Orleans: Cloud Computing for Everyone

Sergey Bykov, Alan Geller, Gabriel Kliot, James Larus, Ravi Pandya, Jorgen Thelin

eXtreme Computing Group, Microsoft Research

SOCC, October 27th, 2011
Cloud Programming Must Get Simpler!

- SW Design GUI
- OOP MVC
- Server Applications
- Java .NET
- Web
- JSP ASP.NET Ruby on Rails
- Cloud
- Orleans
Orleans = Programming Model + Distributed Runtime

Applications

Orleans
Programming Model
Distributed Runtime

- Simplified Programming Model
- Transparent Scalability
- Adaptive Performance Management

.NET + Azure
Grains

Shopping Cart Grain

<table>
<thead>
<tr>
<th>Behavior</th>
<th>State</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buy(...)</td>
<td>Total Price</td>
<td>$1300</td>
</tr>
<tr>
<td></td>
<td>Products</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Customer</td>
<td></td>
</tr>
</tbody>
</table>

Customer Grain

<table>
<thead>
<tr>
<th>Behavior</th>
<th>State</th>
</tr>
</thead>
</table>

Product X Grain

<table>
<thead>
<tr>
<th>Behavior</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumAvailable()</td>
<td></td>
</tr>
<tr>
<td>Buy(...)</td>
<td></td>
</tr>
<tr>
<td>AddReview(...)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>“Canon EOS T3i”</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKU</td>
<td>B004J3V90Y</td>
</tr>
<tr>
<td>Quantity</td>
<td>12345</td>
</tr>
<tr>
<td>Price</td>
<td>$800</td>
</tr>
</tbody>
</table>

Product Y Grain

<table>
<thead>
<tr>
<th>Behavior</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumAvailable()</td>
<td></td>
</tr>
<tr>
<td>Buy(...)</td>
<td></td>
</tr>
<tr>
<td>AddReview(...)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>“Sony TC412”</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKU</td>
<td>X0322D12</td>
</tr>
<tr>
<td>Quantity</td>
<td>315</td>
</tr>
<tr>
<td>Price</td>
<td>$500</td>
</tr>
</tbody>
</table>
Asynchronous Communication and Promises

Promise

Buy(…)

Reply

Closure can Execute
Simplified Error Handling

Grain A

Grain B

Grain C

Success Closure

Error Closure
Single Threaded Execution Model

Message Queue

Grain A

<table>
<thead>
<tr>
<th>Behavior</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Execution Queue
### Activations

#### NumAvailable()
- Quantity: 1231
- Name: "Canon EOS T3i"
- SKU: B004J3V90Y

#### Buy(...)
- Quantity: 1231
- Name: "Canon EOS T3i"
- SKU: B004J3V90Y

---

### State Reconciliation

- Store A0
- Branch A1
- Branch A2
- Merge
What could go wrong?

- Failure
  - C
  - D

- Interleaving
  - A
  - B
  - D'

- Inconsistency
  - E
Lightweight Transactions
Adaptive Runtime

Silo

Silo
Applications and Grain Size

Chirper
Simplified Twitter-like system

User

Messaging intensive

Horton
Distributed graph database

Data intensive

PageRank
Large sparse matrices (Billions x Billions)

Compute intensive

Graph Partition

Compute Worker
Evaluation

- 200 lines of code
- Near linear scalability
Related Work

- Actor models
  - Erlang
  - E
  - Thorn

- Distributed Objects
  - EJB
  - CORBA

- Transactions
- Futures/Promises
  - [Liskov et al]

- Conflict resolution
  - CRDTs [Marc Shapiro]
  - Concurrent Revisions [Burckhardt & Leijen]
Conclusion

Orleans = Programming Model + Distributed Runtime

- Simplified Programming Model
- Transparent Scalability
- Adaptive Performance Tuning

Make cloud programming accessible to everyone!