SDPA Project:
Solving Large-scale Semidefinite Programs

12/May/2008

Katsuki Fujisawa
Chuo University, Japan
Masakazu Kojima & Mituhiro Fukuda & Kazuhide Nakata & Makoto Yamashita
Tokyo Institute of Technology, Japan
Maho Nakata
The Institute of Physical and Chemical Research, Japan

SDPA (SemiDefinite Programming Algorithm) Project

- Software to solve the SDP (since 1995)
- Many implementations and members (over 10 people)
  - SDPA & SDPA-GMP (Fujisawa, Kojima, K. Nakata, Yamashita, Fukuda, M. Nakata and Kobayashi)
  - SDPARA (Yamashita, Fujisawa and Kojima)
  - SDPA-C (Nakata, Fukuda, Fujisawa and Kojima)
  - SDPARA-C (Nakata, Yamashita, Fujisawa and Kojima)
  - SDPA-M (Futakata, Matsuyama, Nakamura, Fujisawa, Kojima, Nakata and Yamashita)
- SDPA Online Solver (Fujisawa, Osada and Sasajima)
- Web site

http://sdpa.indsys.chuo-u.ac.jp/sdpa/

History : SDPA Project

- Pinpal (1994) : Mathematica, prototype of SDPA
- SDPA 0.1 (1995) : Borland C++ → gcc(g++)
- SDPA 1.x (1995) : Data structures and algorithms for sparse problems
  Binaries of SDPA were available from the Internet
- SDPA 2.x (1996) : Mehrotra-type predictor-corrector IPM
- SDPA 3.x (1997) : Partially support exploiting sparsity in Schur complement matrix
- SDPA 4.x (1998) : Fully support exploiting sparsity, Callable library
- SDPA 5.x (1999) : Accelerate computing some eigenvalues
- SDPA 6.x (2002) : Use the LAPACK and BLAS(ATLAS)
- SDPA 7.x (2008) : Sparse Schur complement matrix & Performance tuning & Optimized BLAS

- SDPA-C 1.x(2003) : Matrix completion
- SDPARA 1.x(2003) : Parallel computation of SDPA
- SDPARA-C 1.x(2004) : Parallel computation of SDPA-C
- SDPA Online Solver(2005) : Grid & Cluster computing technologies
- SDPA-GMP 7.x(2008) : Arbitrary precision

SDPA Family

SDPA&SDPA-GMP Primal-Dual IPM

1 **SDPA-C**

Matrix Completion
Primal-Dual IPM

2 **SDPARA**

MPI based parallel version of SDPA

3 A combination of SDPA-C and SDPARA

SDPARA-C

SDP(SemiDefinite Programming)

Primal

$$\left.\begin{array}{ll} \text{minimize} & \boldsymbol{A}_0 \bullet \boldsymbol{X} \\ \text{subject to} & \boldsymbol{A}_p \bullet \boldsymbol{X} = b_p \ (p = 1, 2, \ldots, m), \ \boldsymbol{X} \in \mathcal{S}^n_+ \end{array}\right\}. \quad (1)$$

Dual

$$\left.\begin{array}{ll} \text{maximize} & \displaystyle\sum_{p=1}^{m} b_p z_p \\ \text{subject to} & \displaystyle\sum_{p=1}^{m} \boldsymbol{A}_p z_p + \boldsymbol{Y} = \boldsymbol{A}_0, \ \boldsymbol{Y} \in \mathcal{S}^n_+ \end{array}\right\}. \quad (2)$$

$\mathcal{S}^n$ $n \times n$ symmetric matrices

$\boldsymbol{X} \bullet \boldsymbol{Y} = \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij} Y_{ij}$

$\boldsymbol{X} \in \mathcal{S}^n_+$ $(\mathcal{S}^n_{++})$: $\boldsymbol{X} \in \mathcal{S}^n$ is positive semidefinite (or positive definite)

$\boldsymbol{A}_p \in \mathcal{S}^n$ $(p = 0, 1, \ldots, m)$ and $\boldsymbol{b} \in \mathbb{R}^m$

LP and SDP Progress

- LP : CPLEX (ILOG) : 1988 → 2004
  - Algorithms 3,300x, Computer 1,600x
  - Net: 5,300,000x
- SDP : SDPA Solver : 1996 → 2008
  - Algorithms 700.7 / 1.8 = 389.3x,
  - Computer 133,892.5 / 700.7 = 191.1 x
  - Net: 133.892.5 / 1.8 = 74,384.7x

- mcp500-1
- Max cut problem
- 500 nodes

SDPA 7.1.0 (2008)	SDPA 2.0.1 (2008)	SDPA 2.0.1 (1996)
1.8 sec.	700.7 sec. (≒ 11.7 min.)	133,892.5 sec. (≒ 37.2 h)
Intel Xeon X5365 3.0GHz, memory 48GB	Intel Xeon X5365 3.0GHz, memory 48GB	MIPS R4400 133MHz, memory 128MB

Main features of the SDPA 7.1.0

- Implementation of the Mehrotra-type primal-dual predictor-corrector interior-point method for SDP
- Written in C++ language
- Handle diagonal and sparse matrices
- Incorporates an efficient method for computing the search direction when the problem is large scale and sparse
- Sparse Schur complement matrix and Sparse Cholesky Factorization
- Performance tunings and Optimized BLAS
- SOCP (Second Order Cone Problem)

Bottlenecks of IPM

- If we don't explot the sparsity….

	Dense	Sparse
n = m (n >> m)	ELEMENTS	ELEMENTS
n << m	ELEMENTS	ELEMENTS

- SDPA 7.x

	Dense	Sparse
n = m (n >> m)	ELEMENTS	Other O(n^3) parts
n << m	ELEMENTS	CHOLESKY

- ELEMENTS :: $O(m^2n^2+mn^3)\Rightarrow$ Computation of all elements of Schur complement matrix B
- CHOLESKY:: $O(m^3)\Rightarrow$ Cholesky factorization of B

Bottlenecks on SDPA

	Control11 (n=165, m=1596)	Theta6 (n=300, m=4375)
ELEMENTS	451.5(90.6%)	77.1 (26.4%)
CHOLESKY	37.7(7.6%)	203.0 (69.4%)
Others	9.2(1.8%)	12.4 (4.2%)
Total	498.4(100%)	292.5(100%)

- ELEMENTS :: $O(m^2n^2+mn^3)\Rightarrow$ Computation of elements of Schur complement matrix B
- CHOLESKY:: $O(m^3)\Rightarrow$ Cholesky factorization of B
- SDPA-6.0
- Pentium 4 (2.2GHz), Memory 1GB, Linux 2.4.18

The SDPARA (SemiDefinite Programming Algorithm PARAllel version)

$$Prim\begin{cases}\min : A_0 \bullet X \\ \text{s.t.} : A_i \bullet X = b_i \\ \qquad X \succeq O \end{cases} \qquad Dual\begin{cases}\max : \sum_{i=1}^{m} b_i z_i \\ \text{s.t.} : \sum_{i=1}^{m} A_i z_i + Y = A_0 \\ \qquad Y \succeq O \end{cases}$$

$i = 1, 2, \ldots, m$

The SDPARA is a parallel version of the SDPA on multiple processors, which replace two bottleneck parts (ELEMENTS and CHOLESKY) with their parallel implementation using MPI and ScaLAPACK.

- **ELEMENTS** ⇒Computation of Schur complement matrix
- **CHOLESKY** ⇒ Cholesky factorization of Schur complement matrix
- PC Cluster(Presto III); CPU Athlon 1900+ , 768MB

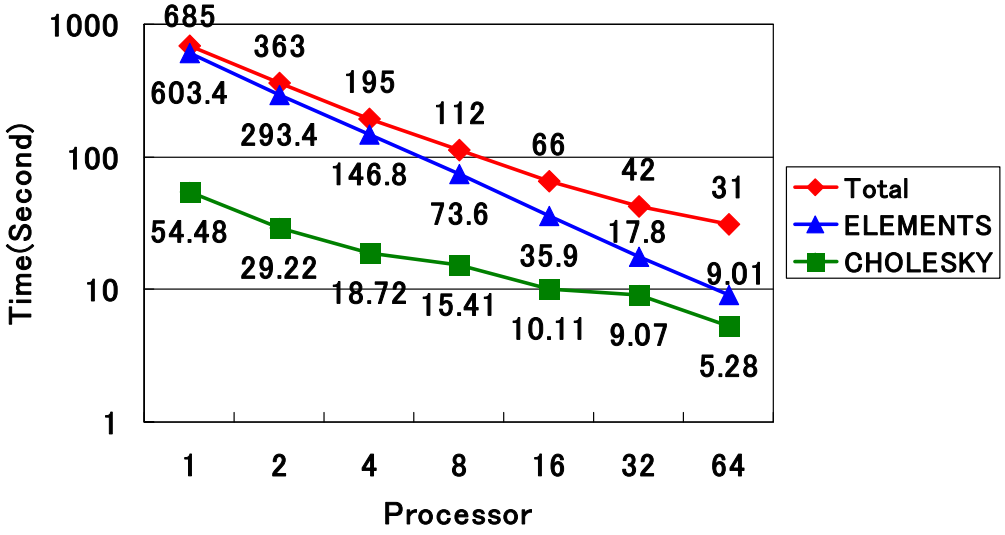


Table 1: SDPs arsing from the quantum chemistry

molecules	$2K$	N	m	n	#nonzero	nBLOCK	bLOCKsTRUCT
HF(PQG)	24	10	15018	1498	105694	14	(288,144×4,. . .,-322)
NH(T1T2)	24	8	15018	10170	2205558	22	(2532×2,792×4,. . .,-322)
BH_3O(T1T2)	26	16	20709	12828	3290622	22	(3224×2,1014×4,. . .,-374)
H_2O(T1T2)	28	10	27888	15914	4766902	22	(4032×2,1274×4,. . .,-430)

Peak performance

1TFlops

Table 2: Numerical results of the SDPARA for huge-scale SDPs

Problem		1 CPU	4 CPU	8 CPU	16 CPU	64 CPU	128 CPU	256 CPU
HF(PQG)	Total time	47483	8939		2549	1120		
	ELEMENTS	16719	4390		706	237		
	CHOLESKY	20995	3395		983	331		
NH(T1T2)	Total time					66015	37028	24499
	ELEMENTS					47416	19958	9875
	CHOLESKY					820	362	285
BH_3O(T1T2)	Total time					148387		
	ELEMENTS					104745		
	CHOLESKY					1989		
H_2O(T1T2)	Total time			2060237				
	ELEMENTS			1985337				
	CHOLESKY			22137				

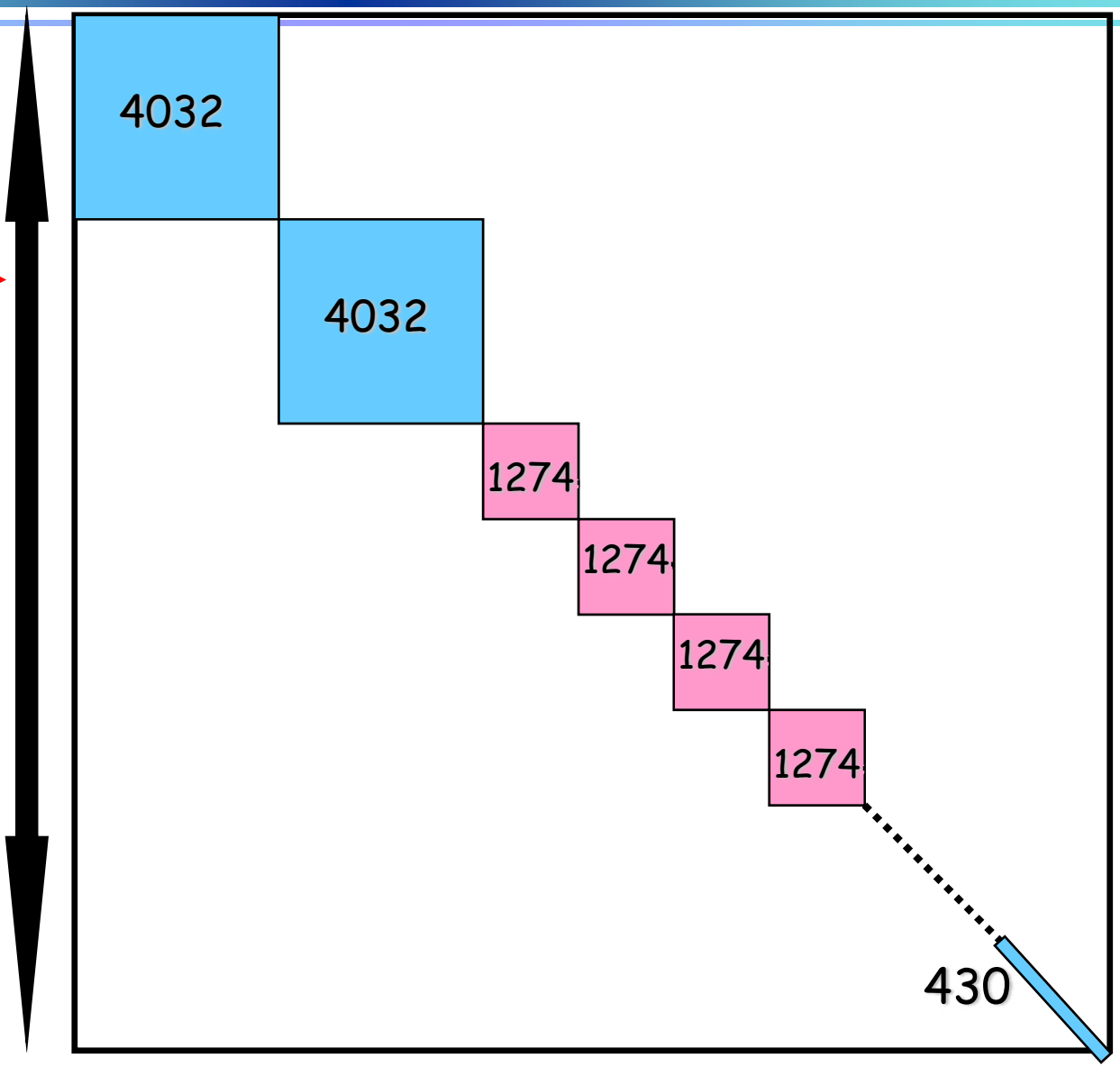
Total memory

409.6GB

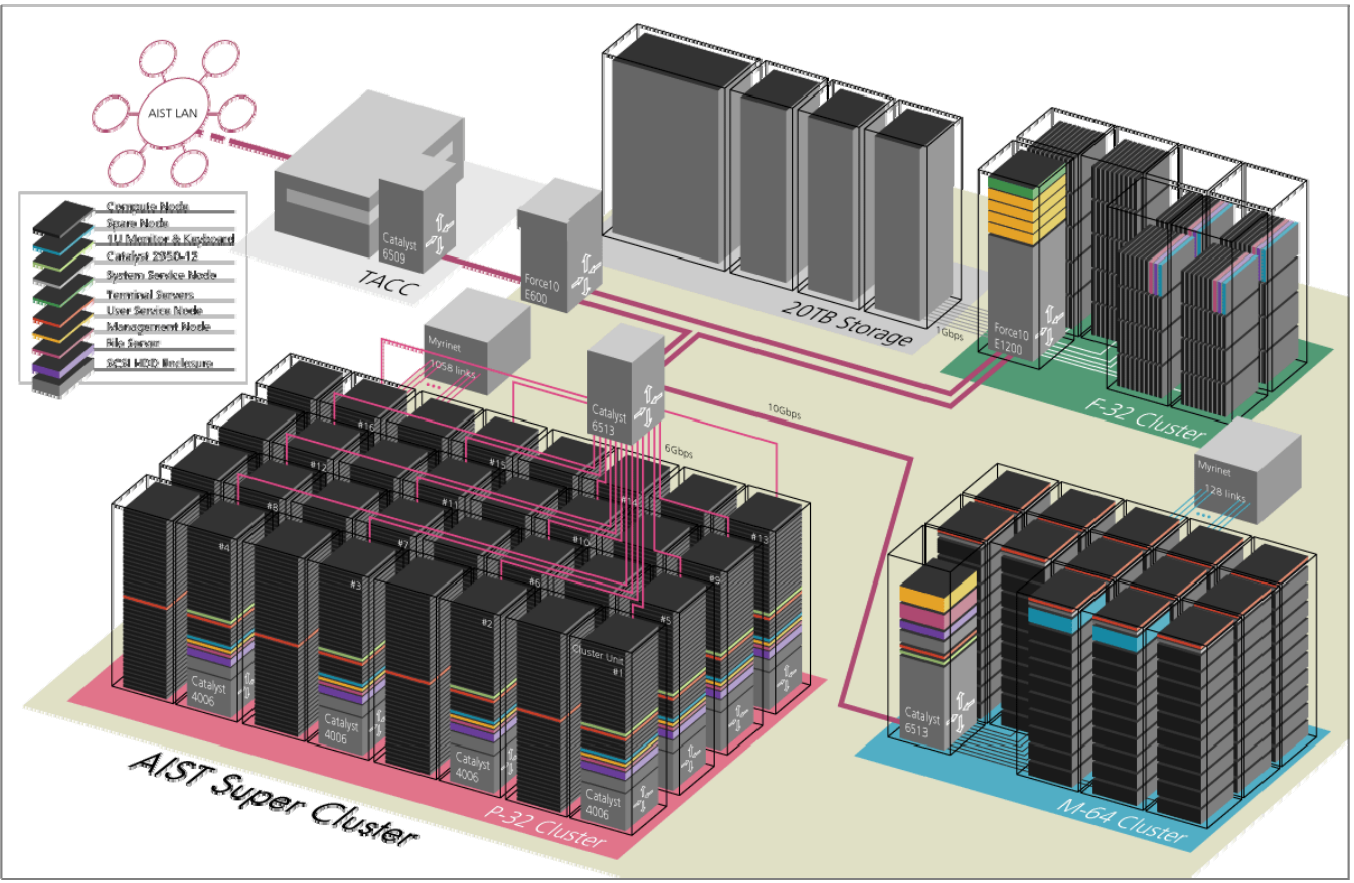
24days!!

Block diagonal structure of H2O SDP

- n = 15914
(matrix size)
- m = 27888
(# of constraints)
- 4766902
(# of nonzero elements)



AIST GTRC (Grid) Super Cluster



P32: IBM eServer325
Opteron 2.0GHz, 6GB
2way x 1072 node
Myrinet 2000
6.16TFlops/LINPACK

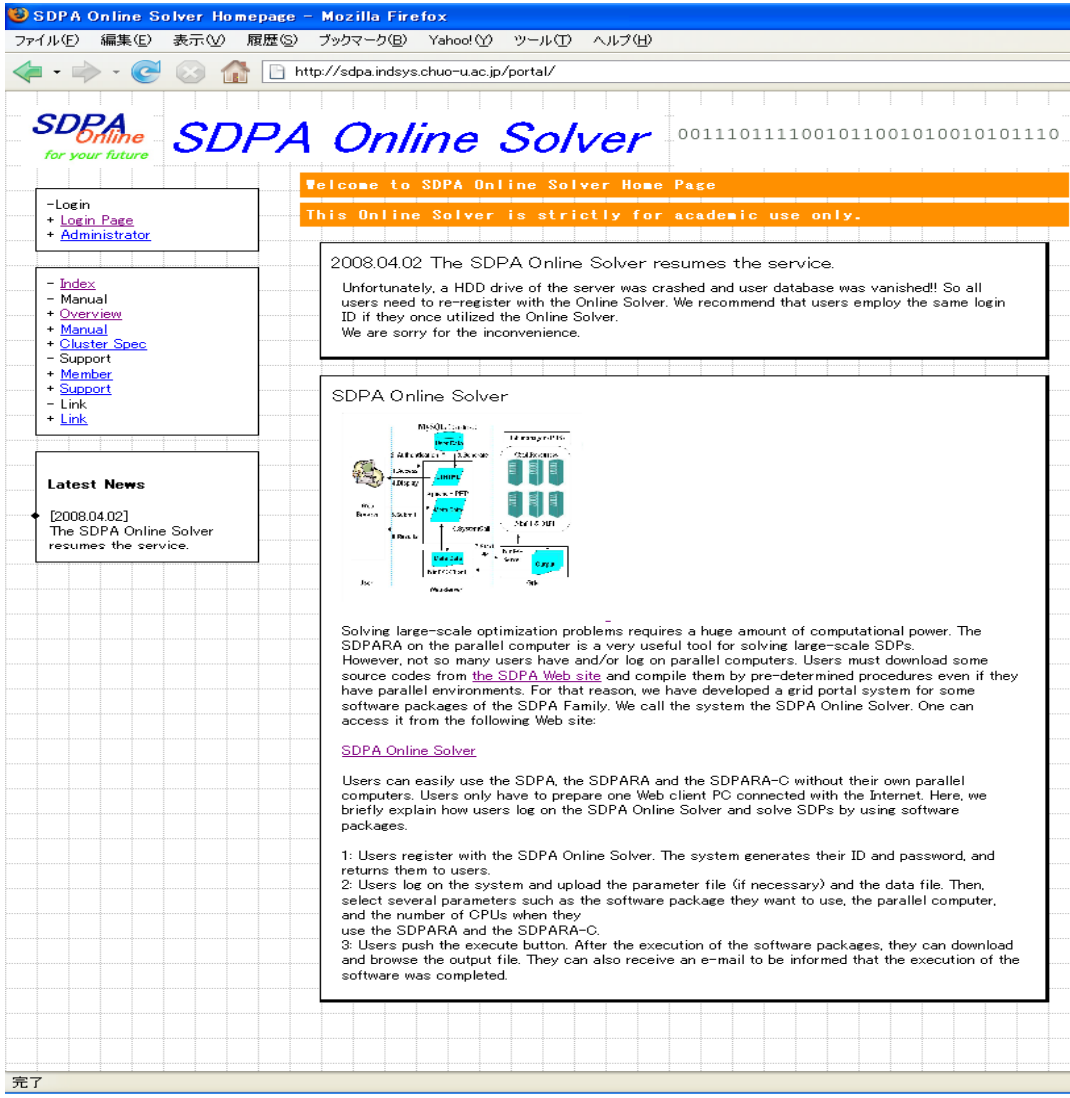
M64: Intel Tiger 4
Madison 1.3GHz, 16GB
4way x 132 node
Myrinet 2000
1.62TFlops/LINPACK

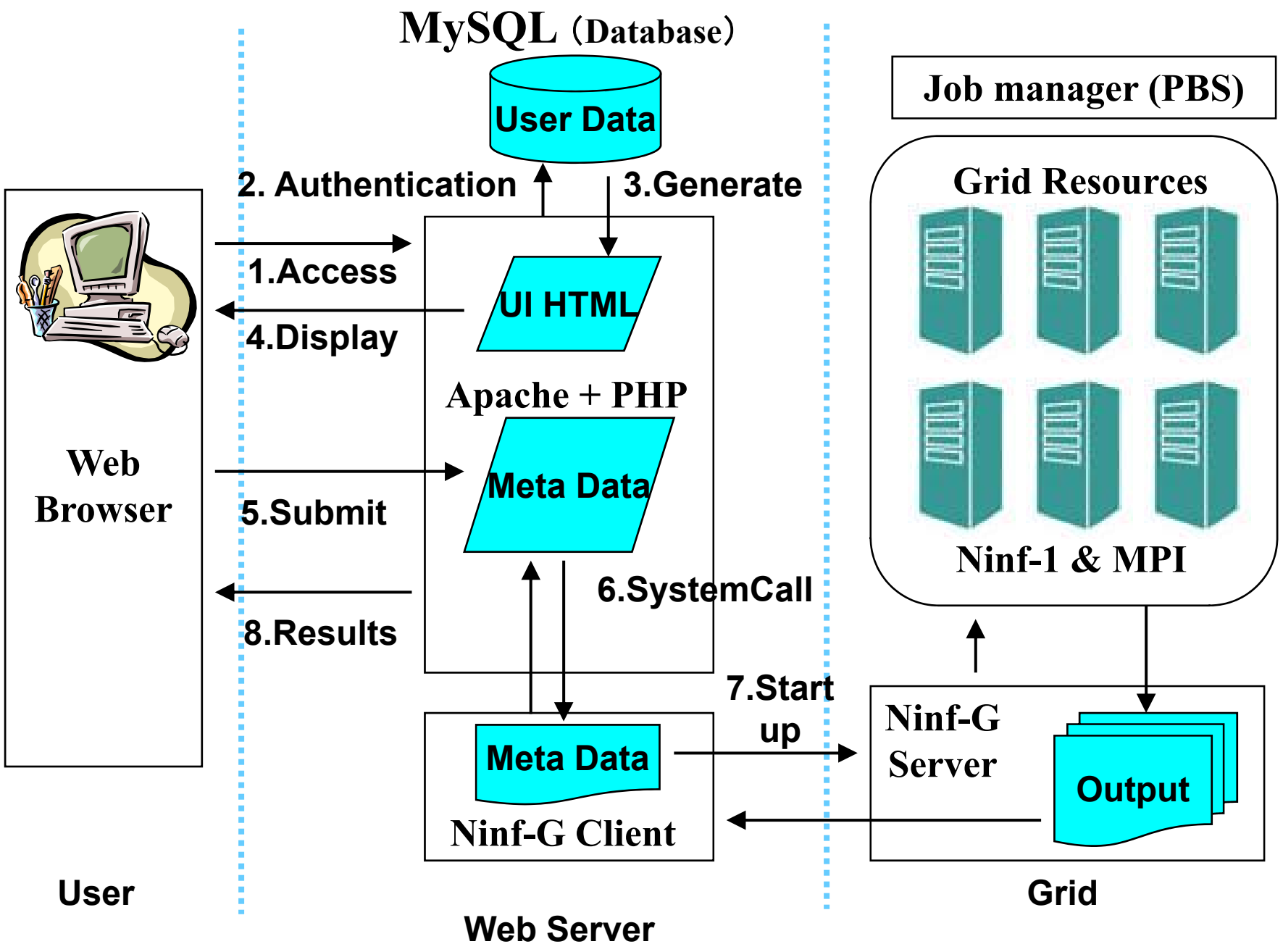
F32: Linux Networx
Xeon 3.06GHz, 2GB
2way x 268 node
GbE
2.00TFlops/LINPACK

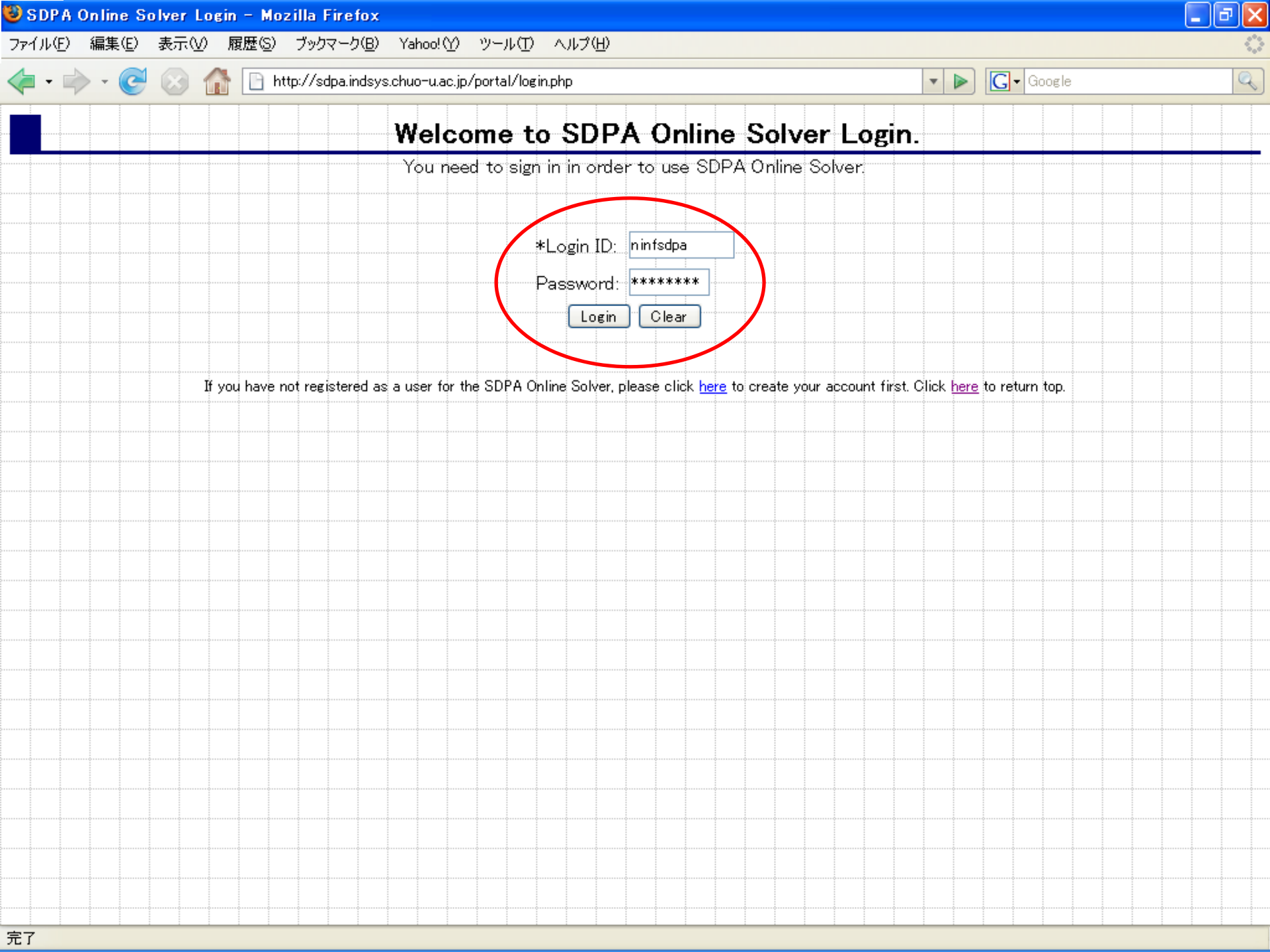
total 9.78TFlops/LINPACK, 3208 CPUs

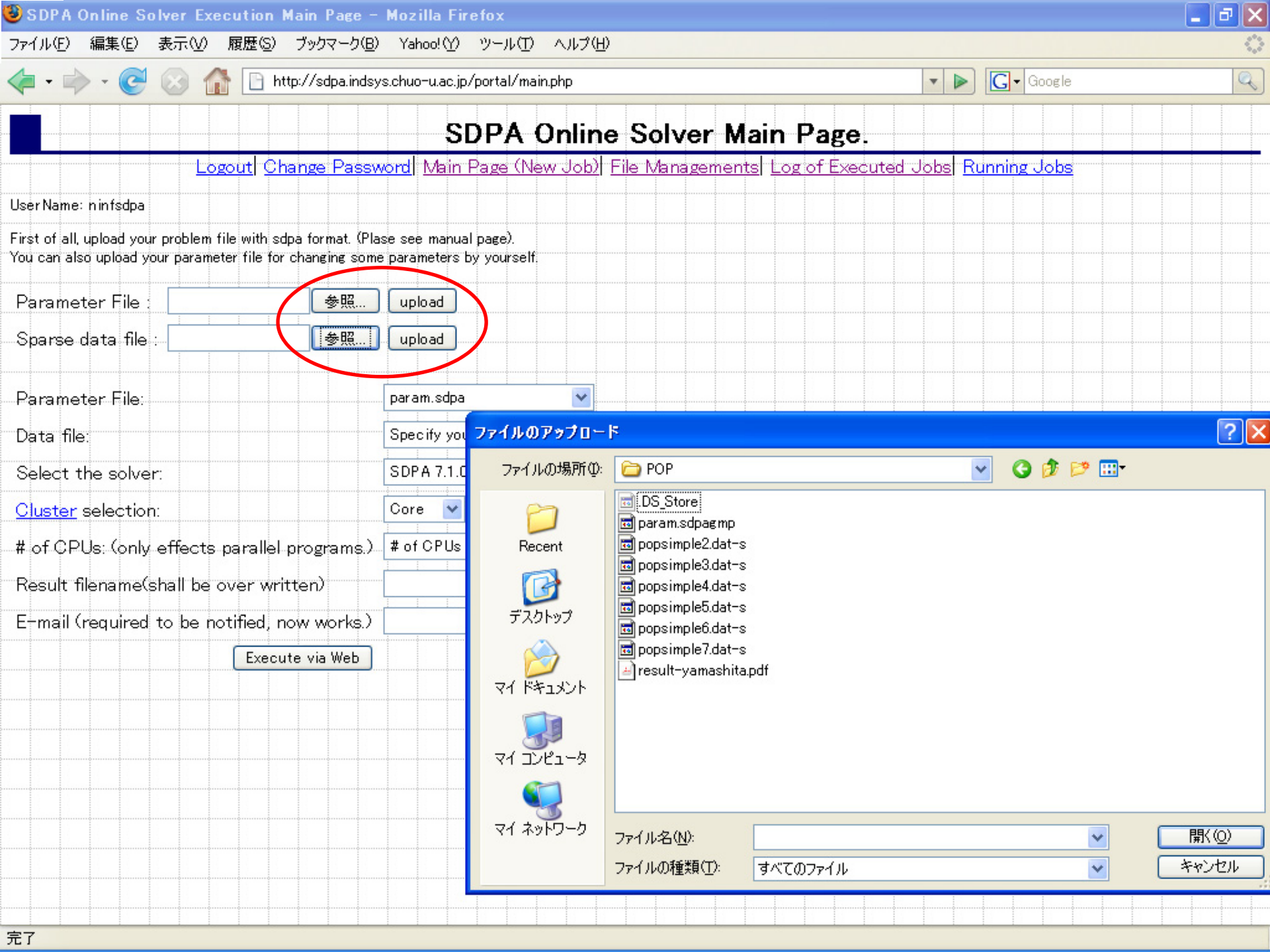
SDPA Online Solver

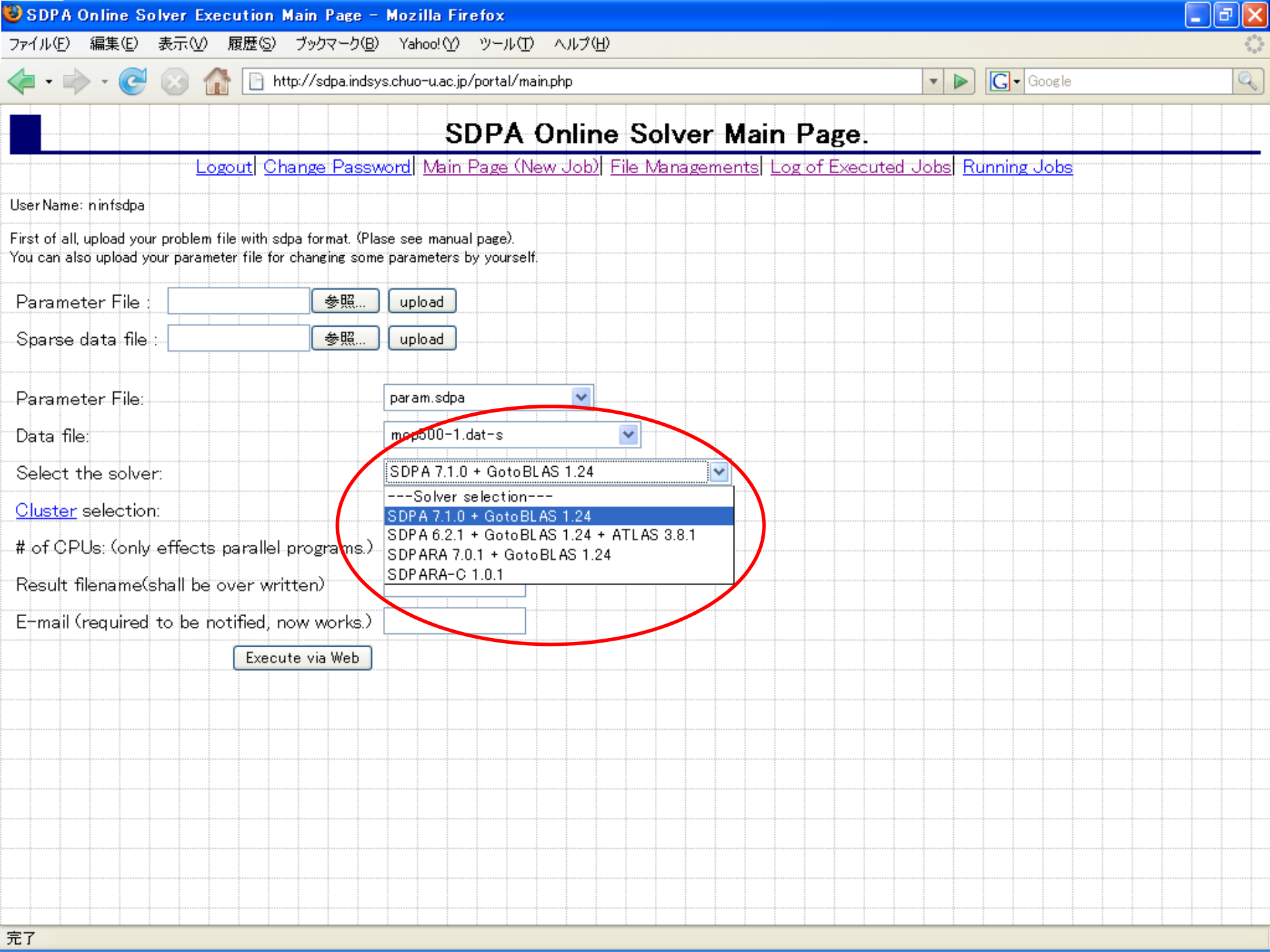
- A grid portal system for the SDPA, SDPARA, SDPARA-C
  - Users access the grid portal site
  - select an software
  - send a problem to the PC cluster
- Enables users to easily perform parallel computation through the Internet.
- Ninf and Globus softwares

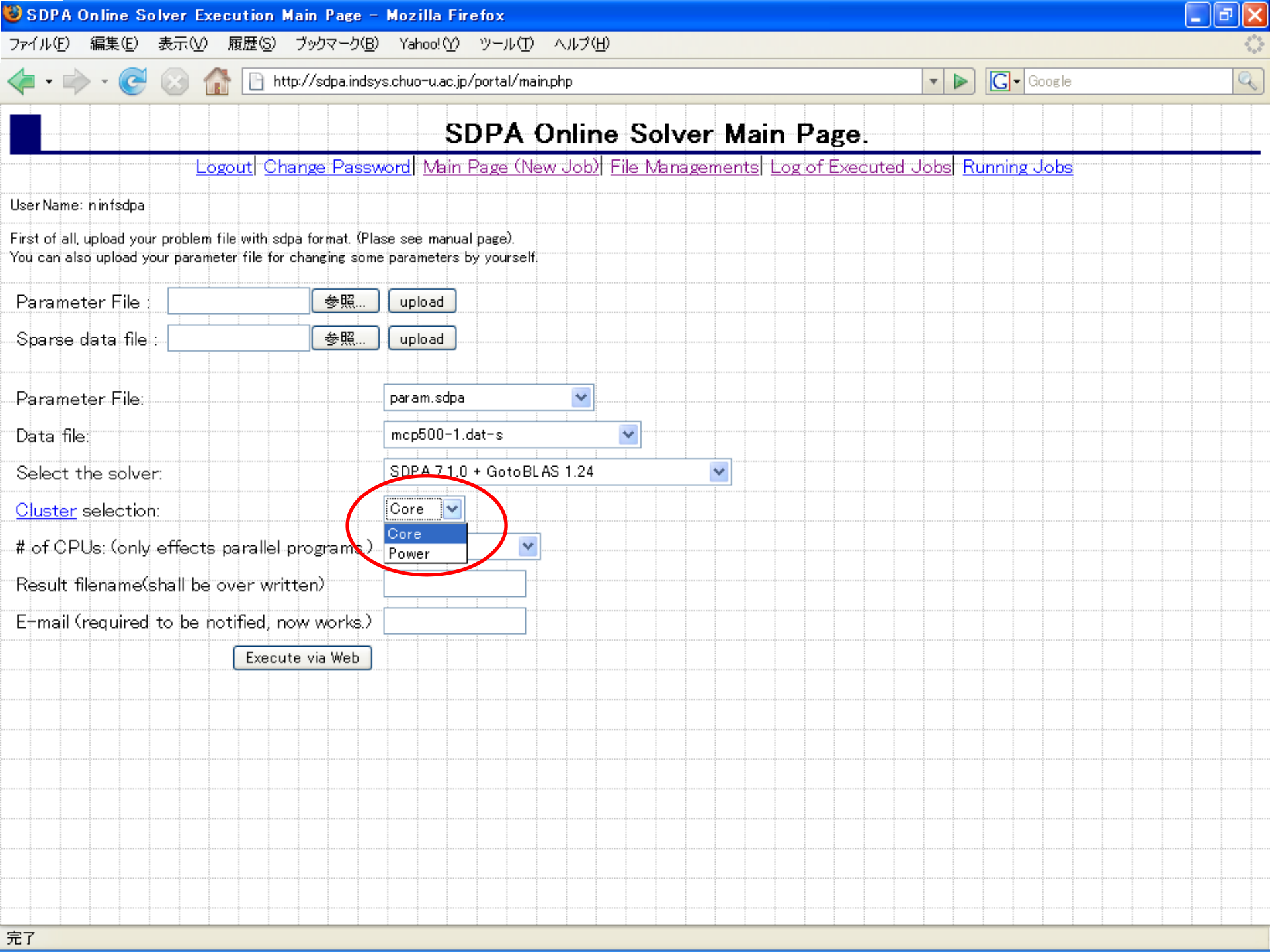


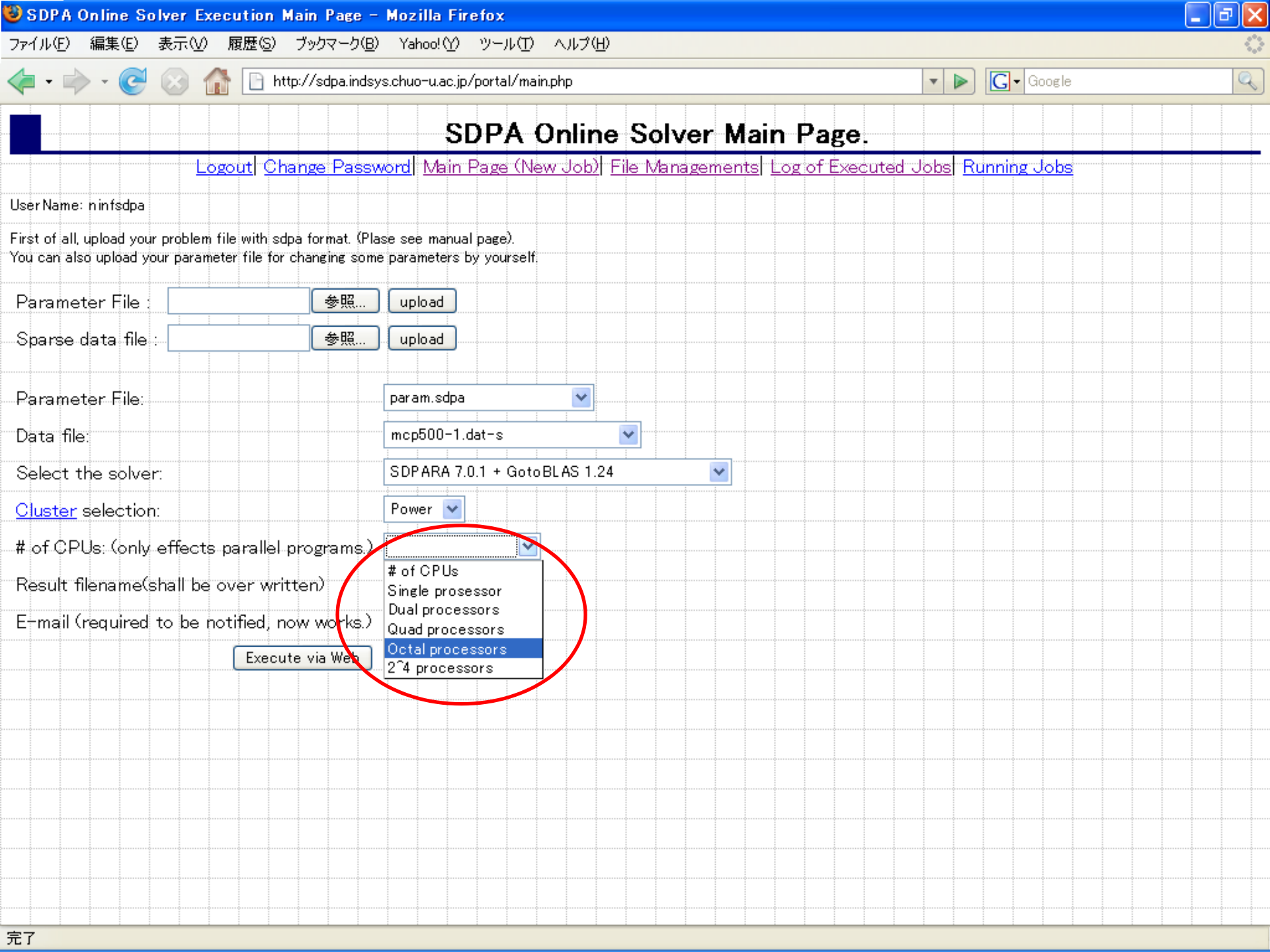

MySQL (Database)
User Data
Job manager (PBS)
2. Authentication
3.Generate
Grid Resources
1.Access
4.Display
UI HTML
Apache + PHP
Web Browser
Meta Data
5.Submit
Ninf-1 & MPI
6.SystemCall
8.Results
7.Start up
Ninf-G Server
Meta Data
Output
Ninf-G Client
User
Grid
Web Server


SDPA Online Solver Login - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) Yahoo!(Y) ツール(T) ヘルプ(H)
http://sdpa.indsys.chuo-u.ac.jp/portal/login.php
Google
Welcome to SDPA Online Solver Login.
You need to sign in in order to use SDPA Online Solver.
*Login ID: ninfsdpa
Password: ********
Login
Clear
If you have not registered as a user for the SDPA Online Solver, please click here to create your account first. Click here to return top.
完了


SDPA Online Solver Execution Main Page - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) Yahoo!(Y) ツール(T) ヘルプ(H)
http://sdpa.indsys.chuo-u.ac.jp/portal/main.php
Google
SDPA Online Solver Main Page.
Logout | Change Password | Main Page (New Job) | File Managements | Log of Executed Jobs | Running Jobs
User Name: ninfsdpa
First of all, upload your problem file with sdpa format. (Plase see manual page).
You can also upload your parameter file for changing some parameters by yourself.
Parameter File :
参照...
upload
Sparse data file :
参照...
upload
Parameter File:
param.sdpa
Data file:
Specify you
Select the solver:
SDPA 7.1.0
Cluster selection:
Core
# of CPUs: (only effects parallel programs.)
# of CPUs
Result filename(shall be over written)
E-mail (required to be notified, now works.)
Execute via Web
ファイルのアップロード
ファイルの場所(I):
POP
Recent
デスクトップ
マイ ドキュメント
マイ コンピュータ
マイ ネットワーク
.DS_Store
param.sdpapmp
popsimple2.dat-s
popsimple3.dat-s
popsimple4.dat-s
popsimple5.dat-s
popsimple6.dat-s
popsimple7.dat-s
result-yamashita.pdf
ファイル名(N):
ファイルの種類(T):
すべてのファイル
開く(O)
キャンセル
完了


SDPA Online Solver Execution Main Page - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) Yahoo!(Y) ツール(T) ヘルプ(H)
http://sdpa.indsys.chuo-u.ac.jp/portal/main.php
Google
SDPA Online Solver Main Page.
Logout | Change Password | Main Page (New Job) | File Managements | Log of Executed Jobs | Running Jobs
User Name: ninfsdpa
First of all, upload your problem file with sdpa format. (Plase see manual page).
You can also upload your parameter file for changing some parameters by yourself.
Parameter File :
参照...
upload
Sparse data file :
参照...
upload
Parameter File:
param.sdpa
Data file:
mcp500-1.dat-s
Select the solver:
SDPA 7.1.0 + GotoBLAS 1.24
---Solver selection---
SDPA 7.1.0 + GotoBLAS 1.24
SDPA 6.2.1 + GotoBLAS 1.24 + ATLAS 3.8.1
SDPARA 7.0.1 + GotoBLAS 1.24
SDPARA-C 1.0.1
Cluster selection:
# of CPUs: (only effects parallel programs.)
Result filename(shall be over written)
E-mail (required to be notified, now works.)
Execute via Web
完了


SDPA Online Solver Execution Main Page - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) Yahoo!(Y) ツール(T) ヘルプ(H)
http://sdpa.indsys.chuo-u.ac.jp/portal/main.php
Google
SDPA Online Solver Main Page.
Logout | Change Password | Main Page (New Job) | File Managements | Log of Executed Jobs | Running Jobs
User Name: ninfsdpa
First of all, upload your problem file with sdpa format. (Plase see manual page).
You can also upload your parameter file for changing some parameters by yourself.
Parameter File :
参照...
upload
Sparse data file :
参照...
upload
Parameter File:
param.sdpa
Data file:
mcp500-1.dat-s
Select the solver:
SDPA 7.1.0 + GotoBLAS 1.24
Cluster selection:
Core
Core
Power
# of CPUs: (only effects parallel programs.)
Result filename(shall be over written)
E-mail (required to be notified, now works.)
Execute via Web
完了


SDPA Online Solver Execution Main Page - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) Yahoo!(Y) ツール(T) ヘルプ(H)
http://sdpa.indsys.chuo-u.ac.jp/portal/main.php
Google
SDPA Online Solver Main Page.
Logout | Change Password | Main Page (New Job) | File Managements | Log of Executed Jobs | Running Jobs
User Name: ninfsdpa
First of all, upload your problem file with sdpa format. (Plase see manual page).
You can also upload your parameter file for changing some parameters by yourself.
Parameter File :
参照...
upload
Sparse data file :
参照...
upload
Parameter File:
param.sdpa
Data file:
mcp500-1.dat-s
Select the solver:
SDPARA 7.0.1 + GotoBLAS 1.24
Cluster selection:
Power
# of CPUs: (only effects parallel programs.)
Result filename(shall be over written)
E-mail (required to be notified, now works.)
Execute via Web
# of CPUs
Single prosessor
Dual processors
Quad processors
Octal processors
2^4 processors
完了

File management page.

You can download/delete your files.

DataFile:

BH.1Sigma+.DZ.pqgt1t2p.dat-s	Download	Delete	47272317Bytes
Bex5_2_2_case1.dat-s	Download	Delete	343998Bytes
Bex5_2_2_case3.dat-s	Download	Delete	343998Bytes
Bex9_1_5.dat-s	Download	Delete	71342Bytes
Bex9_1_8.dat-s	Download	Delete	592646Bytes
CH4.1A1.STO6G.pq.dat-s	Download	Delete	887356Bytes
control11.dat-s	Download	Delete	8616690Bytes
g1717.dat-s	Download	Delete	462217Bytes
gpp1000.dat-s	Download	Delete	10501537Bytes
gpp250-1.dat-s	Download	Delete	489486Bytes
mcp.dat-s	Download	Delete	7692Bytes
mcp500-1.dat-s	Download	Delete	33663Bytes
theta4.dat-s	Download	Delete	374526Bytes
theta5.dat-s	Download	Delete	591344Bytes

Parameter File:

param.sdpa	Download	Delete	356Bytes

Result file:

output00:38_27-Apr-2008.out	Download	Delete	1415318Bytes
output16:53_24-Apr-2008.out	Download	Delete	22028125Bytes
output21:37_25-Apr-2008.out	Download	Delete	5515364Bytes

(Total file size: 99547975 Bytes)

Gmail - SDPA Online Solver Result - katsuki.fujisawa@gmail.com - Mozilla Firefox

ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) Yahoo!(Y) ツール(T) ヘルプ(H)

http://mail.google.com/mail/#search/ninfsdpa/1178d5aad3e1c2a7

Gmail by Google BETA

ninfsdpa メールを検索 ウェブを検索 検索オプションを表示 フィルタを作成

メールを作成

受信トレイ (4051)
スター付き
チャット
送信済みメール
下書き
すべてのメール
迷惑メール (13)
ゴミ箱

連絡先

▼チャット

検索、追加、招待

Katsuki Fujisawa
ステータスの設定

Akiyoshi Shioura
Hayato Waki
MIKIO KUBO
MIYAMOTO Yuichir
Yuichiro Yasui
村松正和
Katsuki Fujisawa
Kazushige Goto
kojima-sdpa
Maho NAKATA

オプション 連絡先を追加

▼ラベル

本日の必読記事 - 【4月30日の必読記事】森永卓郎: 総理がわびるべきは暫定税率よりほかにある - 04/30 ウェブ クリップ

« 検索結果に戻る 迷惑メールを報告 削除 その他の操作 ‹ 前 2 / 3 次 ›

SDPA Online Solver Result

受信トレイ ×

noreply@sdpa.indsys.chuo-u.ac.jp 宛先 自分 詳細を表示 1月18日 返信

Dear ninfsdpa.

Your job had successfully calculated.

----------Execution Details----------
Result : output23:39_18-Jan-2008.out
Parameter : param.sdpa
Data : mcp500-1.dat-s
Solver : SDPA 7.0.2 Goto BLAS 1.19@sdpa cluster
CPU : 4

To download result, please visit download page via our site:

http://sdpa.indsys.chuo-u.ac.jp/portal/login.php

If you have any question, advice, and error report, please contact with any member of sdpa.indsys.chuo-u.ac.jp or, maintainer: wata2or3@gmail.com .

Sincerely.

SDPA Online Solver
Chuo University in Tokyo
Department: Engineering of Management System
1-13-27 Kasuga Bunkyoh-ku Tokyo-to
Postal code: 112-8551

別のウィンドウで開く
すべて印刷
すべて閉じる
すべて転送
ハイライト表示オン

noreply@sdpa.indsys.chuo-u.ac.jp

完了

SDPA 7.1.0 and SDPA 6.2.1

- The SDPA 7.1.0 is completely revised from the SDPA 6.2.1
  - (i) great performance improvements on its computational time and memory usage.
  - (ii)fast sparse Cholesky factorization when the Schur Complement Matrix is sparse.
  - (iii) some improvements on its numerical stability due to a better control in the interior-point algorithm.
  - (iv) arbitrary precision arithmetic when the condition numbers of some matrices at any iterate become ill-conditioned. $\Rightarrow$ SDPA-GMP

SDPA and optimized BLAS

Table 1: Numerical experiments (SDPA 7.0.5 + BLAS library) : time:second(# of iterations) : CPU; Xeon 5345 (2.33GHz), OS; CentOS Ver 5 (64bit), Compiler : Intel (C/C++ & Fortran) 10.1

BLAS library(# of threads)	Prob(1)	Prob(2)	Prob(3)	Prob(4)
BLAS/LAPACK 3.1.1(1)	72.32(35)	126.63(15)	221.31(18)	344.81(20)
ATLAS 3.8.0(1)	51.09(35)	49.73(15)	40.35(18)	180.64(19)
ATLAS 3.8.0(4)	30.06(35)	17.70(15)	17.09(18)	80.57(20)
Intel MKL 10.0.1.014(1)	44.97(36)	39.63(15)	34.27(18)	170.58(21)
Intel MKL 10.0.1.014(2)	31.44(34)	24.03(15)	22.23(18)	110.19(22)
Intel MKL 10.0.1.014(4)	26.18(35)	16.15(15)	16.17(18)	67.17(18)
Intel MKL 10.0.1.014(8)	24.06(35)	13.17(15)	14.18(18)	59.50(19)
GotoBLAS 1.22(1)	43.09(35)	40.23(15)	32.06(18)	157.18(21)
GotoBLAS 1.22(2)	32.16(35)	23.48(15)	21.15(18)	96.01(19)
GotoBLAS 1.22(4)	26.25(35)	15.11(15)	15.43(18)	68.77(19)
GotoBLAS 1.22(8)	24.66(36)	11.79(15)	13.67(18)	60.70(19)

Prob(1) : Structural Optimization
Prob(2) : Combinatorial Optimization(1)
Prob(3) : Combinatorial Optimization(2)
Prob(4) : Quantum Chemistry

- We strongly recommend you to use optimized BLAS and LAPACK, e.g.,

Table 1 shows how the SDPA 7.x performs on several benchmark problems when changing the BLAS and LAPACK library. Typically, the SDPA 7.x with optimized BLAS and LAPACK seems much faster than one with BLAS/LAPACK 3.1.1.

Numerical Results 1(Sparse SDPs) (SDPA, CSDP, SDPT3, SeDuMi)

Table 1: Numerical experiments(Sparse SDP) : time:sec.(# of iterations)

	SDPA 7.1.0	CSDP 6.0.1	SDPT3-4.0β	SeDuMi 1.1
mater-1	0.04(16)	0.06(16)	0.90(19)	0.31(21)
mater-2	0.24(20)	0.40(19)	2.30(19)	0.98(25)
mater-3	1.38(23)	5.23(23)	9.50(23)	4.53(27)
mater-4	7.98(26)	144.06(27)	44.30(30)	30.69(30)
mater-5	24.61(27)	1345.85(29)	113.30(33)	121.17(33)
mater-6	88.99(36)	-	333.40(41)	490.10(36)
r2S_broydenTri600	2.16(20)	1577.47(21)	18.40(18)	11.04(14)
r2S_broydenTri700	2.63(20)	-	22.20(18)	14.83(15)
r2S_broydenTri800	3.08(20)	-	25.60(18)	18.03(14)
r2S_broydenTri900	3.46(20)	-	29.50(18)	22.52(14)
nonc_500	0.72(29)	180.19(31)	8.60(32)	3.65(22)
ros_500	0.60(23)	129.73(23)	8.50(32)	3.03(17)
rabmo	175.38(21)	258.12(28)	747.60(51)	2044.61(21)
trto3	2.73(21)	19.10(39)	47.0(23)	67.09(59)
trto4	23.38(24)	116.56(38)	31.50(29)	840.10(72)
trto5	297.76(22)	1124.85(52)	354.80(29)	-
vibra3	8.64(32)	37.61(43)	12.40(32)	159.31(69)
vibra4	66.86(35)	288.69(50)	96.00(46)	1920.80(95)
vibra5	1024.25(40)	2088.95(60)	1499.90(68)	-

- SDPA and CSDP with GotoBLAS 1.24

CPU : **Intel Xeon 5635 3GHz** : 2 CPU, 8 cores
OS : RHEL5 64bit
Compiler : Intel C++/Fortran 10.1.012
MATLAB : 7.5.0.338 (R2007b)

Numerical Results 2 (Dense SDPs) (SDPA, CSDP, SDPT3, SeDuMi)

•GotoBLAS on 4 threads

Table 2: Numerical experiments(Dense SDP (1)) : time:sec.(# of iterations)

	SDPA 7.1.0	CSDP 6.0.1	SDPT3-4.0β	SeDuMi 1.1	SDPA 7.1.0(4)	CSDP 6.0.1(4)
arch8	1.36(26)	1.62(25)	2.60(24)	4.09(28)	1.38(26)	1.59(25)
control10	56.08(40)	60.12(28)	51.60(28)	86.84(43)	51.66(40)	52.93(29)
equalG11	24.35(18)	73.09(24)	47.10(16)	316.11(16)	10.10(17)	24.54(19)
equalG51	50.00(20)	97.96(19)	89.70(18)	1690.10(30)	19.10(18)	47.66(21)
gpp250-1	0.93(20)	2.66(25)	2.70(18)	24.56(33)	0.66(19)	2.30(29)
gpp250-4	0.93(19)	1.78(20)	2.10(14)	28.93(40)	0.59(17)	1.57(20)
gpp500-1	6.91(20)	70.05(58)	16.60(20)	212.53(40)	3.54(20)	67.48(66)
gpp500-4	6.96(21)	15.55(21)	14.70(17)	175.73(30)	3.40(19)	7.69(20)
maxG11	17.49(16)	17.35(16)	16.90(15)	180.02(13)	7.34(16)	11.42(16)
maxG32	243.42(17)	182.87(17)	176.60(16)	-	88.65(17)	98.94(17)
maxG51	32.52(16)	48.92(17)	37.30(17)	533.80(16)	13.38(16)	23.82(17)
mcp250-1	0.55(15)	0.82(15)	1.10(14)	6.50(15)	0.41(15)	0.75(15)
mcp250-4	0.53(14)	0.77(14)	1.20(13)	6.13(14)	0.40(14)	0.71(14)
mcp500-1	4.26(16)	4.86(16)	3.80(15)	49.21(16)	2.14(16)	3.58(16)
mcp500-4	4.03(15)	6.54(15)	5.80(14)	44.50(14)	2.05(14)	3.78(15)
qpG11	120.22(16)	100.15(17)	16.00(16)	2690.48(14)	44.25(16)	54.68(17)
qpG51	250.57(19)	209.40(20)	34.50(19)	-	84.63(19)	116.86(20)
ss30	6.12(22)	8.38(21)	9.90(21)	30.99(27)	5.69(22)	5.47(23)
truss8	1.64(20)	0.94(20)	2.70(17)	2.11(23)	2.16(20)	1.28(20)
theta4	7.64(18)	7.98(17)	11.70(16)	82.59(16)	4.63(18)	4.61(17)
theta5	23.78(18)	26.44(17)	32.90(16)	287.03(16)	12.36(18)	12.37(17)
theta6	64.60(18)	72.56(17)	110.40(17)	865.99(16)	29.96(18)	30.70(17)

GMP(GNU Multi Precision) & SDPA-GMP

- GMP : A free library for arbitrary precision arithmetic, operating on signed integers, rational numbers, and floating point numbers
- The cost of computation is very expensive
- widely used in various fields
- replace some (not all) double type variables in source codes of SDPA, LAPACK and BLAS with variables with arbitrary precision defined by the GMP → SDPA-GMP

Replacing variables

```
1.  //http://www.netlib.org/blas/dtrmm.f
2.  //B := alpha*A'*B
3.  int gmp_dtrmmLLTN(int M, int N, mpf_class alpha,
    mpf_class *A, int LDA, mpf_class *B, int LDB)
4.  {
5.    mpf_class mtmp;
6.    for(int j=0; j<N; j++){
7.      for(int i=0; i<M; i++){
8.        mtmp = B[i+j*LDB] *A[i+i*LDA];
9.        for (int k=i+1; k<M; k++){
10.         mtmp = mtmp + A[k+i*LDA]*B[k+j*LDB];
11.       }
12.       B[i+j*LDB] = alpha * mtmp;
13.     }
14.   }
15. }
```

double tmp

Parameter file of SDPA-GMP

```
200     unsigned int maxIteration;
1.0E-30 double 0.0 < epsilonStar;
1.0E4   double 0.0 < lambdaStar;
2.0     double 1.0 < omegaStar;
-1.0E5  double lowerBound;
1.0E5   double upperBound;
0.1     double 0.0 <= betaStar <  1.0;
0.3     double 0.0 <= betaBar  <  1.0, betaStar <= betaBar;
0.9     double 0.0 < gammaStar  <  1.0;
1.0E-30 double 0.0 < epsilonDash;
384     precision  ⟸ the number of significant bit of
                     GMP library (double variable has 53
                     significant bits)
```

Comparison of SDPA and SDPA-GMP(384bit)

- SDPA(7.1.0)

Relative gap

5.320136190426011e-07

Objective Function

-7.343076174864592e+00

(Primal)

-7.343080081482162e+00

(Dual)

Feasibility

5.45696821063e-12 (Primal)

1.68252292320e-07 (Dual)

Computation time

0.14 sec. (20 iterations)

- SDPA-GMP(7.1.0)

Relative gap

1.7163710368162993e-26

Objective Function

-7.343076265246537e+00

(Primal)

-7.343076265246537e+00

(Dual)

Feasibility

2.071019484472 1e-57 (Primal)

1.232941703970 2e-29 (Dual)

Computation time

228.95 sec. (59 iterations)

Benchmark problem : gpp124-1(SDPLIB)

Future Plans

- The new SDPA automatically selects the algorithm including SDPA, SDPARA, SDPA-C, SDPARA-C.