Bonjour

- Multicast IP
- Multicast JGroups
- L'API JGroups
- Exemple
- Démonstration
- Fonctionnement en détail
- Conclusions
- Questions
- Références

JGroups: un système Java pour communiquer avec un groupe de machines

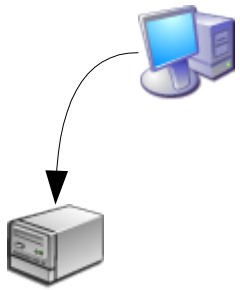
Savaş Ali TOKMEN

M2P Génie Informatique

UFR IMA, Grenoble, FRANCE

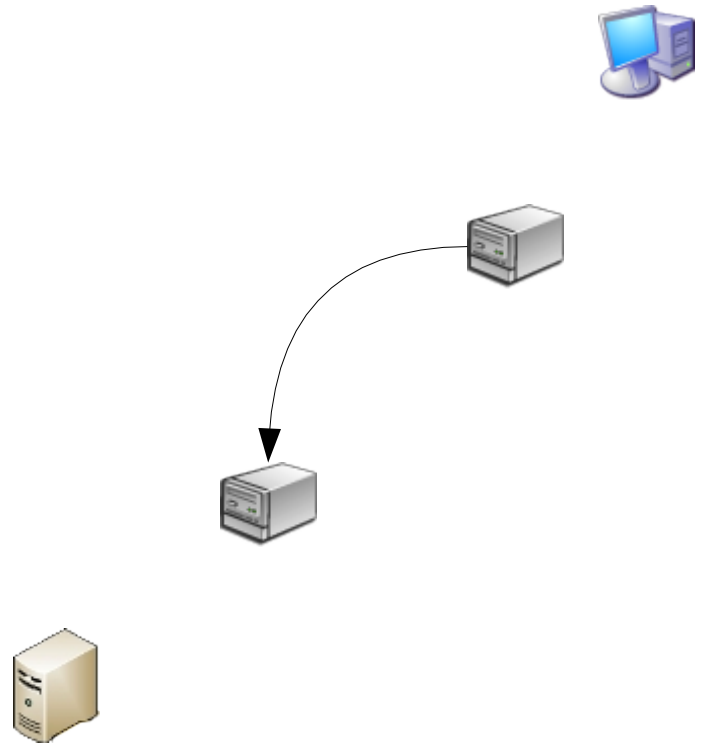
Multicast IP

- But: diffusion de données à plusieurs cibles de façon économe
- Protocole IGMP



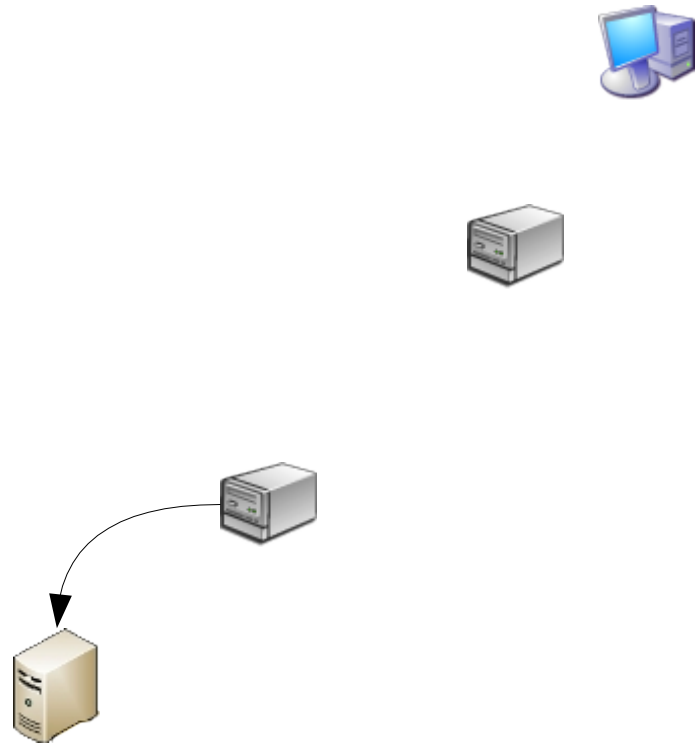
Multicast IP

- But: diffusion de données à plusieurs cibles de façon économe
- Protocole IGMP



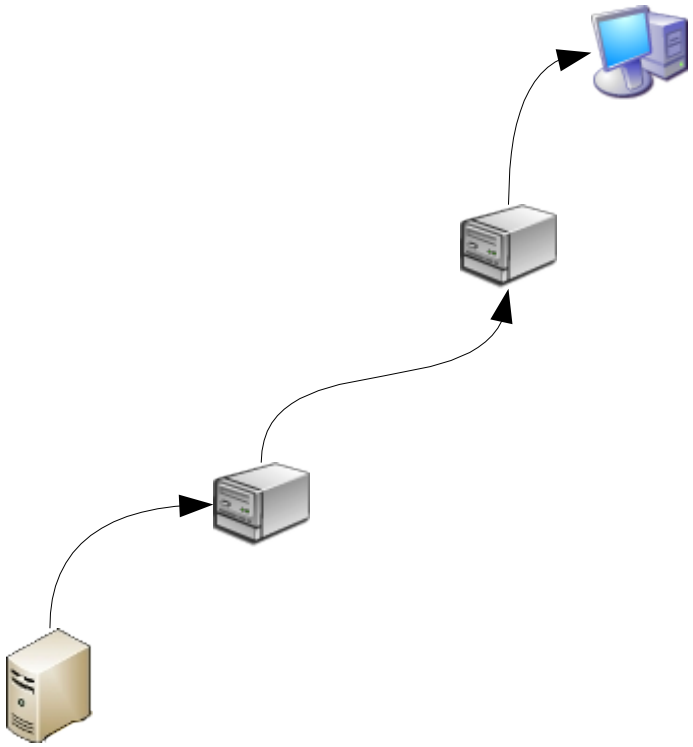
Multicast IP

- But: diffusion de données à plusieurs cibles de façon économe
- Protocole IGMP



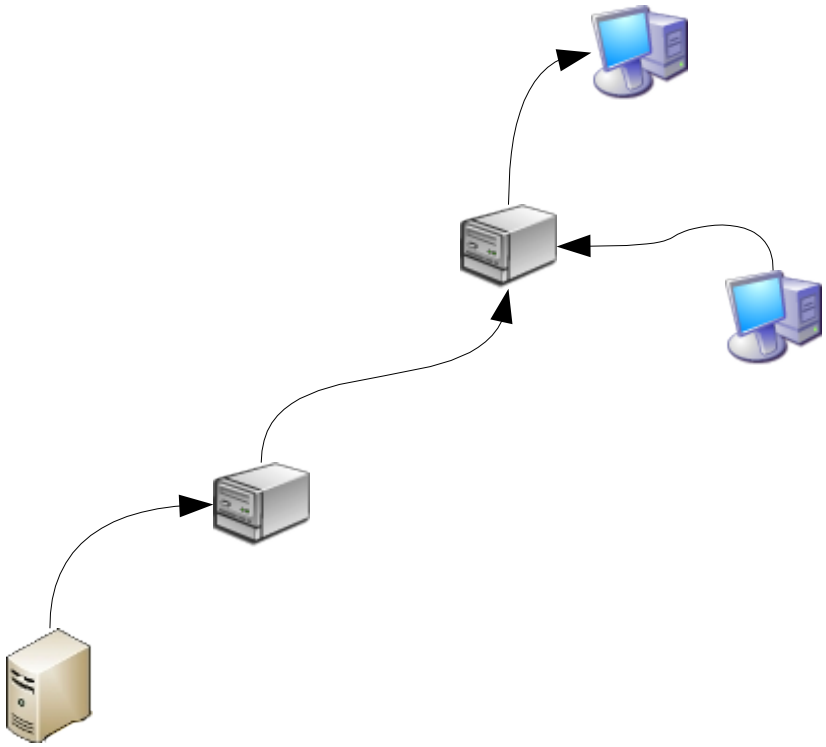
Multicast IP

- But: diffusion de données à plusieurs cibles de façon économe
- Protocole IGMP



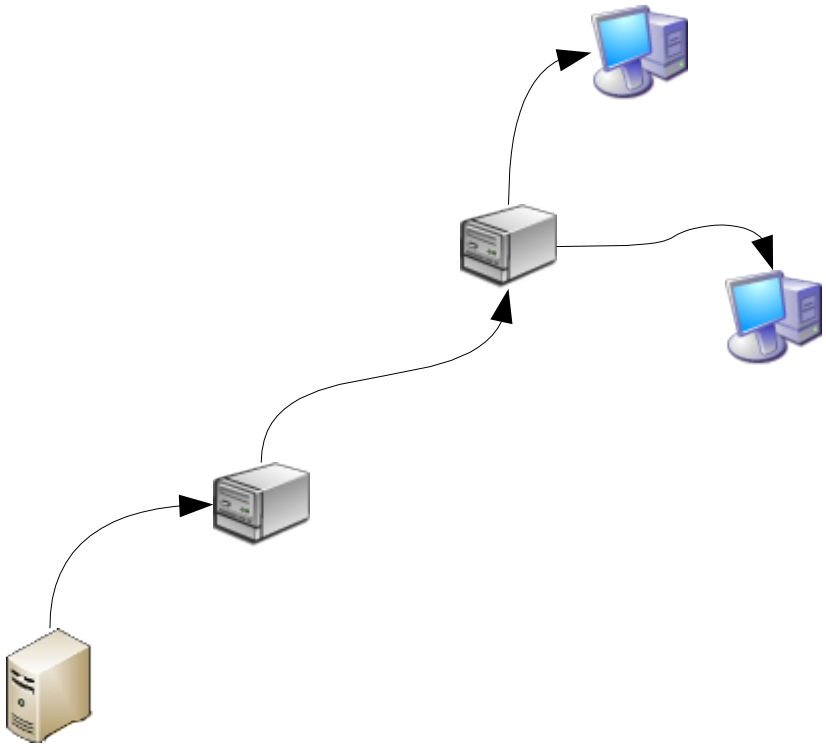
Multicast IP

- But: diffusion de données à plusieurs cibles de façon économe
- Protocole IGMP



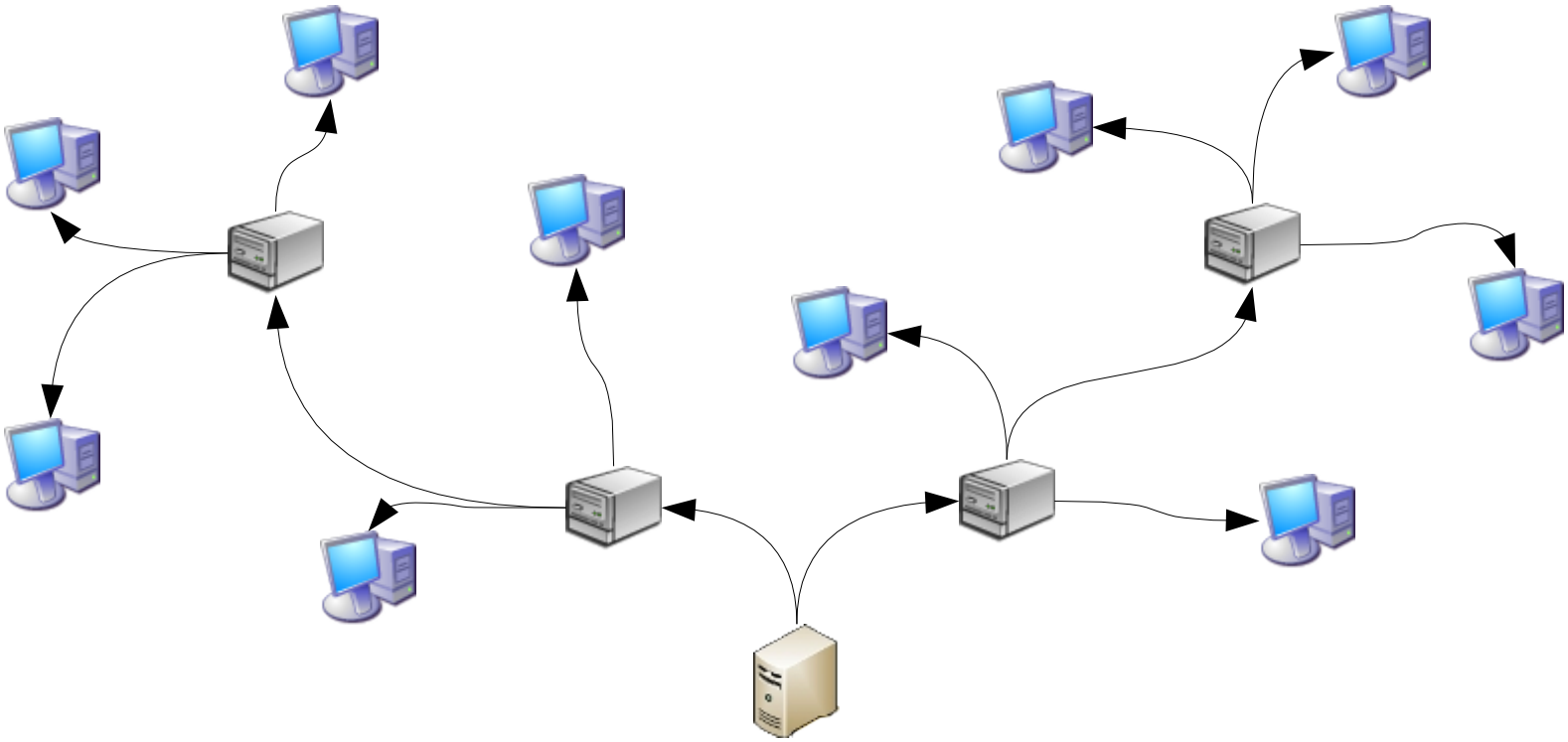
Multicast IP

- But: diffusion de données à plusieurs cibles de façon économe
- Protocole IGMP



Multicast IP

- But: diffusion de données à plusieurs cibles de façon économe
- Protocole IGMP



- Diffusions radio / vidéo, OSPF

Multicast JGroups

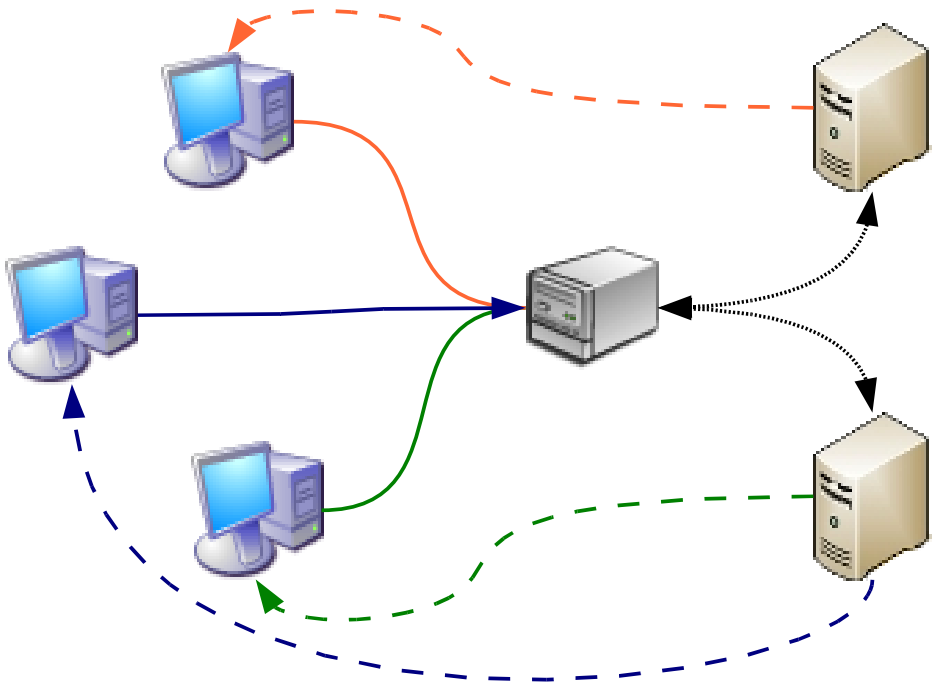
- But initial: classe Java pour le multicast IP
- Extensions
  - Choix de protocoles: UDP, TCP, JMS
  - Notification d'abonnement et de désabonnement
  - Détection de paires ayant plantés
  - Transmission fiable et ordonné des messages
  - Messages point-à-point
  - Fragmentation des messages trop grands
  - Politiques d'ordonnancement: atomique (tout ou rien), FIFO, ordre total
  - Encryption

L'API JGroups

- Création d'un canal: `new JChannel(props);` avec propriétés tel que:
  - Protocole de transport
  - Transmission fiable / non-fiable
  - Politiques d'ordonnancement
- Appel de la méthode `connect` pour rejoindre le groupe
- Méthode `send` pour l'envoi multicast tout comme unicast
- Méthode `receive` pour recevoir ou notification par publish / subscribe
- Méthode `disconnect` pour terminer

Exemple

- Système de répartition de requêtes



Types de communications:

Requête HTTP
Réponse HTTP
JGroups

Types de machines:

Client
Serveur
Point d'entrée au réseau

Démonstration

Code répartisseur

```
public void OnRequest( HTTPRequest request ) {
    // Récupérer la liste des serveurs disponibles
    Vector<Address> servers = channel.getView().getMembers();
    if( servers.size() <= 1 ){
        // Dans ce cas là, il n'y a aucun serveur!
        OnError( 501 );
    } else {
        int gatewayPosition = servers.indexOf(channel.getLocalAddress());
        int targetServer = gatewayPosition;
        // Sélectionner le serveur qui doit executer la requête en
        // vérifiant bien que c'est un serveur et pas le point d'entrée
        while ( targetServer  == gatewayPosition ) {
            targetServer = random.nextInt(servers.size());
        }
        // Rediriger la requête au serveur, qui lui répondra directement au client
        channel.send(new Message( servers.get(targetServer), null, request ));
    }
}
```

Code serveur

```
while( true ) {
    // Attendre pour une requête
    Object o = channel.receive(0);
    // Ne traiter que les messages normaux
    if( o instanceof Message ) {
        o = ((Message) o).getObject();
        // Vérifier que c'est une requête HTTP
        if( o instanceof HTTPRequest ) {
            // Appeler le serveur HTTP classique avec la requête
            ProcessRequest( (HTTPRequest) o );
        }
    }
}
```

Fonctionnement en détail

- Ethereal

JGroups - filtered.pcap - Ethereal

File Edit View Go Capture Analyze Statistics Help

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.25.131	228.8.8.8	IGMP	V2 Membership Report
2	0.007931	192.168.25.131	224.0.0.75	IGMP	V2 Membership Report
3	0.084068	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
4	1.150895	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
5	2.443899	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
6	9.965719	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
7	11.028988	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
8	11.146212	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
❄	❄	❄	❄	❄	❄
10	21.598201	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
11	22.212492	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
12	27.688201	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
13	28.136554	192.168.25.131	228.8.8.8	UDP	Source port: 1033 Destination port: 45566
14	29.132816	192.168.25.131	228.8.8.8	UDP	Source port: 1033 Destination port: 45566
15	30.118251	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
16	30.207289	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
17	30.400785	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
18	30.407729	192.168.25.131	228.8.8.8	UDP	Source port: 1033 Destination port: 45566
❄	❄	❄	❄	❄	❄
20	45.593701	192.168.25.131	228.8.8.8	UDP	Source port: 1033 Destination port: 45566
21	45.612956	192.168.25.131	228.8.8.8	UDP	Source port: 1033 Destination port: 45566
22	48.458657	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
23	48.734481	192.168.25.131	228.8.8.8	UDP	Source port: 1029 Destination port: 45566
24	51.434913	192.168.25.131	228.8.8.8	UDP	Source port: 1033 Destination port: 45566
25	51.554688	192.168.25.131	224.0.0.2	IGMP	V2 Leave Group
26	51.582727	192.168.25.131	224.0.0.2	IGMP	V2 Leave Group

Bons cotés et perspectives

- Bons cotés
  - Système fiable pour groupes dynamiques
  - Système très modulaire (OSI)
  - Vient avec 19 exemples et 117 tests
- Projets ouverts
  - Plugin Ethereal
  - Support Bluetooth
  - Compatibilité standards Java: JavaSpaces, ...
  - Répartiteur de charge HTTP “tout fait”
  - ...

Mauvais cotés

- Fonctionnalités absentes (présents chez les concurrents comme .NET):
  - Coalescence de paquets
  - Priorités
  - Options d'envoi par message
  - Sous-groupes
  - Emissions et réceptions en flux
  - Échange d'états
- JGroups n'a pas de logo!
- Manuel de l'utilisateur en partie vide

Questions ?

Questions ?

Références

- Site web JGroups et les liens de ce site (surtout les présentations JBoss)
- Divers sites web comme Wikipédia ou le site de Cisco pour les protocoles de groupe et de répartissement de charge TCP/IP
- Ethereal
- Cours de Didier Donsez sur Jini
- Librairie MSDN de Microsoft

Merci

Merci

Documents disponibles sur

http://scholar.alishomepage.com/Master/JGroups/