Multithreading of Kostka Numbers Computation for the BonjourGrid Meta-Desktop Grid Middleware – AOC Team –

Heithem Abbes[1,2] Franck Butelle[1], Christophe Cérin[1]

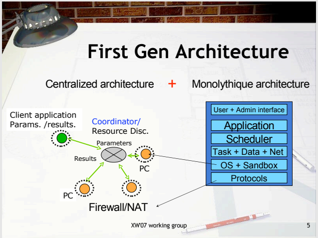
[1]Université de Paris 13, CNRS UMR 7030, Villetaneuse, France
[2]Research unit UTIC, ESSTT, Tunis, Tunisie

ICA3PP'2010, Busan

Table of contents

Desktop Grid Architectures

Desktop Grid

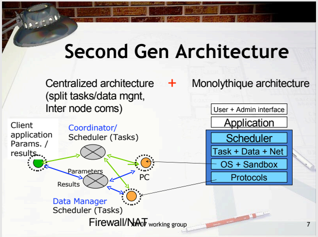


Key Points

- Federation of thousand of nodes;
- Internet as the communication layer: no trust!
- Volatility; local IP; Firewall

Desktop Grid Architectures

Desktop Grid



Future Generation (in 2006)

- Distributed Architecture
- Architecture with modularity: every component is “configurable”: scheduler, storage, transport protocole
- Direct communications between peers;
- Security;
- Applications coming from any sciences (e-Science applications)

In search of distributed architecture

First line: publish/subscribe system to notify and coordinate services and multiple DG without a central broker ⇒ BonjourGrid;

Second line: approach based on structured overlay network to discover (on the fly) the next node executing the next task ⇒ PastryGrid;

https://sourceforge.net/projects/pastrygrid/

(main contributions of Heithem Abbes in his PhD)

Main objectives of BonjourGrid

- Count on existing distributed tools for services discovering (publish/subscribe paradigm);

Main objectives of BonjourGrid

- Count on existing distributed tools for services discovering (publish/subscribe paradigm);
- Design and implement a platform able to manage multiple instances of DG middleware;

Main objectives of BonjourGrid

- Count on existing distributed tools for services discovering (publish/subscribe paradigm);
- Design and implement a platform able to manage multiple instances of DG middleware;
- Reduce as much as possible the use of any central element;

Main objectives of BonjourGrid

- Count on existing distributed tools for services discovering (publish/subscribe paradigm);
- Design and implement a platform able to manage multiple instances of DG middleware;
- Reduce as much as possible the use of any central element;
- Create a coordinator, on the fly, without any system administrator intervention; From a vision with a single coordinator towards a vision with multiple coordinators.

Main objectives of BonjourGrid

- Count on existing distributed tools for services discovering (publish/subscribe paradigm);
- Design and implement a platform able to manage multiple instances of DG middleware;
- Reduce as much as possible the use of any central element;
- Create a coordinator, on the fly, without any system administrator intervention; From a vision with a single coordinator towards a vision with multiple coordinators.
- Each coordinator searches, in a concurrent way, participants (idle machines)

Communities we serve

- A user can participate to any Boinc, Condor, XtremWeb project (enhanced 'scalability');

Communities we serve

- A user can participate to any Boinc, Condor, XtremWeb project (enhanced 'scalability');
- A user access in a uniform way the ressource of others;

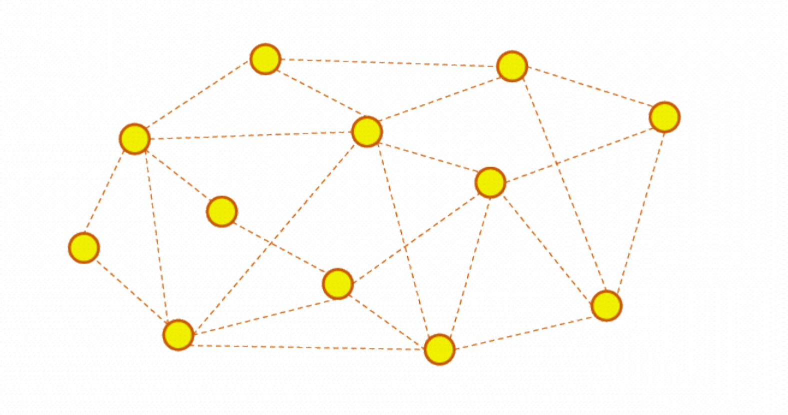
Communities we serve

- A user can participate to any Boinc, Condor, XtremWeb project (enhanced 'scalability');
- A user access in a uniform way the ressource of others;
- A user is not attached (prisoner - banking service) to a provider but can count on an universal open source protocol to choose the participants he wants or to become a slave for the servers he wants;

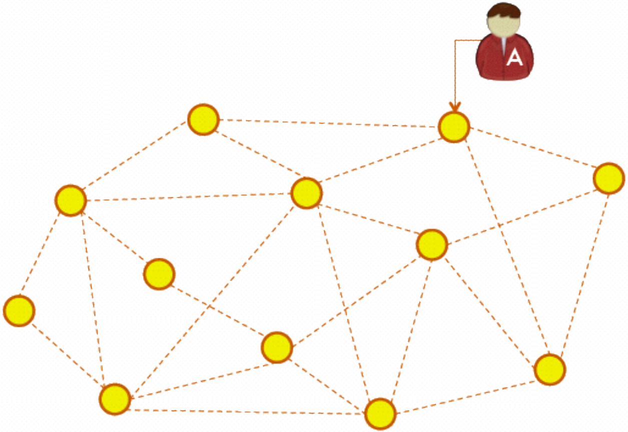
Communities we serve

- A user can participate to any Boinc, Condor, XtremWeb project (enhanced 'scalability');
- A user access in a uniform way the ressource of others;
- A user is not attached (prisoner - banking service) to a provider but can count on an universal open source protocol to choose the participants he wants or to become a slave for the servers he wants;
- This paper: a use case with BonjourGrid + parallelization of a computational problem;

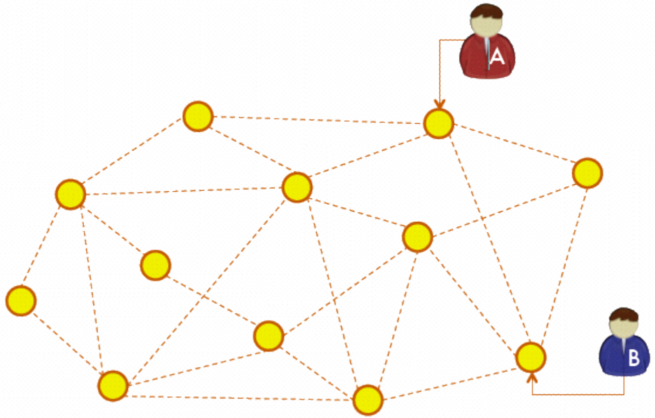
How BonjourGrid works



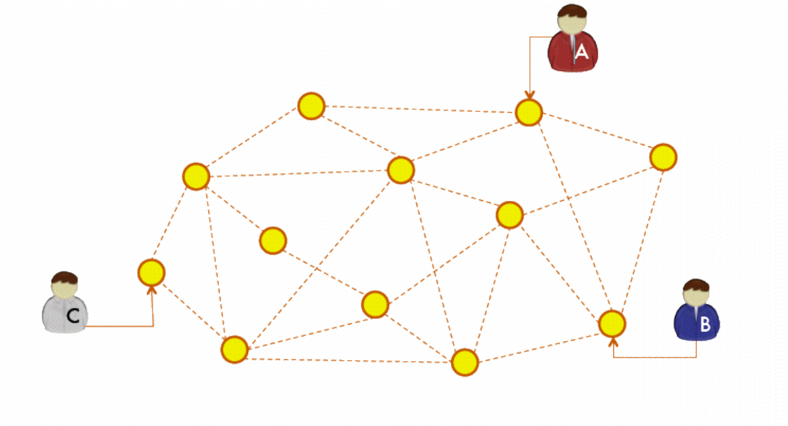
How BonjourGrid works



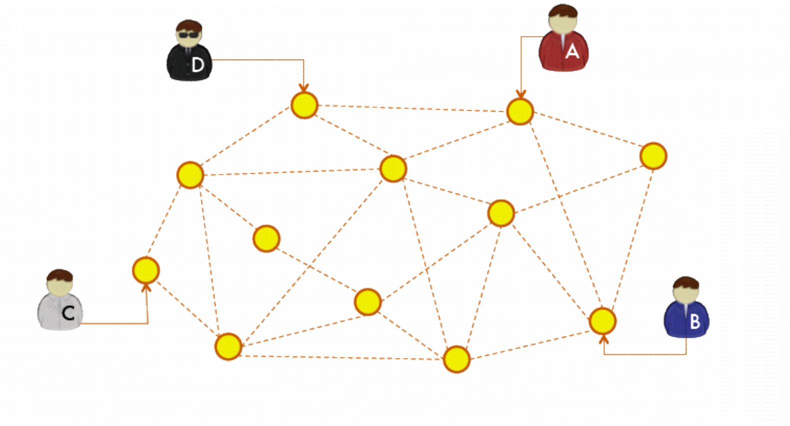
How BonjourGrid works



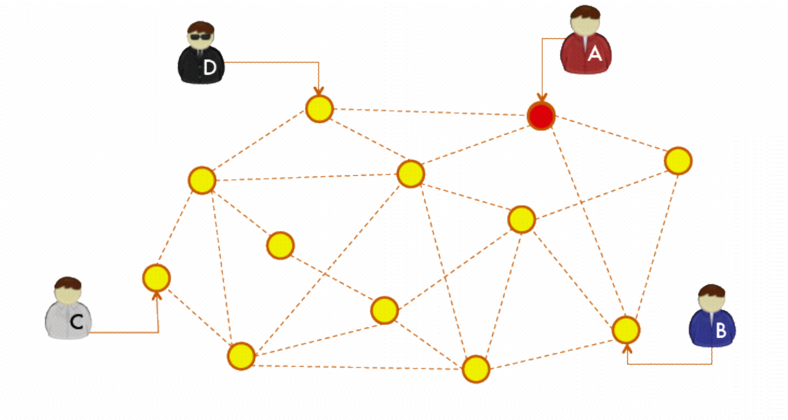
How BonjourGrid works



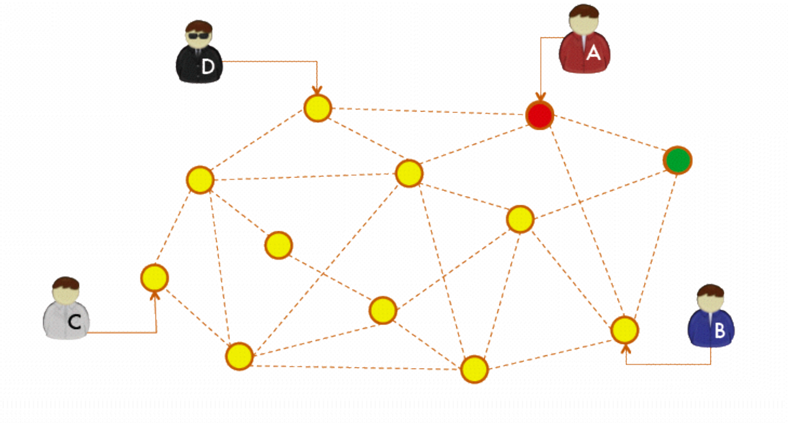
How BonjourGrid works



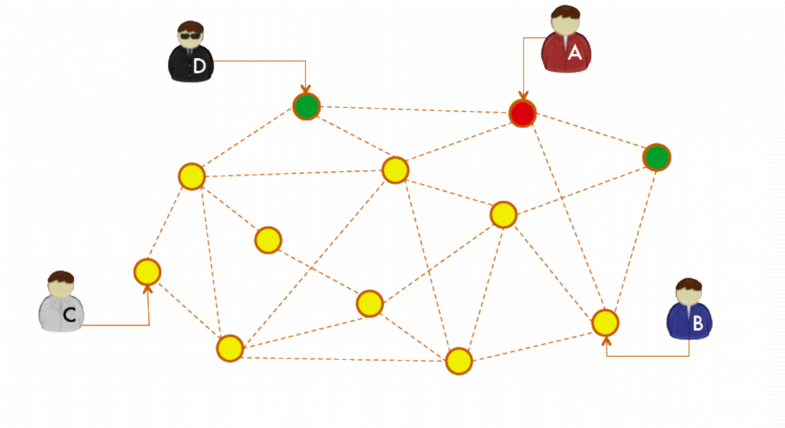
How BonjourGrid works



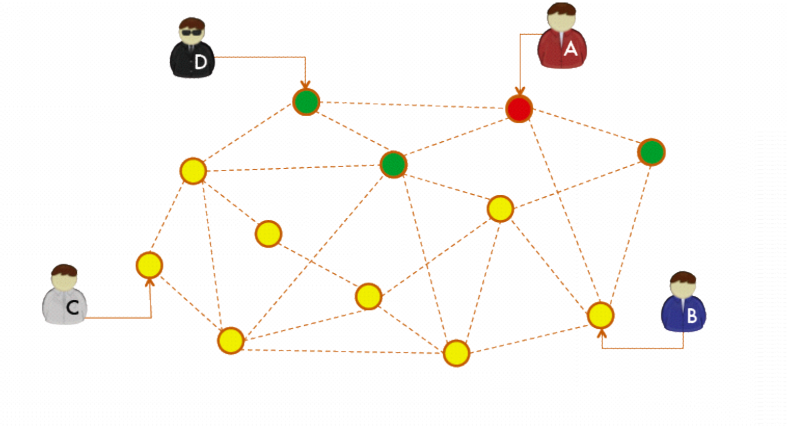
How BonjourGrid works



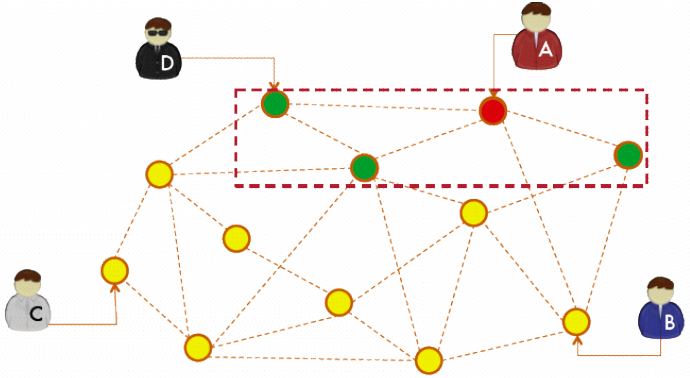
How BonjourGrid works



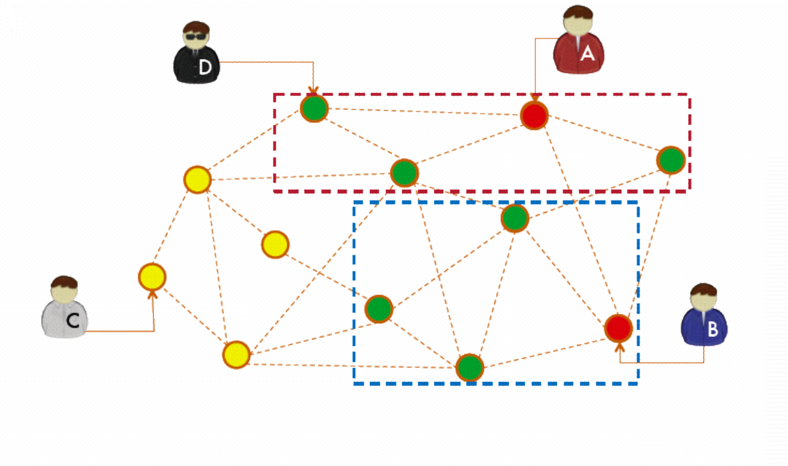
How BonjourGrid works



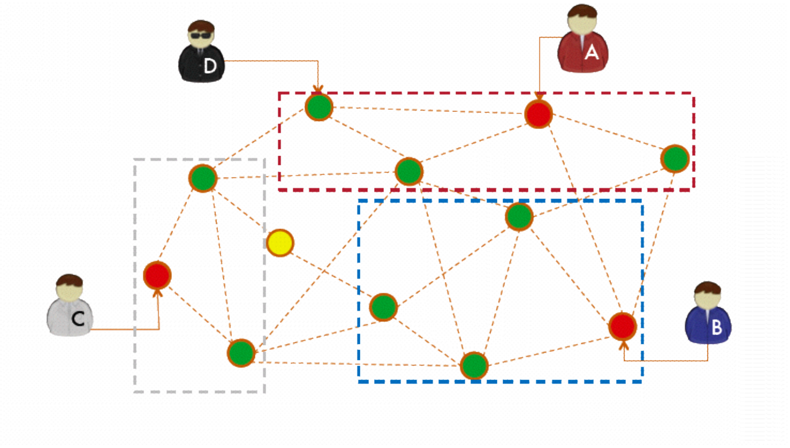
How BonjourGrid works



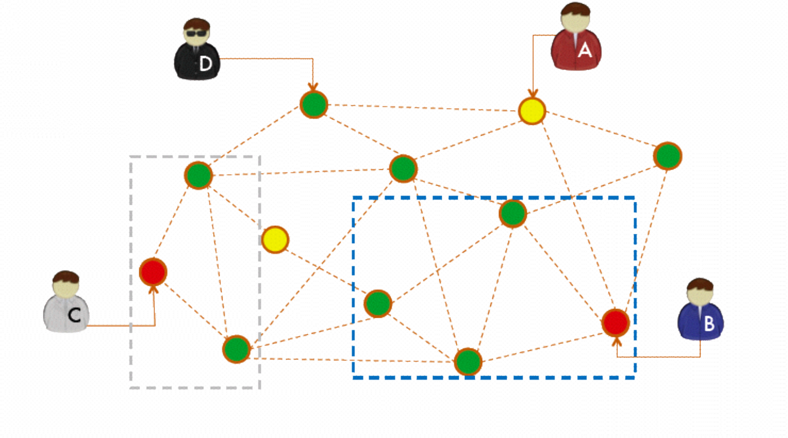
How BonjourGrid works



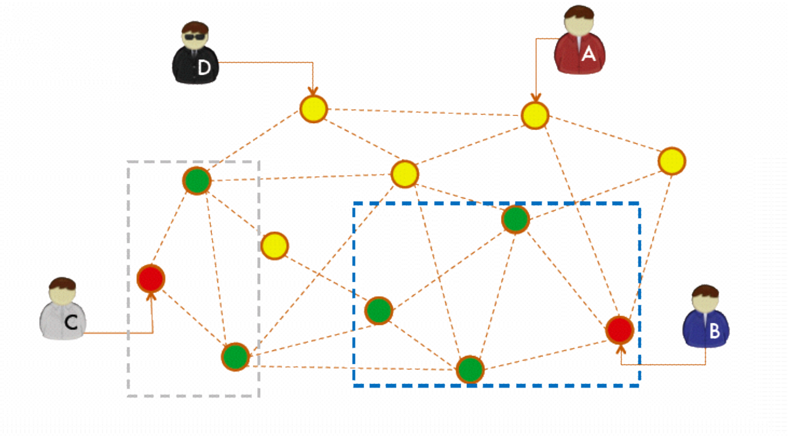
How BonjourGrid works



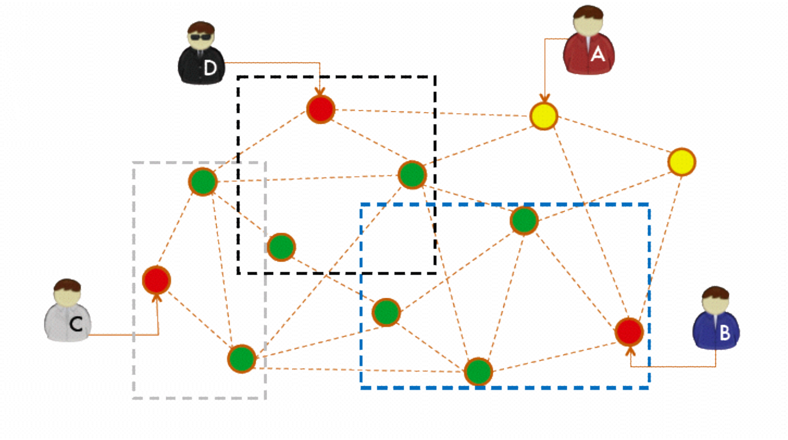
How BonjourGrid works



How BonjourGrid works



How BonjourGrid works



BonjourGrid vision

- The user requests for computation;

BonjourGrid vision

- The user requests for computation;
- The user provides the control flow graph, binaries, input data;

BonjourGrid vision

- The user requests for computation;
- The user provides the control flow graph, binaries, input data;
- The user deploys locally a coordinator and requests for participants; We support XtremWeb, Condor, Boinc.

BonjourGrid vision

- The user requests for computation;
- The user provides the control flow graph, binaries, input data;
- The user deploys locally a coordinator and requests for participants; We support XtremWeb, Condor, Boinc.
- The coordinator selects a set of machines (criteria: RAM, CPU, costs...)

BonjourGrid vision

- The user requests for computation;
- The user provides the control flow graph, binaries, input data;
- The user deploys locally a coordinator and requests for participants; We support XtremWeb, Condor, Boinc.
- The coordinator selects a set of machines (criteria: RAM, CPU, costs. . .)
- Upon completion, the coordinator returns to the idle state, slaves are freed and the coordination protocol:

BonjourGrid vision

- The user requests for computation;
- The user provides the control flow graph, binaries, input data;
- The user deploys locally a coordinator and requests for participants; We support XtremWeb, Condor, Boinc.
- The coordinator selects a set of machines (criteria: RAM, CPU, costs...)
- Upon completion, the coordinator returns to the idle state, slaves are freed and the coordination protocol:
  - manages and controls resources, services and computing elements;
  - does not depend on any specific machine nor any central element.

BonjourGrid vision

- The user requests for computation;
- The user provides the control flow graph, binaries, input data;
- The user deploys locally a coordinator and requests for participants; We support XtremWeb, Condor, Boinc.
- The coordinator selects a set of machines (criteria: RAM, CPU, costs. . .)
- Upon completion, the coordinator returns to the idle state, slaves are freed and the coordination protocol:
  - manages and controls resources, services and computing elements;
  - does not depend on any specific machine nor any central element.

The protocol for resources discovering

- Based on Bonjour from Apple;

The protocol for resources discovering

- Based on Bonjour from Apple;
- A publish/subscribe system is easy to use (toolbox = publish(), subscribe(), browse())

The protocol for resources discovering

- Based on Bonjour from Apple;
- A publish/subscribe system is easy to use (toolbox = publish(), subscribe(), browse())
- Bonjour is dedicated to LAN (wide area Bonjour? We need some experiments)

The protocol for resources discovering

- Based on Bonjour from Apple;
- A publish/subscribe system is easy to use (toolbox = publish(), subscribe(), browse())
- Bonjour is dedicated to LAN (wide area Bonjour? We need some experiments)
- Concerning the Wide Area implementation: we can also think about Apache Kandula (http://ws.apache.org/kandula/) or even cisco Jabber protocol (http://www.jabber.com):

The protocol for resources discovering

- Based on Bonjour from Apple;
- A publish/subscribe system is easy to use (toolbox = publish(), subscribe(), browse())
- Bonjour is dedicated to LAN (wide area Bonjour? We need some experiments)
- Concerning the Wide Area implementation: we can also think about Apache Kandula (http://ws.apache.org/kandula/) or even cisco Jabber protocol (http://www.jabber.com):
  - Extensible Messaging and Presence Protocol (XMPP) (formerly named Jabber) is an open, XML-based protocol originally aimed at near-real-time, extensible instant messaging (IM) and presence information, but now expanded into the broader realm of message-oriented middleware

The protocol for resources discovering

- Based on Bonjour from Apple;
- A publish/subscribe system is easy to use (toolbox = publish(), subscribe(), browse())
- Bonjour is dedicated to LAN (wide area Bonjour? We need some experiments)
- Concerning the Wide Area implementation: we can also think about Apache Kandula (http://ws.apache.org/kandula/) or even cisco Jabber protocol (http://www.jabber.com):
  - Extensible Messaging and Presence Protocol (XMPP) (formerly named Jabber) is an open, XML-based protocol originally aimed at near-real-time, extensible instant messaging (IM) and presence information, but now expanded into the broader realm of message-oriented middleware
- The current protocol has been developed/specified with 'ad-hoc' methods → we need to consolidate the trust (ongoing project to verify it, based on Colored Petri Nets)

Fault Tolerance with BonjourGrid

- Intrinsic property of any large scale system;

Fault Tolerance with BonjourGrid

- Intrinsic property of any large scale system;
- We assume that any coordinator is responsible for its FT (it manages the volatility of attached slaves)

Fault Tolerance with BonjourGrid

- Intrinsic property of any large scale system;
- We assume that any coordinator is responsible for its FT (it manages the volatility of attached slaves)
- Our solution: tolerate the failure of coordinators

Fault Tolerance with BonjourGrid

- Intrinsic property of any large scale system;
- We assume that any coordinator is responsible for its FT (it manages the volatility of attached slaves)
- Our solution: tolerate the failure of coordinators
  - For any application we create and manage dynamically copies of the coordinator;
  - We manage k copies; based on passive replication.
  - When a service disappears: we added a special status flag to distinguish between 'end of the application' / 'failure' ⇒ slaves can redirect the communication to a copy.

Intensive Experiments

- BonjourGrid has been tested intensively: stressed scenario to more relaxing scenario

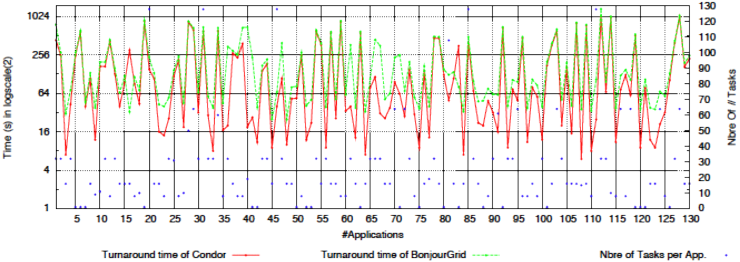
Intensive Experiments

- BonjourGrid has been tested intensively: stressed scenario to more relaxing scenario
  - in terms of #coordinator versus #nodes
  - in terms of using virtual machines to reach 1000 nodes;
  - in terms of comparing Boinc, Condor, XtremWeb over our protocol;
  - in terms of robustness in supporting FT;

Intensive Experiments

- BonjourGrid has been tested intensively: stressed scenario to more relaxing scenario
  - in terms of #coordinator versus #nodes
  - in terms of using virtual machines to reach 1000 nodes;
  - in terms of comparing Boinc, Condor, XtremWeb over our protocol;
  - in terms of robustness in supporting FT;
- Example Condor: 130 applications (2 to 128 // tasks), 200 nodes, application task: 1s to 500s. Result: with BonjourGrid, 35% of applications generate a delay of about 30s.

Experiments: one example



Problem Definition

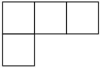
Integer Partitions and Ferrer's Diagramms

Definition : Integer Partition

write n as a sum of decreasing integers.

Example : $4 = 3 + 1 = 2 + 2 = 2 + 1 + 1 = 1 + 1 + 1 + 1$

Definition : Ferrer's Diagramm

of a $\lambda = (3, 1)$ partition of an integer. $|\lambda| = 4$: $F^{\lambda} =$ 

Important role :

- group representation theory
- Symmetric functions theory
- Frobenius (1849-1917) : irreductible representations of symmetric groups are indexed by integer partitions...

Ferrer diagrams and tableaux

- Each partition λ specifies a Ferrer diagram F^λ consisting of $|\lambda|$ boxes arranged in left-adjusted rows of lengths λ_i.
- A semistandard Young tableaux T of shape λ and weight $\beta(T)$ is a numbering of the boxes F^λ with $\beta_i(T)$ entries i for $i = 1, 2, \ldots, n$ that are weakly increasing across rows and strictly increasing down columns.

Example: $n = 6$, $\lambda = (4, 2, 2, 1)$, $\beta(T) = (1, 3, 1, 1, 2, 1)$

$F^\lambda =$

$T =$

1	2	2	5
2	4		
3	6		
5			

Calculation of Kostka coefficients

- Theorem: $K_{\lambda\beta}$ is the number of semistandard Young Tableau of shape λ and weight β
- Example: $n = 3$, $\lambda = (3, 2, 0)$, $\beta = (2, 1, 2)$.

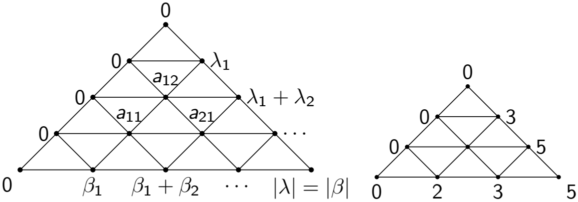
1	1	2
3	3	

1	1	3
2	3	

  Hence $K_{\lambda\beta} = 2$.
- Note: we compute the coefficients according to a dilatation N: stretched Kostka coefficient $K_{N\lambda,N\beta}$ and then we have to interpolate on the results to find the Kostka polynome in N;
- Another way to compute Kostka coefficients is using Hives model.

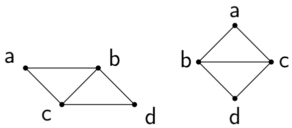
Integer Hives

- n-hive with vertex labels $a_{ij} \in \mathbb{Z}$ for $0 \leq i, j, i+j \leq n$
- Vertex labels increase from left to right.
- Example : $n = 3$, $\lambda = (3, 2, 0)$, $\beta = (2, 1, 2)$.



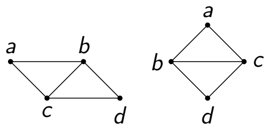
Hives Conditions

- Distinct types of rhombi, with vertex labels :



Hives Conditions

- Two distinct types of rhombi, with vertex labels:

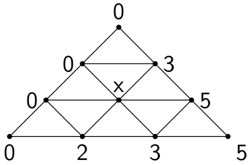


- Hive conditions in terms of vertex labels:

$$b + c \geq a + d$$

Simple example using K-hives

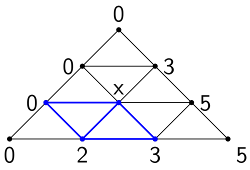
$n = 3$, $\lambda = (3, 2, 0)$, $\beta = (2, 1, 2)$ $(x \in \mathbb{Z})$.



Simple example using K-hives

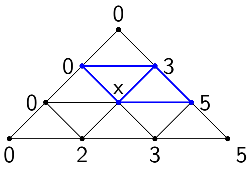
$n = 3$, $\lambda = (3, 2, 0)$, $\beta = (2, 1, 2)$ $(x \in \mathbb{Z})$.

- $x + 2 \geq 3 \Leftrightarrow x \geq 1$



Simple example using K-hives

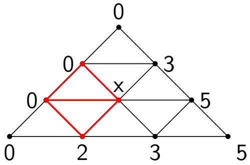
$n = 3$, $\lambda = (3, 2, 0)$, $\beta = (2, 1, 2)$ $(x \in \mathbb{Z})$.



- $x \geq 1$
- $x + 3 \geq 5 \Leftrightarrow x \geq 2$

Simple example using K-hives

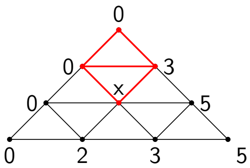
$n = 3$, $\lambda = (3, 2, 0)$, $\beta = (2, 1, 2)$ $(x \in \mathbb{Z})$.



- $x \geq 2$
- $x \geq 2$

Simple example using K-hives

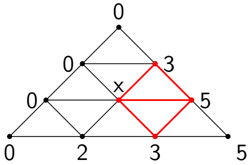
$n = 3$, $\lambda = (3, 2, 0)$, $\beta = (2, 1, 2)$ $(x \in \mathbb{Z})$.



- $x \geq 2$
- $3 \geq x$

Simple example using K-hives

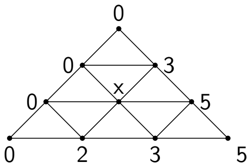
$n = 3$, $\lambda = (3, 2, 0)$, $\beta = (2, 1, 2)$ ($x \in \mathbb{Z}$).



- $x \geq 2$
- $3 \geq x$
- $x + 5 \geq 6$

Simple example using K-hives

$n = 3$, $\lambda = (3, 2, 0)$, $\beta = (2, 1, 2)$ ($x \in \mathbb{Z}$).



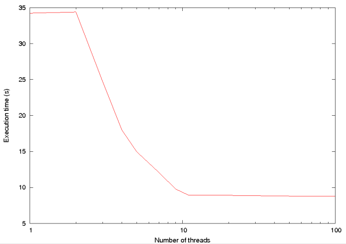
- Finally $3 \geq x \geq 2$: 2 solutions so $K_{\lambda\beta} = 2$.
- But when there is more than one free variable, enumeration must be done.

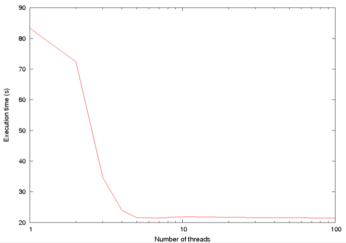
Experiments

Tools & Results

- Pthreads library: portability;
- OS: Linux;
- CPU: Bi-AMD Opteron dual core at 2.8GHz;
- Results: performance depends on the size of first interval ($n \geq 1 \Rightarrow n$ threads at the beginning) ; The number of feasible values tried by each thread depends on the interval it received from the first step; Input dependant;

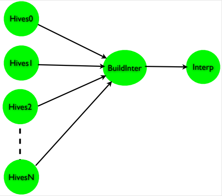
Graphical representation





The view of the user

GUI and the engineering part




```xml
<Deployment>
    <Application ApplicationDescription="monapp" Client="heithem">
        <Module ModuleDescription="hives" ModuleDirIn="/bin/">
            <Binary BinaryCpuType="amd64" BinaryExecutable="hives"
                    BinaryOsName="Linux" BinarySubFolder="linux"/>
        </Module>
        <Module ModuleDescription="buildinter" ModuleDirIn="/bin/">
            <Binary BinaryCpuType="amd64" BinaryExecutable="buildinter"
                    BinaryOsName="Linux" BinarySubFolder="linux"/>
        </Module>
        <Module ModuleDescription="interp" ModuleDirIn="/bin/">
            <Binary BinaryCpuType="amd64" BinaryExecutable="interp"
                    BinaryOsName="Linux" BinarySubFolder="linux"/>
        </Module>
    </Application>
```

Facilities

- SDAD: system for deployment; No need for learning XML syntax;
- Programming language: Python, Bonjour for Python;
- Embedded code: x86 executable, javac, scripts (Perl, Python, Bash...) ; Depending on the DG used.
- What you need to install: package containing one DG middleware (Condor, Boinc, XtremWeb) + client + server;

The view of the user

GUI and the engineering part

```
<Table>
    <Task Application="monapp" ApplicationModule="hives"
     Description="hives1" DirIn="/hives/" FileIn="dir.zip" Final="0">
        <Input InputName="x1"/>
        <Output OutputName="y1"/>
        <cmdLine>11,10,8,5 20,17,3 26,25,8,8,7 out1 1</cmdLine>
        <QoS>
           <Power NbMax="0" ErrorMargin="0.2">3000.0</Power>
        </QoS>
    </Task>

    <Task Application="monapp" ApplicationModule="hives"
      Description="hives2" DirIn="/hives/" FileIn="dir.zip" Final="0">
        <Input InputName="x2"/>
        <Output OutputName="y2"/>
        <cmdLine>11,10,8,5 20,17,3 26,25,8,8,7 out2 2 </cmdLine>
        <QoS>
           <Power NbMax="0" ErrorMargin="0.2">3000.0</Power>
        </QoS>
    </Task>

    <Task Application="monapp" ApplicationModule="buildinter"
       Description="build" DirIn="/hives/" FileIn="dir.zip" Final="0">
        <Input InputName="y1"/>
        <Input InputName="y2"/>
        <Output OutputName="y3"/>
        <cmdLine>2 out interresult  </cmdLine>
        <QoS>
           <Power NbMax="0" ErrorMargin="0.2">3000.0</Power>
        </QoS>
    </Task>

    <Task Application="monapp" ApplicationModule="interp"
      Description="interp" DirIn="/hives/" FileIn="dir.zip" Final="1">
        <Input InputName="y3"/>
        <Output OutputName="y4"/>
        <cmdLine>interresult 2 finalresult </cmdLine>
        <QoS>
          <Power NbMax="0" ErrorMargin="0.2">3000.0</Power>
        </QoS>
    </Task>

</Table>
</Deployment>
```

Facilities

- SDAD: system for deployment; No need for XML learning;
- In this example: command line parameters for tasks ; QoS parameters: helping the schedular for selecting machines ⇒ information generated automatically.

Conclusion

From similar to diversity in large scale experiments

- Contrarily to previous works, we do not attempt to exploit only one DG middleware → coordination; The user may select his favorite tool (we do not impose anything) for computing;

Conclusion

From similar to diversity in large scale experiments

- Contrarily to previous works, we do not attempt to exploit only one DG middleware → coordination; The user may select his favorite tool (we do not impose anything) for computing;
- A full description of steps necessary for using BonjourGrid; A use case in E-science;
- New multithreaded code for the computation of Kostka Numbers;

Conclusion

From similar to diversity in large scale experiments

- Contrarily to previous works, we do not attempt to exploit only one DG middleware → coordination; The user may select his favorite tool (we do not impose anything) for computing;
- A full description of steps necessary for using BonjourGrid; A use case in E-science;
- New multithreaded code for the computation of Kostka Numbers;
- Experiments showed the easy to use approach ; Different contexts for the executions;

Conclusion

From similar to diversity in large scale experiments

- Contrarily to previous works, we do not attempt to exploit only one DG middleware → coordination; The user may select his favorite tool (we do not impose anything) for computing;
- A full description of steps necessary for using BonjourGrid; A use case in E-science;
- New multithreaded code for the computation of Kostka Numbers;
- Experiments showed the easy to use approach ; Different contexts for the executions;
- Future work for BonjourGrid: reservation rules; wide area Bonjour or XMPP (Jabber protocol for presence) or Web services ; Formal verification of the protocol.

Multithreading of Kostka Numbers Computation for the BonjourGrid Meta-Desktop Grid Middleware – AOC Team –

Heithem Abbes[1,2] Franck Butelle[1], Christophe Cérin[1]

[1]Université de Paris 13, CNRS UMR 7030, Villetaneuse, France
[2]Research unit UTIC, ESSTT, Tunis, Tunisie

ICA3PP'2010, Busan