computació distribuïda (Volunteer, Grid, Cloud) encaixen amb cadascun dels conceptes següents:

|  | Volunteer | Grid | Cloud |
|--|-----------|------|-------|
| Recursos heterogenis                         |           |      |       |
| Recursos geogràficament distribuïts          |           |      |       |
| Recursos de diferents organitzacions/usuaris |           |      |       |
| Qualsevol tipus de recurs                    |           |      |       |
| Aplicacions científiques                     |           |      |       |
| Pay-as-you-use                               |           |      |       |
| Work-units                                   |           |      |       |
| Virtualització                               |           |      |       |
| Transparencia                                |           |      |       |
| Model de computació batch                    |           |      |       |
| BOINC  |           |      |       |
| Analogia amb la xarxa elèctrica              |           |      |       |
| Aplicacions transaccionals/interactives      |           |      |       |
| OGSA   |           |      |       |
| SOA  |           |      |       |
| Amazon EC2                                   |           |      |       |
| Virtual Organizations                        |           |      |       |
| Map-Reduce                                   |           |      |       |
| SETI@home                                    |           |      |       |
| PaaS   |           |      |       |

6. (READINGS) Contesta les següents preguntes referents als articles llegits a l'assignatura.

i) Descriu breument les diferències entre els datacenters on s'executen serveis Web i els equips *desktop* tradicionals segons l'article *Barroso09*.

ii) Contesta una de les següents preguntes en funció de l'article que vas escollir:

a) Explica en què consisteix la política de *Anti-snobbing* que usa BitTorrent segons l'article *Cohen03*.

b) Explica que conté la taula de routing de cada node en Kademlia segons l'article *Maymounkov02*.

iii) Resumeix el problema de *Performance Unpredictability* tal com es descriu a l'article *Armbrust10*.

## 7. (SEMINARIS) GEQ.

- a) **Groupy**: Completa el següent extracte de codi corresponent al procediment d'elecció de nou líder en la implementació *gms3*.

```
% APIs: leader(Name, Master, N, Slaves)
%       slave(Name, Master, Leader, N, LastMsg, Slaves, Ref)
%       bcast(Name, Msg, Nodes)
%       {view, N, Leader, Slaves}

election(Name, Master, N, LastMsg, Slaves) ->
    Self = self(),
    case Slaves of
        [Self|Rest] ->
            bcast(Name, ... , ... ),
            bcast(Name, {view, ... , ... , ... }, ... ),
            leader(Name, Master, ... , ... );
        [NewLeader|Rest] ->
            NewRef = erlang:monitor(process, ... ),
            slave(Name, Master, ... , ... , ... , ... , ... )
    end.
```

- b) **Namy**: Donada la versió per defecte de Namy amb resolució iterativa i sense caching:

- i) Quin seria el comportament si resolem el nom d'un host, després l'apaguem i finalment intentem resoldre el seu nom una altra vegada?

- ii) Quin seria el comportament si usem la versió de Namy modificada de manera que quan s'apaga un host es desenregistra el seu nom en el seu domini pare?

- iii) Quin seria el comportament si tenim el caching al resolver activat?

Nom i Cognoms:

8. (SEMINARIS) IEQ. **Chordy**.

- a) Completa el següent extracte de codi corresponent al procediment d'estabilització en la implementació del *node3* (la versió que suporta fallades de nodes).

```
% APIs: Ref = monit(Pid)
%       demonit(Ref)
%       between(Key, From, To)

stabilize(Pred, Next, MyKey, Successor) ->
{Skey, Sref, Spid} = Successor,
case Pred of
  nil ->
    Spid ! {notify, {MyKey, self()}},
    {Successor, Next};
  {MyKey, _} ->
    {Successor, Next};
  {Skey, _} ->
    Spid ! {notify, {MyKey, self()}},
    {Successor, Next};
  {Xkey, Xpid} ->
    case key:between( ... , ... , ... ) of
      true ->
        ...
        ...
        ...
        { ... , ... };
      false ->
        ...
        { ... , ... }
    end
end.
```

- b) Enumera les quatre situacions en què cal enviar un missatge **replicate** o **pushreplica** al node successor en la versió de Chordy que implementa replicació.