DryadLINQ: A System for General-Purpose Distributed Data-Parallel Computing Using a High-Level Language

D F B
B 1 G 0 G
Microsoft Research Silicon Valley [1]*joint affiliation, Reykjavík University, Iceland*

Abstract

D D E A D E D D D D 10^{12} B 319 240- , 960-

1 Introduction

D D

E

A

(6.1) 15 D 26 E D

D (2 E) C#, B, F#. D E D G

15, 26, 31, 34, 35 .

14 .

15

0

E .

32 , 31 ,

F H E.

H

25

virtualized

21, 24, 27, 28, 29 . A

:

2

3

4

5

6

7

2 System Architecture

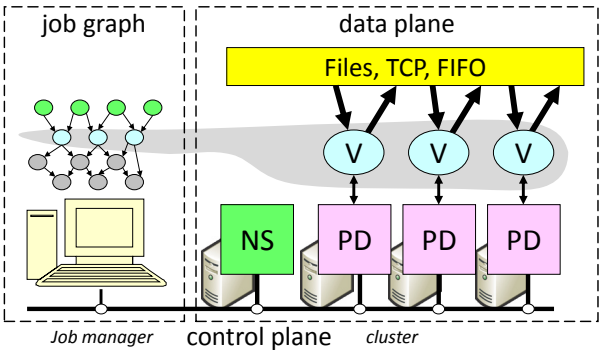
26 . A

A

E 1

: (1) ; (2) ; (3) ; (4) ; (5)

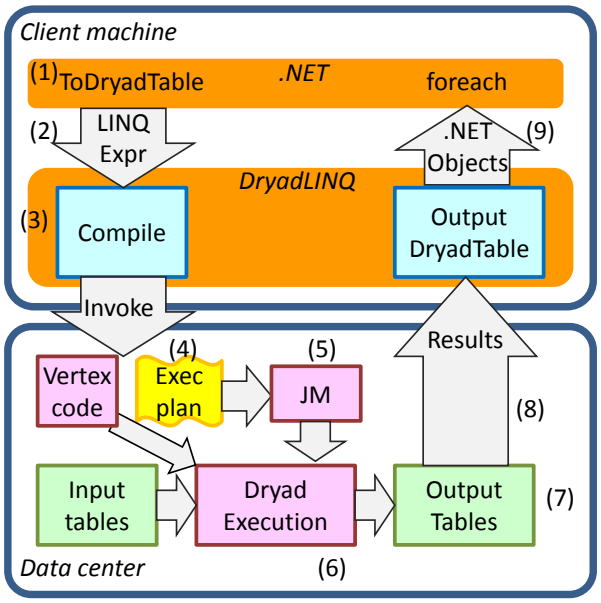
A



F 1: *Dryad system architecture. NS is the name server which maintains the cluster membership. The job manager is responsible for spawning vertices (V) on available computers with the help of a remote-execution and monitoring daemon (PD). Vertices exchange data through files, TCP pipes, or shared-memory channels. The grey shape indicates the vertices in the job that are currently running and the correspondence with the job execution graph.*

2.1 DryadLINQ Execution Overview

F 2 D



F 2: *LINQ-expression execution in DryadLINQ.*

Step 1. A . E D B (3).

Step 2. ToDryadTable D

Step 3. D D ; ( ).

D ( ) 4

Step 4. D , D , D

Step 5. 3. F 1.

Step 6. E D (3).

Step 7. D ( ).

Step 8. D D DryadTable D DryadTable

Step 9. G DryadTable E .

Step 10. D 2. 9.

3 Programming with DryadLINQ

D 2 D 1 A 38

3.1 LINQ

2 E C#. D E F#. IEnumerable<T>. F

T
IQueryable<T>
IEnumerable<T>
( )
: ( )
IEnumerable
;
( ) D
IQueryable
IQueryable
A IQueryable

F 3

0
scoreTriples
QueryScoreDocIDTriple.

38 .

C#
Join
0 (staticRank)
0
0
var
( )

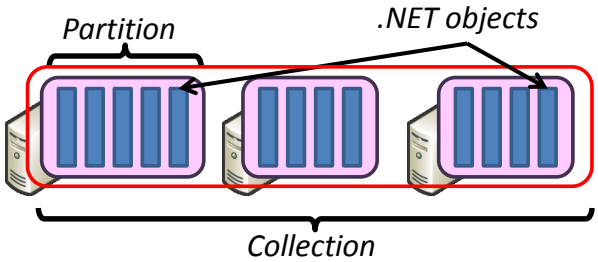
3.2 DryadLINQ Constructs

D
D
D

```
// SQL-style syntax to join two input sets:
// scoreTriples and staticRank
var adjustedScoreTriples =
  from d in scoreTriples
  join r in staticRank on d.docID equals r.key
  select new QueryScoreDocIDTriple(d, r);
var rankedQueries =
  from s in adjustedScoreTriples
  group s by s.query into g
  select TakeTopQueryResults(g);

// Object-oriented syntax for the above join
var adjustedScoreTriples =
  scoreTriples.Join(staticRank,
    d => d.docID, r => r.key,
    (d, r) => new QueryScoreDocIDTriple(d, r));
var groupedQueries =
  adjustedScoreTriples.GroupBy(s => s.query);
var rankedQueries =
  groupedQueries.Select(
    g => TakeTopQueryResults(g));
```

F 3: *A program fragment illustrating two ways of expressing the same operation. The first uses LINQ's declarative syntax, and the second uses object-oriented interfaces. Statements such as* `r => r.key` *use C#'s syntax for anonymous lambda expressions.*



F 4: *The DryadLINQ data model: strongly-typed collections of .NET objects partitioned on a set of computers.*

. E D
F 4.
18 .
D
DryadTable<T>,
IQueryable<T>.
DryadTable<T>
F
. DryadTable
D

4.

all the functions called in DryadLINQ expressions must be side-effect free.

GetTable<T>
ToDryadTable<T>.:

```
var input = GetTable<LineRecord>("file://in.tbl");
var result = MainProgram(input, ...);
var output = ToDryadTable(result, "file://out.tbl");
```

ToDryadTable

DryadTable

HashPartition<T,K> RangePartition<T,K>.

Apply Fork,

Apply f

Apply

$[i, i + d]$

d.

7. Fork

Apply

f, Apply Fork)

f

Apply

Apply

A Apply E

Resource

Apply

3.3 Building on DryadLINQ

C#

15

):

```
public static MapReduce( // returns set of Rs
    source, // set of Ts
    mapper, // function from T → Ms
    keySelector, // function from M → K
    reducer // function from (K,Ms) → Rs
) {
    var mapped = source.SelectMany(mapper);
    var groups = mapped.GroupBy(keySelector);
    return groups.SelectMany(reducer);
}
```

4

5.

14 ;

C#

38

4 System Implementation

3 2.

25 .

4.1 Execution Plan Graph

(E G),

E G

E G

Fork E G

E G. E G

E G

( ).

E G

E

4.2 DryadLINQ Optimizations

4.2.1 Static Optimizations

E G

:

Pipelining:

Removing redundancy:

Eager Aggregation:

OrderBy

4.2.3.

E G

G

: input.Select(x => f(x)). f

x.name,

H f

I/O reduction: [illegible]

4.2.2 Dynamic Optimizations

[illegible]

4.2.3 Optimizations for `OrderBy`

[illegible]

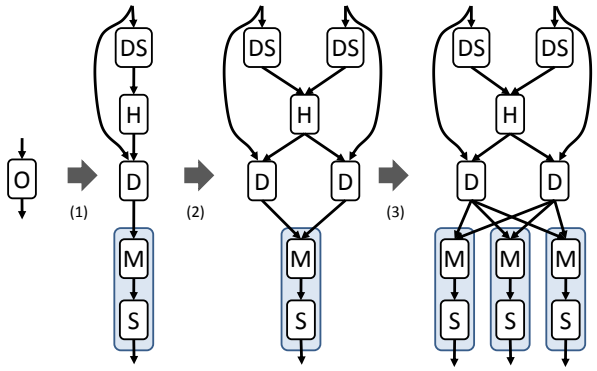


Figure 5: *Distributed sort optimization described in Section 4.2.3. Transformation (1) is static, while (2) and (3) are dynamic.*

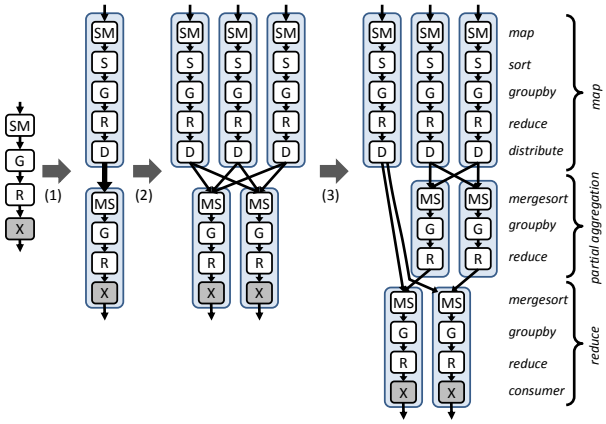


Figure 6: *Execution plan for MapReduce, described in Section 4.2.4. Step (1) is static, (2) and (3) are dynamic based on the volume and location of the data in the inputs.*

[illegible]

4.2.4 Execution Plan for MapReduce

[illegible]

4.2.3, (2)

(3)

15

26

all

4.3 Code Generation

(1)

(2)

(3)

(4)

4.4 Leveraging Other LINQ Providers

19

25

4.5 Debugging

4.4,

A D

D

D

5 Experimental Evaluation

D . A

5.1. D

D

5.1 Hardware Configuration

240 . E 2003 64-
A D 2218 HE C
2.6 GH , 16 GB DD 2
750 GB A A

2048
48 GB. E GB.
E 29 31
E
2048
6 802.3

6 GB
$1000

5.2 Terasort

D
3 . 10
100-B 10-B
3 . D
DryadTable
OrderBy;
4.2.3 (
).
F 3.87 GB
(4, 166, 666, 600 B).
n
$3.87n$ GB .
$n = 240$ 10^{12} B
H
D
10^{12} B , 150 GB

1 1 240,
3.87 GB 10^{12} B .
5%
. F 7
F
D
. F $2 \leq n \leq 20$
$n > 20$

G	1	2	10	20	40	80	240
	119	241	242	245	271	294	319

1: n- $3.87n$ GB
10^{12} B $n = 240$.

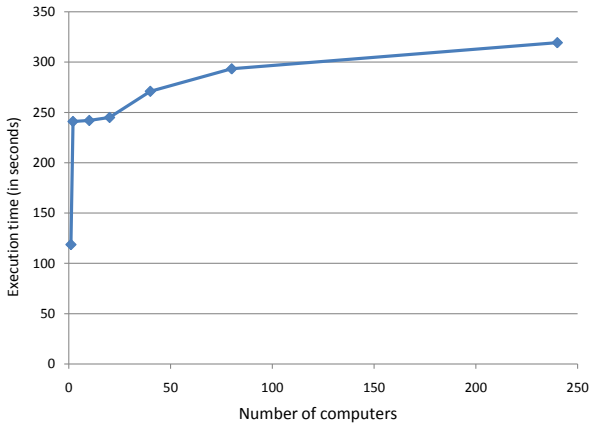


Figure 7: *Sorting increasing amounts of data while keeping the volume of data per computer fixed. The total data sorted by an n-machine experiment is around $3.87n$ GBytes, or 10^{12} Bytes when $n = 240$.*

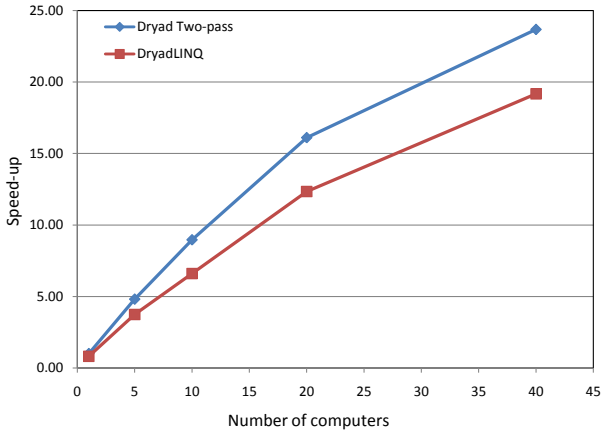


Figure 8: *The speed-up of the Skyserver Q18 computation as the number of computers is varied. The baseline is relative to DryadLINQ job running on a single computer and times are given in Table 2.*

G	1	5	10	20	40
D	2167	451	242	135	92
D	2666	580	328	176	113

2: 18

10%

$n \leq 20$

$n = 40$

5.3 SkyServer

(18) D

23

0 11.8 GB

41.8 GB

D 26

D D 1000

C++ D

100 C#.

40

n,

n / $40/n$

2

D D n

1 40.

3.5% D

1.3

D

5.4 PageRank

D

E

A

E

0

G B

D

( ).

0

93,
35,

H
80%-90%
(
);
F
(+7 C),
(+21 C),
(+18 C).
38 .
(
240)
954 16.5B 1.2 B
10 12,792
116 GB. 10
690.

H
0
( ).
B. D
0
0

5.5 Large-Scale Machine Learning

D

2.1 GB.

3 3.3
160, C#. : (1)
; (2) ; (3)
E
10 (5
5 C
( ), /
10,000,
240 7
11 5
C .

D
33
(514). F
11
(180 GB.
40 GB.).
D
(host,hour)
E
(95%),
174,588
4 22
10 C .

6 Related Work

D

D

D

D .

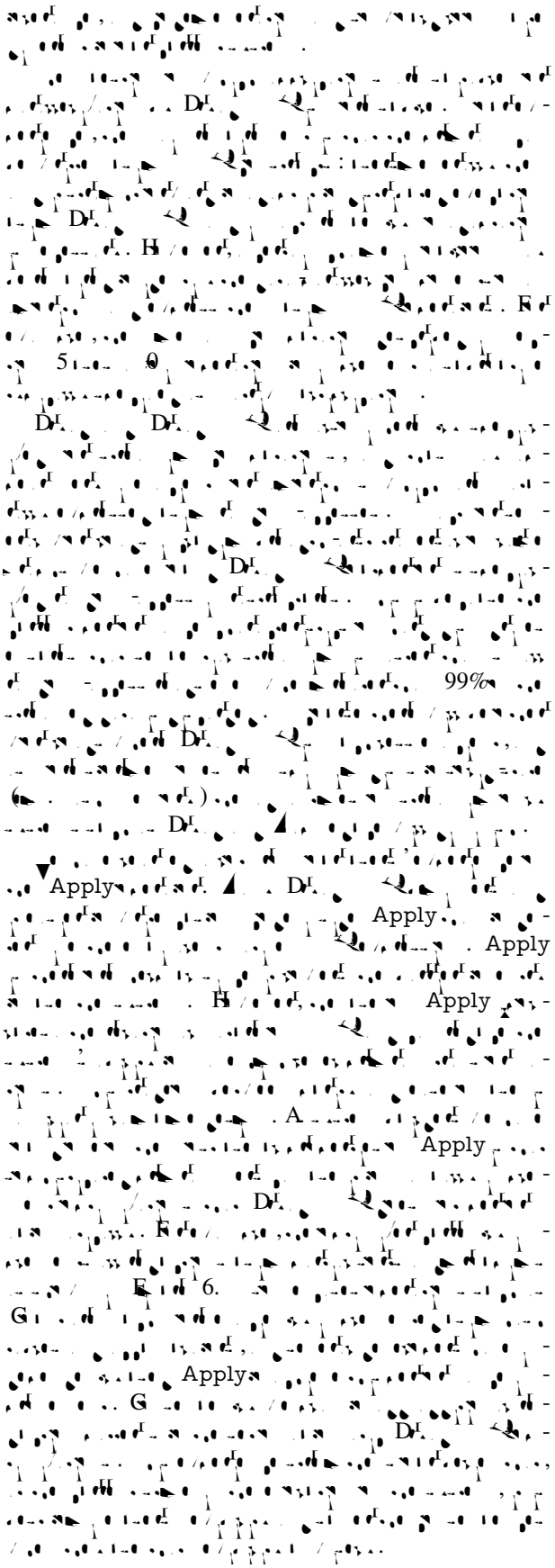
6.1 Parallel Databases

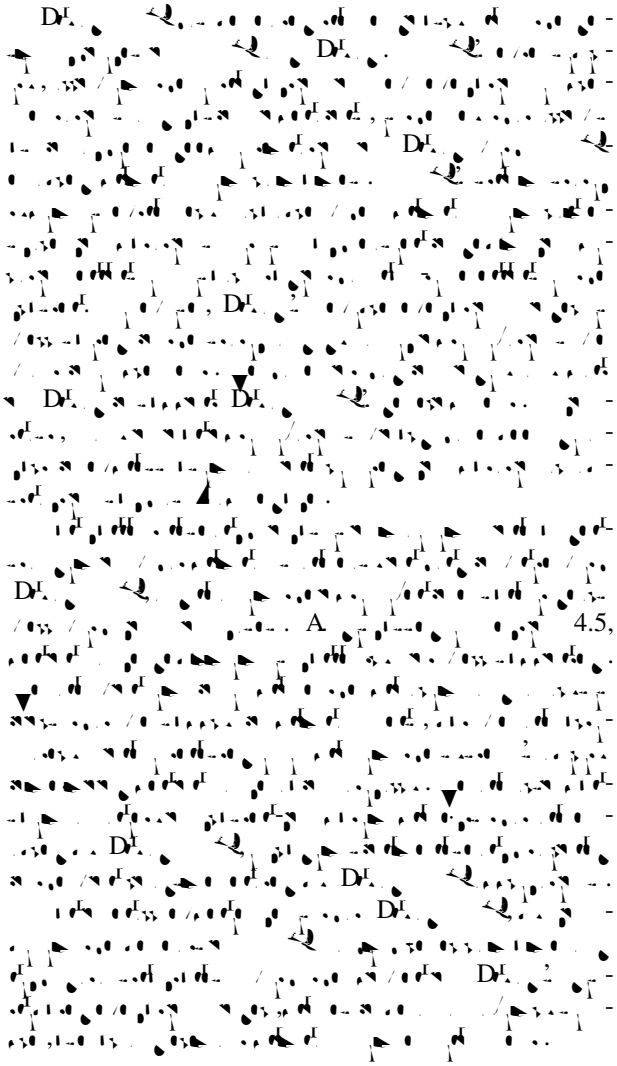
D (

6.2 Large Scale Data-Parallel Computation Infrastructure

6.3 Declarative Programming Languages

7 Discussion and Conclusions





Acknowledgements

F
B , D D , C
A / B , G
A D

References

1 D
://
/D /.

2 //
/ / /.

3
://
/ / B /.

4 B , C. ., F EC EA , G., G A , A., H A , H., 0H -
G A , A., AD A ABHA , ., C E A D, G. ., A D -

, . G. DB2, . *IBM Systems Journal 34*, 2, 1995.

5 BEC , ., D GA A, 0., A D A , 0. . /D: A . *International Parallel and Distributed Processing Symposium (IPDPS)*, 2005.

6 B E CH, G. E. . *Communications of the ACM (CACM) 39*, 3, 1996.

7 B FE, . D., 0 E G, C. F., A , B., E E , C. E., A DA , . H., A D H , . C : A . *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)*, 1995.

8 B A , H., A E A DE , ., C A , ., C E A D, G., DA F H, ., F A , ., HA , B., H, ., A D A D E , . B , . *IEEE Trans. on Knowl. and Data Eng. 2*, 1, 1990.

9 CA AHA , 0., A D DEC E, D. : A . *W4: Learning in Web Search*, 2005.

10 CHA E , ., 0E , B., A , ., A E , B., HA B, D., EA E , ., A D H , 0. C E: E . *International Conference of Very Large Data Bases (VLDB)*, 2008.

11 CHA G, F., DEA , 0., GHE A A , ., H EH, . C., A - ACH, D. A., B , ., CHA D A, ., F E , A., A D G BE , . E. B : A . *Symposium on Operating System Design and Implementation (OSDI)*, 2006.

12 CH H A G, H., DA DA , A., H A , - ., A D A E , D. . : . *SIGMOD international conference on Management of data*, 2007.

13 CH , ., A G, H., A G, ., A D HE , . . *Symposium on Principles and practice of parallel programming (PPoPP)*, 2003.

14 C A E , ., DAGE E, B., A D GH H, B. . *ACM SIGMOD*, 2004.

15 DEA , 0., A D GHE A A , . : . *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI)*, 2004.

16 DE H A DE, A., E , ., A D A A , . A . *Foundations and Trends in Databases 1*, 1, 2007.

17 DE , D., GHA DEHA ADEH, ., CH E DE , D., H A , H., B C E , A., A D A E , . G . *IEEE Transactions on Knowledge and Data Engineering 2*, 1, 1990.

18 DE , D., A D G A , 0. : . *Communications of the ACM 36*, 6, 1992.

19 D FF , 0. A . *Proceedings of the 2007 workshop on Declarative aspects of multicore programming*, 2007.

20 E G E , ., G A E, ., A D HA A , . : A /A . *Sigmod Record*, 1995.

21 FE G, ., , H., A , . C., A D G, A. . H. B : A . *International Conference on Extending Database Technology*, 1998, H.-0 , F. , G. A , E., . 1377. *Lecture Notes in Computer Science*.

22 G AEFE, G. E . *SIGMOD International Conference on Management of data*, 1990.

23 G A , 0., A A , A., HA A , A., , ., GH , C., , D., A D A DE BE G, 0. D. D . *Distributed Data and Structures 4: Records of the 4th International Meeting*, 2002.

24 HA A , ., F E C , D., A D A D E , . . *SIGMOD Rec. 25*, 3, 1996.

25 HE E E , 0. ., EB A E , ., A D HA , 0. A . *Foundations and Trends in Databases 1*, 2, 2007.

26 A D, ., B D , ., , ., B E , A., A D FE - E , D. D : D . *Proceedings of European Conference on Computer Systems (EuroSys)*, 2007.

27 AB A, ., A D DE , D. 0. E . *SIGMOD International Conference on Management of Data*, 1998.

28 A , D. . *ACM Comput. Surv. 32*, 4, 2000.

29 A , F., A D HA E A , A. D. . *Symposium on Applied computing (SAC)*, 2002.

30 , ., G EE D, ., ADD , ., FE , 0., G E , ., G B E, C., H , D., A , D., , ., D, ., C C , . ., E GE , ., A , A., A D E, C. : . *Concurrency and Computation: Practice and Experience 18*, 10, 2005.

31 , C., EED, B., A A A, ., A , ., A D , A. : A . *International Conference on Management of Data (Industrial Track) (SIGMOD)*, 2008.

32 E, ., D A D, ., G E E E , ., A D A , . : . *Scientific Programming 13*, 4, 2005.

33 A A, A., G D D , ., ACC C , 0., BA HA , ., B AC , ., AAC , ., A D E , . C - : . *International Conference on Uncertainty in Artificial Intelligence*, 2008.

34 EB A E , ., BEA , C., E E E , ., CHE - AC , ., GE, ., HACHE , ., HA , ., F E , 0., GE , 0., A D D , . ? 2: B . *Conference on Innovative Data Systems Research (CIDR)*, 2005.

35 EB A E , ., ADDE , ., ABAD , D. 0., HA - , ., HACHE , ., A D HE A D, . . *International Conference of Very Large Data Bases (VLDB)*, 2007.

36 A , ., H E D , ., A G, ., A D HA , A. *Workflows for e-Science*. 2007, . E - : A A . 320. 339.

37 DE , ., D , H.- ., A D , . H . *Journal of Functional Programming 12*, (4&5), 2002.

38 , ., A D, ., FE E , D., B D , ., E G , ., G DA, ., C E , 0., C HE , F., A D ACHA , . D . -2008-74, , 2008.

39 HA , ., HA EGA , ., C FF D, B., F E , ., A E , G., EFED A, ., A C , ., EF- A , ., A D DE, . : F . *IEEE Congress on Services*, 2007.